DeepSide: A Deep Learning Approach for Drug Side Effect Prediction

Onur Can Uner, Halil Ibrahim Kuru, R. Gokberk Cinbis, Oznur Tastan, and A. Ercument Cicek

Abstract—Drug failures due to unforeseen adverse effects at clinical trials pose health risks for the participants and lead to substantial financial losses. Side effect prediction algorithms have the potential to guide the drug design process. LINCS L1000 dataset provides a vast resource of cell line gene expression data perturbed by different drugs and creates a knowledge base for context specific features. The state-of-the-art approach that aims at using context specific information relies on only the high-quality experiments in LINCS L1000 and discards a large portion of the experiments. In this study, our goal is to boost the prediction performance by utilizing this data to its full extent. We experiment with 5 deep learning architectures. We find that a multi-modal architecture produces the best predictive performance among multi-layer perceptron-based architectures when drug chemical structure (CS), and the full set of drug perturbed gene expression profiles (GEX) are used as modalities. Overall, we observe that the CS is more informative than the GEX. A convolutional neural network-based model that uses only SMILES string representation of the drugs achieves the best results and provides 13.0% macro-AUC and 3.1% micro-AUC improvements over the state-of-the-art. We also show that the model is able to predict side effect-drug pairs that are reported in the literature but was missing in the ground truth side effect dataset. DeepSide is available at http://github.com/OnurUner/DeepSide.

Index Terms—Drug side effect prediction, deep learning, LINCS

1 Introduction

Computational methods hold great promise for mitigating the health and financial risks of drug development by predicting possible side effects before entering into the clinical trials. Several learning based methods have been proposed for predicting the side effects of drugs based on various features such as: chemical structures of drugs [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], drug-protein interactions [4], [5], [6], [7], [9], [12], [13], [14], [15], [16], protein-protein interactions (PPI) [4], [8], activity in metabolic networks [17], [18], pathways, phenotype information and gene annotations [4]. In parallel to the above mentioned approaches, recently, deep learning models have been employed to predict side effects: (i) [19] uses biological, chemical and semantic information on drugs in addition to clinical notes and case reports to train a fully connected multi-layered perceptron, and (ii) [20] uses chemical fingerprints and learns to predict side effects using a convolutional neural network architecture.

While these methods have proven useful for predicting adverse drug reactions (ADRs - used interchangeably with drug side effects), the features they use are solely based on external knowledge about the drugs (i.e., drug-protein interactions, etc.) and are not cell or condition (i.e., dosage) specific. To address this issue, Wang *et al.* (2016) utilize the data from the LINCS L1000 project [21]. This project profiles gene expression changes in numerous human cell lines after treating them with a large number of drugs and small-molecule compounds. By using the gene expression profiles of the treated cells, [21] provides the first comprehensive, unbiased, and cost-effective prediction of ADRs. The paper formulates the problem as a multilabel classification task. Authors train an *Extra Trees* classifier for this purpose. This is a tree-based ensemble method that strongly randomizes the feature and cut-point choices while constructing the trees [22]. Their results suggest that the gene expression profiles provide context-dependent information for the side-effect prediction task. While the LINCS dataset contains a total of 473,647 experiments for 20,338 compounds, their method utilizes only the highest quality experiment for each drug to minimize noise. This means that most of the expression data are left unused, suggesting a potential room for improvement in the prediction performance. Moreover, their framework performs feature engineering by transforming gene expression features to enrichment vectors of biological terms. In this work, we investigate whether the incorporation of gene expression data along with the drug structure data can be leveraged better in a deep learning setting, which is potentially more complex and does not require feature engineering.

In this study, we propose a deep learning based approach, DeepSide, for ADR prediction. DeepSide uses only (i) *in vitro* gene expression profiling experiments (GEX) and their experimental meta data (i.e., cell line and dosage - META), and (ii)

- *Onur Can Uner and Halil Ibrahim Kuru are with the Computer Engineering Department, Bilkent University, 06800 Ankara, Turkey. E-mail: {can.uner, ibrahim.kuru}@bilkent.edu.tr.*
- *R. Gokberk Cinbis is with the Computer Engineering Department, Middle East Technical University, 06800 Ankara, Turkey. E-mail: gcinbis@ceng.metu.edu.tr.*
- *Oznur Tastan is with the Faculty of Engineering and Natural Sciences, Sabanci University, 34956 Istanbul, Turkey. E-mail: otastan@sabanciuniv.edu.*
- *A. Ercument Cicek is with the Computer Engineering Department, Bilkent University, 06800 Ankara, Turkey, and also with the Computational Biology Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA. E-mail: cicek@cs.bilkent.edu.tr.*

Manuscript received 25 Sept. 2020; revised 6 Nov. 2021; accepted 4 Jan. 2022. Date of publication 7 Jan. 2022; date of current version 3 Feb. 2023.
This work of A. Ercument Cicek was supported in part by TUBA - GEBIP 2017 Award, in part by Bilim Akademisi BAGEP 2020 Award, and in part by TUSEB Aziz Sancar Reseach Incentive 2020 Award.
(Corresponding authors: Oznur Tastan and A. Ercument Cicek.)
Digital Object Identifier no. 10.1109/TCBB.2022.3141103

the chemical structure of the compounds (CS). Our models train on the full LINCS L1000 dataset and use the SIDER dataset as the ground truth for drug - ADR pair labels [23]. We experiment with five architectures: (i) a multilayer perceptron (MLP), (ii) MLP with residual connections (ResMLP), (iii) multi-modal neural networks (MMNN.Concat and MMNN.Sum), (iv) multi-task neural network (MTNN), and finally, (v) SMILES convolutional neural network (SMILESConv). Please see Section 2.3 for details about the possible advantages of these approaches.

We present an extensive evaluation of the above-mentioned architectures and investigate the contribution of different features. Our experiments show that CS is a robust predictor of side effects. The base MLP model, which uses CS features as input, produces ~11% macro-AUC and ~2% micro-AUC improvement over the state-of-the-art results provided in [21], which uses both GEX (high quality) and CS features. The multi-modal neural network model, which uses CS, GEX and META features and uses summation in the fusion layer (MMNN.Sum), achieves 0.79 macro-AUC and 0.877 micro-AUC which is the best result among MLP based approaches. We also find out that when the chemical structure features are fully utilized in a complex model like ours, it overpowers the information that is obtained from the GEX dataset. The convolutional neural network that only uses the SMILES string representation of the drug structures achieves the best result among all the proposed architectures with provides 13.0% macro-AUC and 3.1% micro-AUC improvement over the state-of-the-art algorithm. Finally, inspecting the confident false positives predictions reveal side effects that are not reported in the ground truth dataset, but are indeed reported in the literature.

Our study has several novel aspects and our contributions can be summarized as follows. First, this is the first study to employ rich deep learning models on large scale experimental gene expression data along with drug structure information to predict drug side effects. While many studies in the literature opt to utilize various other data sources such as drug-drug or drug-protein interactions, this information is not available for many compounds. This minimal feature requirement enables our model to work with under-studied compounds with little or no background information. Second, we develop deep learning models that use state-of-the-art neural network architectural blocks for the first time to solve the drug side effect prediction problem. These models include multi-task learning, multi-modal learning, residual networks and convolutional neural networks. The most successful model uses 1D convolution operation on SMILES strings of the drugs. In contrast to the common approach in the literature which is using a small number of fixed-sized filters, we find that using many and highly-varying-sized filters to learn the relation between local (short) / global (long) structural motives is highly effective for the prediction of the side effects. Finally, we find that once utilized with a complex model like convolutional neural networks, drug structure (SMILES representations) is the most informative source of information for this task. DeepSide is implemented and released at http://github.com/OnurUner/DeepSide.

2 Methodology

2.1 Problem Formulation

The problem of side effect prediction is modelled as a multilabel classification task. For a given drug i, the target label is a binary vector, $\mathrm{y}_i = [y_{i,1}, y_{i,2}, \ldots, y_{i,d}]$, where d is the number of side effects and $y_{i,j} = 1$ indicates that the drug i has side effect j; $y_{i,j} = 0$ indicates otherwise. Our dataset contains n samples (drugs), each represented by a pair of drug feature vector $\mathrm{x_i}$ and an accompanying side effect vector (classes) y_i: $(\mathrm{x}_i, \mathrm{y}_i)_{i=1}^{n}$.

2.2 Datasets

The LINCS L1000 dataset (GSE92742) contains the GEX profiles of 76 cell lines, treated with 20,413 small-molecule compounds [24]. There are 473,647 signature experiments that differ by the dosage, timing, and cell line (Level 5 data). In each experiment, the expression levels of 978 landmark genes are recorded. The study has two development phases: Phase 1 and Phase 2. Phase 1 contains approved drugs, whereas Phase 2 contains drugs that are at an experimental stage. To be able to compare our results with those in [21], we use Phase 1 data and process the dataset in the same manner. The authors report that their best result is obtained with the feature set that is a combination of gene ontology (GO) transformed gene expression profiles and chemical structures (CS). Their set of drugs with this feature set (GO + CS) contains 791 compounds. We use these 791 drugs to build our models. In total, there are 18,832 experiments for these 791 drugs in the LINCS L1000 dataset.

The META information for each of the 18,832 experiments from the LINCS project is also used as features. META information contains (i) the cell line on which the experiment is conducted on, (ii) the timing of the experiment, and (iii) dosage information. The meta information exists for 70 cell lines, 20 dosage levels and 3 time points (i.e., 6h, 24h, 48h). Note that for a given drug, the experiments do not cover all possible combinations of these conditions. META data is represented as one-hot encoding vectors. The corresponding feature vector has a length of 93. The total length of the concatenated GEX and META feature vectors is 1071. For all models, whenever META data is used, it is concatenated with the 978 landmark GEX features.

We obtain the drug side effect information (labels) from the SIDER Database [23] (downloaded on Feb 5, 2018). The side effects that are observed with fewer than ten drugs are excluded as also done in [21]. This filtering stage leaves us with 1052 side effects in total. In order to group side effects, we utilize the ADR ontology database (ADReCS), which provides a hierarchical classification of side effects in a four-level tree [25].

The CS features are encoded with OpenBabel Chemistry Toolbox [26] to create a 166-bit MACCS chemical fingerprint matrix for each drug (a binary vector of length 166). A SMILES string is an alternative representation for the 2D molecular graph of a drug/small molecule as a 1D string. The SMILES strings are downloaded from PubChem [27]. These are used to create the chemical fingerprints of the drugs for the 1D convolution used in SMILESConv model. RDKit Cheminformatics toolbox is used to extract extended SMILES Strings of the drugs [28]. The extended SMILES

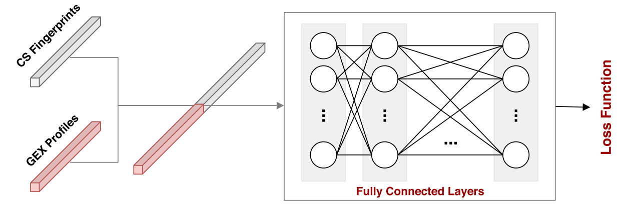


Fig. 1. Our multi-Layer perceptron (MLP) architecture, which takes the concatenation of GEX and CS features.

strings contain all the primary chemical bonds as well as the hydrogen bonding information explicitly. Zero-padding is used to have a uniform representation among all drugs. The alphabet contains 33 unique characters, including the end of sequence character. We further generate a pruned drug dataset to compare SMILESConv model with others. We filter out drugs with SMILES representation that have less than 100 characters and more than 400 characters. 615 out of 791 drugs pass this filtering step. For these drugs, we apply the additional filtering for removing side effects with less than ten drugs. In the end, 615 drugs and 1042 side effects pairs remain in this pruned dataset. Finally, we remove the characters that occur only once in all SMILES strings from the character vocabulary and replace them with underscore symbol.

2.3 The DeepSide Architectures

We propose the following deep learning architectures for ADR prediction: (i) a simple multi-layer perceptron, (ii) its residual variant, (iii) multi-modal network architectures that pre-transform inputs from each domain separately, (iv) multi-task neural network, and finally, (v) a convolutional neural network based approach for incorporating SMILES representation.

2.3.1 Multi-Layer Perceptron (MLP)

Our MLP [29] model takes the concatenation of all input vectors and applies a series of fully-connected (FC) layers. Each FC layer is followed by a batch normalization layer [30]. We use ReLU activation [31], and dropout regularization [32] with a drop probability of 0.2. The sigmoid activation function is applied to the final layer outputs, which yields the ADR prediction probabilities. The loss function is defined as the sum of negative log-probabilities over ADR classes (i.e., the multi-label binary cross-entropy loss (BCE)). An illustration of the architecture for CS and GEX features is given in Fig. 1.

2.3.2 Residual Multi-Layer Perceptron (ResMLP)

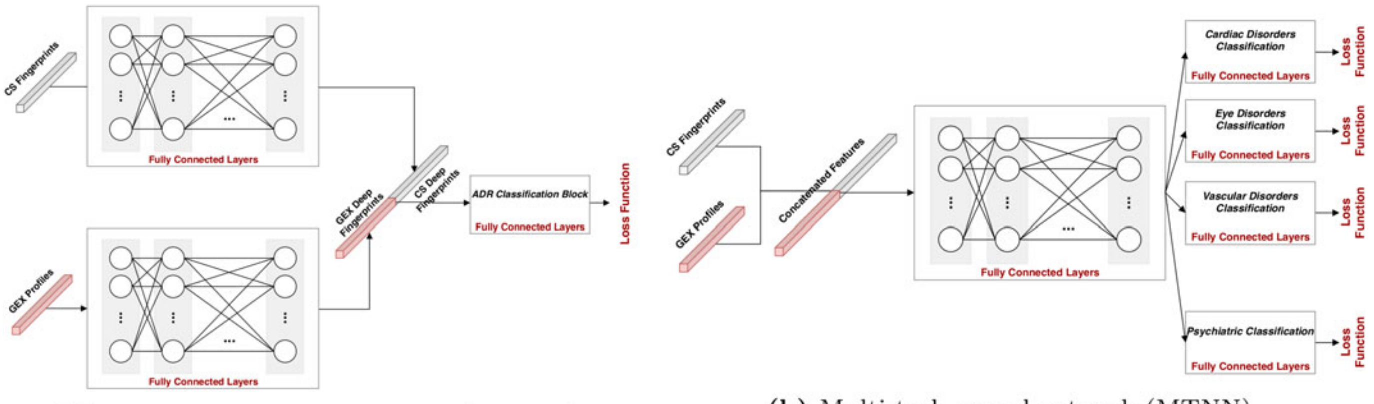
The residual multi-layer perceptron (ResMLP) architecture is very similar to MLP, except that it uses residual-connections across the fully-connected layers. More specifically, the input of each intermediate layer is element-wise added to its output, before getting processed by the next layer. Such residual connections have been shown to reduce the vanishing gradient problem to a large extend [33]. This effectively allows deeper architectures, therefore, potentially learning more complex and parameter-efficient feature extractors.

2.3.3 Multi-Modal Neural Networks (MMNN)

The multi-modal neural network approach contains distinct MLP sub-networks where each one extract features from one data modality only. The outputs of these sub-networks are then *fused* and fed to the classification block. For feature fusion, we consider two strategies: concatenation and summation. While the former one concatenates the domain-specific feature vectors to a larger one, the latter one performs element-wise summation. By definition, for summation based fusion, the domain-specific feature extraction sub-networks have to be designed to produce vectors of equivalent sizes. We refer to the concatenation and summation based MMNN networks as MMNN.Concat and MMNN.Sum, respectively. The MMNN.Concat approach is illustrated in Fig. 2.

2.3.4 Multi-Task Neural Network (MTNN)

Our multitask learning (MTL) based architecture aims to take the side effect groups obtained from the taxonomy of ADReCS into account. For this purpose, the approach defines



(a) Multi-modal neural networks (MMNN)

(b) Multi-task neural network (MTNN)

Fig. 2. Multi-modal and Multi-task Neural Network architectures. (a) The concatenation variant of the multi-modal neural network (MMNN.Concat) architecture, which has two input branches for the GEX and CS features. The outputs of these networks are concatenated and fed into a fully connected multi-layer classification block. b) The multi-task neural network (MTNN) architecture, which learns a shared embedding for all class groups in the shared layers. The embedding is then fed into separate fully-connected multi-layer classification blocks for each class group, which learn task specific models.

shared and *task-specific* MLP sub-network blocks. The shared block takes the concatenation of GEX and CS features as input and outputs a joint embedding. Each task-specific sub-network then converts the joint embedding into a vector of binary prediction scores for a set of inter-related side-effect classes.

We define 24 side-effect groups according to the ADR ontology (see Section 2.2). Here, a side effect is allowed to be a member of multiple groups. For instance, in the ADR ontology, *nausea* is grouped under both *stomach disorders* and *dizziness* sub-groups. For such side effects, our model will output more than one probability estimate. The maximum estimate among multiple predictions for such cases is taken as the final prediction, during both training, (i.e., when computing the log-loss), and testing. The architecture is illustrated in Fig. 2.

2.3.5 SMILES Convolutional Network (SMILESConv)

Convolutional neural networks (CNN) are known to provide a powerful way of automatically learning complex features in vision tasks, see e.g., [34]. More recently, convolutional networks have also been shown to be effective for modeling sequential data, such as natural texts, see e.g., [35]. Our SMILESConv architecture is built upon 1D convolutional operators for representation learning on the SMILES strings. In this case, the kernels are vectors and they learn to leverage the relations across the consecutive characters. [36] and [37] use convolutional and recurrent networks for learning vector space embeddings of SMILES strings for solving other prediction tasks.

Our network contains 200 1D-convolutional layers where the kernel sizes range from 1 to 200. Each layer has 32 output channels, which are followed by batch normalization [30]. We use ReLU activation function and max-pooling operators. The size of the pooling operations is equal to size the feature map that has been extracted after convolution, batch normalization, and ReLU operations. Each vector is concatenated to pass through classification layers. The extracted feature vector has 6400 units (32x200). We use dropout with a drop probability of 0.2 before the fully connected classification layers. The classification block contains 2000 units. Batch normalization and ReLU activation follow each fully connected layer. The sigmoid activation function is applied to the output layer. The overall SMILESConv architecture is shown in Fig. 3.

3 Results

3.1 Experimental Setup

We use 3-fold cross-validation to evaluate our models; the folds are stratified based on drugs. That is, all experiments of a single drug are either completely in the training set or completely in the test set, and therefore, a model is expected to predict the side-effects of previously unseen drugs at test time. To accomplish a fair comparison among models, we use 6 different data settings. The first 3 settings consider 791 drugs and are used to train and test only the MLP based models. The first setting uses all ~18k experiments conducted for the 791 drugs in different cell lines, dosages and time points. In this setting, each instance is an experiment for a drug and can accompany chemical structure information. The training data contains ~12k instances, while the test data contains ~6k instances. The second setting covers only the *highest quality* experiment for each of the 791 drugs, as marked in the meta-data of the LINCS L1000 dataset. Again, each instance is an experiment for a drug. The training data contains 528 instances and the test data contains 263 instances. Note that this setting is the same as the one used in [21]. The third setting uses a mixture of the first two ones: ~12k instances are used for training, and 263 highest quality experiments are used for testing.

The last three settings use the 615 drugs (out of 791) which are selected according to the SMILES string criteria described in Section 2.2. To make a fair comparison between the SMILESConv and MLP based models, we re-evaluate the MLP based models in these settings and choose the best performing one to compare against SMILESConv. The fourth setting uses only the CS or SMILES string features and uses 410 samples for training and 215 samples for testing. The fifth setting uses ~9K experiments from the GEX dataset for the 410 drugs used for training and ~4K experiments for the 205 drugs for testing. Again, each instance is an experiment for a drug and can accompany CS information. The sixth setting also uses ~9K experiments from the GEX dataset for the 410 drugs like but the test data includes only the highest quality experiments of the 205 drugs.

We use binary cross entropy (BCE) as the loss function. We investigate the benefit of employing weighted BCE (WBCE) on the SMILESConv model to address the imbalance in our dataset (i.e.,, some side effects are observed rarely.) Adam optimizer is used for training the neural networks. While the initial learning rate for Adam optimizer is tuned separately for each model and dataset pair, the same set of hyper-parameters is used across the folds.

The hyperparameters of the MLP models are decided based on the input dimension and the number of the side effects in the label set. We varied the number of layers from 1 to 5. The number of neurons are searched in the range 200–2000. The learning rate parameter was changed in the range 0.01 and 0.0001. We report the results of the hyperparameters that yields distinct performance results in Table 1. We determine the shape and number of kernels for the SMILESConv model based on the maximum and minimum sizes of the input sequence. We observe that even when we use fewer number of kernels (i.e., 20 kernels with shapes in range 0 to 20), SMILESConv's performance does not degrade. As the number of kernels increases, on the other hand, we would expect a minor increase in performance along with a significant increase in complexity. To sum it up, we have tried to select hyperparameters within a sensible parameter range, with due regard to performance & complexity trade-off.

To assess how well we predict the side-effects of drugs overall, we use the micro-averaged Area Under Curve (AUC), micro-averaged mean Average Precision (mAP) and Hamming loss metrics. To evaluate per side effect prediction performance, we use the macro-averaged Area Under Curve (AUC) and macro-averaged mean Average Precision (mAP) metrics.

3.2 Performances of DeepSide Architectures

We present MLP-based model results in Table 1. Our first finding is that the base MLP model that uses only the CS

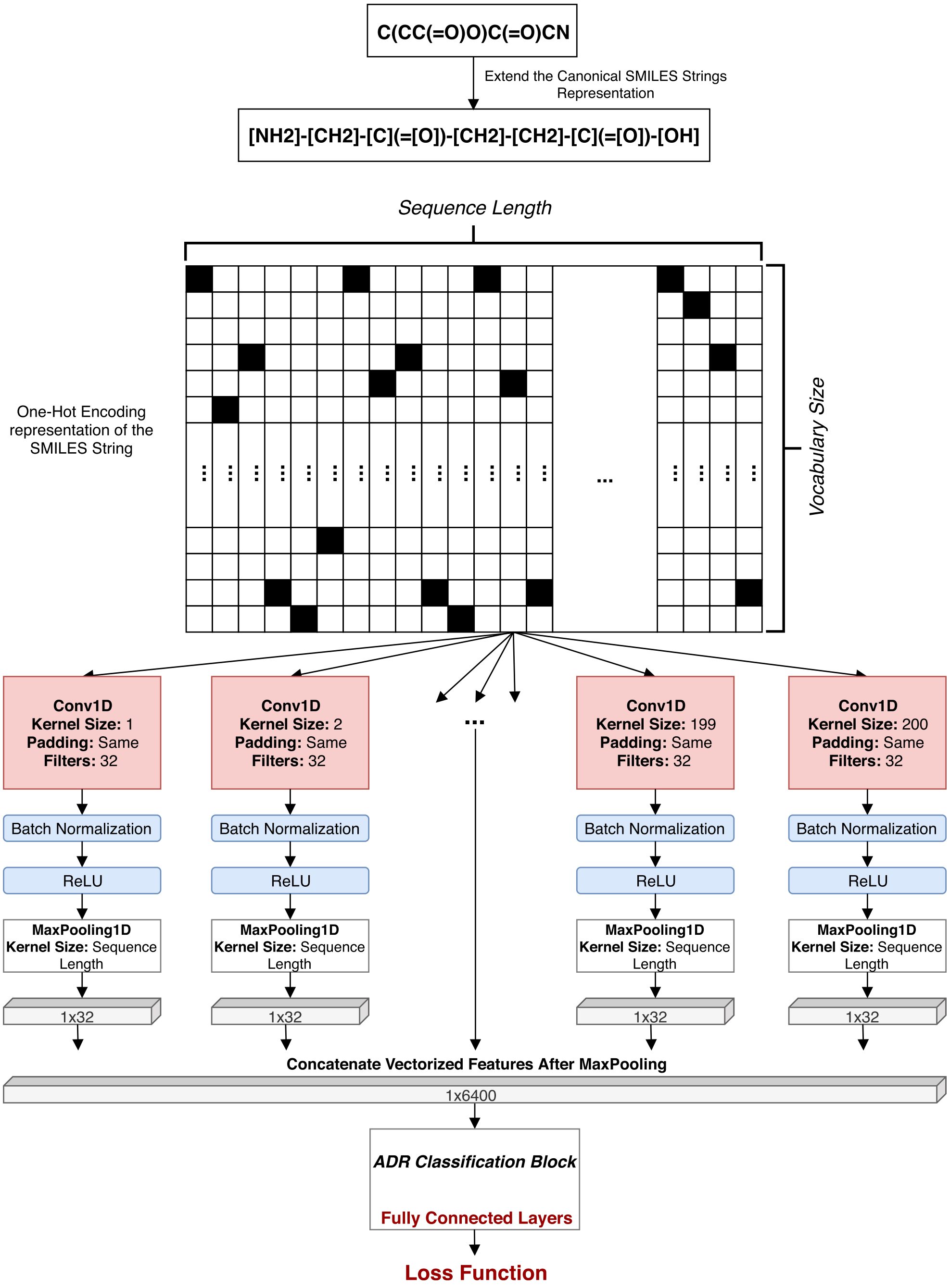

C(CC(=O)O)C(=O)CN
Extend the Canonical SMILES Strings Representation
[NH2]-[CH2]-[C](=[O])-[CH2]-[CH2]-[C](=[O])-[OH]
Sequence Length
One-Hot Encoding representation of the SMILES String
Vocabulary Size
Conv1D Kernel Size: 1 Padding: Same Filters: 32
Conv1D Kernel Size: 2 Padding: Same Filters: 32
Conv1D Kernel Size: 199 Padding: Same Filters: 32
Conv1D Kernel Size: 200 Padding: Same Filters: 32
Batch Normalization
ReLU
MaxPooling1D Kernel Size: Sequence Length
1x32
Concatenate Vectorized Features After MaxPooling
1x6400
ADR Classification Block
Fully Connected Layers
Loss Function

Fig. 3. The SMILESConv architecture that performs 1D convolution operations on the SMILES representations of drugs. Fused embeddings are fed into a fully connected multi-layer classification block.

fingerprints outperforms the state of the art model [21], which uses the same CS fingerprints along with the GO-transformed GEX dataset, in terms of both micro and macro AUC scores. Note that this comparison is based on the same set of drugs and side effects.

The MLP model based purely on GEX features yields the lowest scores in both Settings 1 and 3 (Table 1), the macro-AUC is at most 67% and macro-AUC is 80%. This indicates that GEX features alone are not sufficiently informative for side effect prediction. When we combine GEX and CS features through concatenation and input to the MLP model, the performance increases to 76.8% macro AUC and 84.1% micro AUC scores (Setting 3; similar for Setting 1), which are still below the performance of the MLP model trained only with the CS features. In fact, when we investigate the feature importance for the MLP model that takes the concatenated GEX, CS and META as input, we find that top-100 of the most important are all CS related features, see Section 3.3 for details.

We further input different combinations of CS and GEX features into various machine learning algorithms in order to check if our observation that CS is more informative than the GEX features. We used all possible feature combinations for each model (i.e., only CS, only GEX, CS + GEX). We used three different machine learning algorithms, namely Logistic Regression, XGBoost, and Random Forest.

We explore different hyperparameter combinations to train these models on the same train test split we used for

TABLE 1
Performance Comparison Between MLP Models That Use GEX, CS and META Information

Model	Features	# train	# test	FC neurons	layers	Macro AUC	Micro AUC	Macro mAP	Micro mAP	Hamming
Wang et al. [21]	GO+CS	528	263	-	-	0.679	0.854	-	-	0.083
MLP	CS	528	263	800	3 - 1	0.784	0.866	**0.457**	0.578	0.072
MLP	CS	528	263	2000	3 - 1	0.781	0.860	0.454	0.557	0.075
ResMLP	CS	528	263	800	101 - 1	0.768	0.843	0.428	0.520	0.077
MLP	GEX	12K	6K	800	3 - 1	0.621	0.781	0.382	0.491	0.203
			263			0.674	0.801	0.401	0.498	0.176
MLP	[GEX, CS]	12K	6K	2000	5 - 1	0.761	0.838	0.404	0.538	0.089
			263			0.768	0.841	0.411	0.541	0.081
MLP	[GEX, CS, META]	12K	6K	2000	5 - 1	0.767	0.845	0.421	0.558	0.086
			263			0.774	0.844	0.426	0.528	0.076
MLP	[GEX, CS, META]	528	263	2000	5 - 1	0.727	0.832	0.390	0.497	0.089
ResMLP	[GEX, CS, META]	12K	6K	2000	5 - 1	0.760	0.857	0.422	0.577	0.084
			263			0.771	0.856	0.428	0.547	0.075
MTNN	[GEX, CS, META]	12K	6K	2000	5 - 1	0.759	0.841	0.401	0.511	0.087
			263			0.772	0.851	0.418	0.522	0.079
MMNN.Sum	CS & [GEX,META]	12K	6K	800 - 800	3 - 1	0.772	0.871	0.435	0.600	0.081
			263			**0.790**	**0.877**	**0.457**	0.592	**0.070**
MMNN.Concat	CS & [GEX,META]	12K	6K	800 - 800	3 - 1	0.768	0.868	0.436	0.598	0.080
			263			0.787	0.875	0.460	0.586	0.072
MMNN.Sum	CS & GEX	12K	6K	800 - 800	3 - 1	0.764	0.868	0.431	**0.602**	0.081
			263			0.779	0.872	0.445	0.582	0.071
MMNN.Sum	CS & GEX	12K	6K	2000 - 2000	3 - 1	0.772	0.864	0.440	0.588	0.082
			263			0.783	0.867	0.444	0.569	0.073
MMNN.Sum	CS & GEX	528	263	2000 - 2000	3 - 1	0.772	0.863	0.424	0.557	0.075

X&Y represents the independent two datasets that are used as inputs for the MMNN architecture. X is an input for one of the branches and Y is the input for the other branch of the MMNN-based models. $[X, Y]$ represents the concatenated features of the X and Y datasets. FC neurons *column denotes neuron size in the fully connected layers.* layers *column states the number of fully connected layers in the feature extractor and classification parts of the network.*

DeepSide architectures. For XGBoost model, we choose the best performing model by using the learning rate $\in \{0.05, 0.10, 0.15\}$ and maximum tree depth $\in \{4, 6, 8, 10, 12\}$. For the Random Forest model, we tune the number of trees $\in \{10, 100, 500, 1000\}$ and maximum tree dept $\in \{4, 6, 8, 10, 12\}$. Finally, for the Logistic Regression model, the regularization parameter C $\in \{0.25, 0.50, 0.75, 1.0\}$ is tuned. As the results in Table 2, the models that use only CS features result in better performance without an exception. These results align well with the earlier experimental results.

The ResMLP architecture, which uses residual connections across the fully connected layers does not improve upon the base MLP model. MTNN, which aims to leverage the side effect group information based on the side effect ontology, does not improve over the base MLP model either. On the other hand, the MMNN model, which uses two modalities (one for the concatenated GEX profiles and META information and the other for the CS fingerprints), produces the best predictive performance among all MLP-based architectures in terms of all metrics, with the exception of micro mean average precision (micro mAP). This architecture achieves 0.111 macro AUC improvement and 0.023 micro AUC improvement over state of the art in Setting 3 when summation based embedding fusion is used. Concatenation based fusion yields similar results. MMNN is the only architecture that benefits from adding GEX features on to the CS features. Since we consistently obtain very similar or better results by incorporating the META information, we exclude the results of some of the models without META features for brevity.

Setting 2 only uses the highest quality experiments (as in [21]), whereas Setting 3 uses the all experiments for a compound during training. For testing, both settings use the highest quality experiments. Here, we validate our hypothesis that a deep learning based solution should be able to perform better by utilizing the full dataset in Setting 3. First, we compare the performance of the MLP model under Setting 2 and Setting 3 (using GEX, CS, and META features): using Setting 3 provides 4.7% macro AUC and 1.2% micro AUC, 3.1% macro mAP and 3.6% micro mAP improvement over Setting 2. We also compare the performance of the best MLP-based model (MMNN.Sum) under

TABLE 2
The Results of Off-the-Shelf Machine Learning Algorithms When Using CS, GEX and CS & GEX Features

Model	Features	Macro AUC	Micro AUC	Macro mAP	Micro mAP	Hamming
Logistic Regression	CS	**0,576**	**0,645**	**0,194**	**0,237**	0,089
	GEX	0,501	0,569	0,095	0,169	0,089
	[CS, GEX]	0,570	0,643	0,186	0,235	0,089
XGBoost	CS	**0,660**	**0,714**	**0,273**	**0,315**	**0,083**
	GEX	0,506	0,586	0,101	0,151	0,108
	[CS, GEX]	0,511	0,592	0,106	0,159	0,106
Random Forest	CS	**0,778**	**0,884**	**0,464**	**0,600**	**0,070**
	GEX	0,536	0,800	0,131	0,365	0,090
	[CS, GEX]	0,587	0,811	0,157	0,376	0,089

The results confirm our observation that CS is the most informative modality.

TABLE 3
Performance Comparison Between MLP and Conv Models Which are Trained With 615 Drugs for the 1042 Side Effects

Model	Feature Set	# train	# test	Macro AUC	Micro AUC	Macro mAP	Micro mAP	Hamming
MLP	CS	410	205	0.788	0.849	0.484	0.577	0.080
MMNN.Sum	CS & [GEX, META]	9K	4K	0.779	0.841	0.465	0.562	0.088
			205	0.794	0.852	0.485	0.579	0.079
MMNN.Sum.SMILESConv	SMILES String & [GEX, META]	9K	4K	0.774	0.854	0.471	0.588	0.082
			205	0.787	0.866	0.486	0.557	0.074
SMILESConv (BCE)	SMILES String	410	205	0.805	0.876	0.493	0.594	**0.074**
SMILESConv (WBCE)	SMILES String	410	205	**0.809**	**0.885**	**0.501**	**0.601**	0.082

$[X, Y]$ represents the concatenated features of the X and Y datasets. $[X]\&[Y]$ represents the two separate datasets applied different braches of the MMNN-based models. BCE denotes binary cross entropy and WBCE denotes the weighted binary cross entropy.

these two settings using the CS and GEX features. Indeed, setting 3 provides 1.1% macro AUC, 0.4% micro AUC, 2.4% macro mAP and 1.2% micro mAP improvement over Setting 2. While the margin of improvement is smaller for the more complex model, both results show the benefit of using all experiments in the LINCS L1000 dataset.

We investigate the benefit of using SMILES strings for representing drug structures and employing convolutional neural networks to extract features on them. Table 3 shows the results of SMILESConv models that are trained with unweighted (BCE) and class weighted loss (WBCE) functions. To make a fair comparison to the SmilesConv models, we retrain separate MLP and MMNN.Sum architectures with datasets of Settings 4 - 6. In SMILESConv models, cost-sensitive training with WBCE improves the results compared to training with BCE; all performance measures are higher for WBCE except for the hamming loss. SMILESConv outperforms both the MMNN.Sum and the MLP based model; with WBCE, it achieves 0.809 macro AUC and 0.885 micro AUC. This corresponds to about 2.1% improvement in macro AUC and 3.6% improvement in micro AUC compared to the MLP model that uses only the CS structures. It also improves upon the MMNN.Sum about 1.5% in macro AUC and 3.3% in micro AUC. Similar improvements are observed for the other performance metrics MAP and Hamming loss. The predicted probabilities by SMILESConv WBCE for every compound - side effect pair are listed in Supplementary Table 1, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2022.3141103.

We further investigate whether uniting the GEX and META features with SMILES strings improves the performance. We train a new MMNN.Sum model in which we replace the chemical fingerprint representation (CS) with the SMILES representation. We observe that this improves MMNN.Sum model's performance but cannot outperform SMILESConv only model (Table 3).

We also compared SMILESConv with a recent method from the literture that also utilizes drug chemical structure, called MEDICASCY [11]. To perform the comparison, we used our dataset which is arranged according to SMILESConv preprocessing rules detailed in Section II-B (615 drugs for the 1042 side effects) and used the 166-bit MACCS features to train MEDICASCY-MACCS-BRF method. We used exactly the same 3-fold to compare MEDICASCY with our approaches.

We observe that SMILESConv achieves 0.809 macro AUC and 0.885 micro AUC, whereas MEDICASY achieves 0.781 macro AUC and 0.885 micro AUC. While the performances are close we conclude that SMILESConv is better.

3.3 Feature Importance

We investigate the feature importance for the MLP model that takes the concatenated GEX, CS and META as input using Deep SHAP algorithm, which is a method to compute SHAP values [38] of deep learning models. The method relies on Shapley values [39] and assess the contribution of each feature to prediction by comparing the predictive performance of all models with and without the corresponding feature. We quantify the feature importance at each fold separately and use the average feature importance computed over three folds to rank features. We observe all top-100 features are CS related. Of the top-200 features, 140 of them are CS related features, while 42 of them are GEX and remaining 18 are META features. We list the top 20 features obtained in Fig. 4. The feature importance for each feature is provided in Supplementary Table 1, available online.

4 Discussion

We investigate the easiest (top-10) and the hardest (bottom-10) side effects to predict by the SMILESConv model (WBCE) in Table 4. For both cases, these side effects have less than 100 drugs. Although there is no clear pattern, we observe that the easy examples are relatively more specific compared to the hard examples (i.e., Myocardial rupture, Lupus miliaris disseminatus faciei, and Paraplegia are examples for easiest side effects; while Ear disorder, Personality Disorder and Sensory disturbance are examples for hardest side effects ones). Based on their presence in drugs, we calculate the correlation coefficients (MCC) of the side effects. We do not observe any correlation among most-difficult-to-predict side effect pairs. However, of the 45 possible side effect

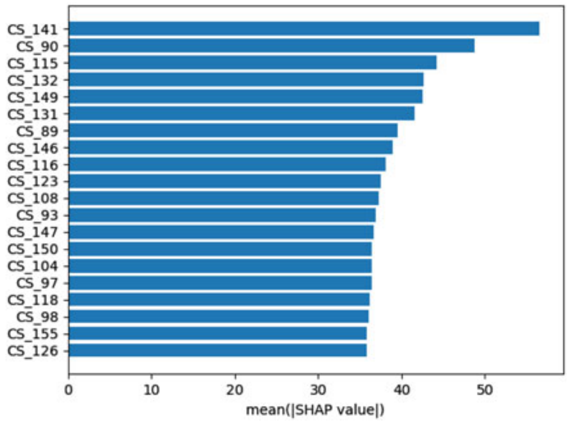


Fig. 4. The most important top-20 features between [GEX&CS&META] features for the MLP model. Y axis shows the index of the given feature type. All top-20 features are CS features.

TABLE 4
The Easiest (Top-10) and the Hardest (Bottom-10) Side Effects to Predict by the SMILESConv Model Trained With the Weighted Binary Cross-Entropy Loss

Side Effect	# Pos. Samples	SMILESConv AUC
Skin test positive	21	1.00
Cushing's syndrome	12	1.00
Myocardial rupture	19	1.00
Alkalosis hypokalaemic	21	1.00
Fat embolism	15	1.00
Muscle mass	21	1.00
Coombs direct test positive	10	1.00
Paraplegia	17	1.00
Lupus miliaris disseminatus faciei	23	1.00
Nitrogen balance	23	1.00
Skin burning sensation	7	0.54
Panic attack	9	0.55
Tachypnoea	10	0.64
Sensory disturbance	8	0.50
Hepatitis fulminant	11	0.58
Ear disorder	28	0.57
Arrhythmia supraventricular	15	0.65
Respiratory disorder	87	0.64
Personality disorder	26	0.62
Congenital eye disorder	11	0.62

Number of positive samples column indicates the number of drugs annotated with a given side effect.

pairs among 10 side effects, in 22 of them, we observe positive correlation (MCC > 0.5). From the easiest side effects, 'Skin test positive', 'Alkalosis hypokalaemic' and 'Muscle mass' side effects are the most positively correlated side effects with an MCC value of 1.0 between each other.

We also investigate our most confident but incorrect predictions. Table 5 shows the top-10 false positive and top-10 false negative predictions. For the following false positive examples, we find evidence in the literature that the predicted side effects might be relevant. Daunorubicin, which is a chemotherapeutic compound, is predicted to cause anemia by DeepSide. Chemotherapy-induced anemia is a common side effect in cancer patients [40]. In particular for this drug, Hazardous Substances Data Bank[1] (a toxicology database curated by NIH NLM Toxicology Network) lists anemia as a possible adverse reaction for daunorubicin.[2] Similarly, we find that sulfasalazine (a drug used to treat rheumatoid arthritis and ulcerative) causes vomiting. This finding is supported by [41], which reports that 64 out of 152 people developed adverse reactions due to this drug, and 19 out of that 64 had vomiting. Finally, our model predicts halcinonide to cause hypertension. Halcinonide is a corticosteroid that is used to treat various skin conditions. It is a glucocorticoid and [42] lists hypertension as an adverse effect for glucocorticoids. Note that none of the above findings are reported in SIDER. We also find support for 9 out of the top 10 false positives through commercial online resources. Nevertheless, it is hard to assess the reliability as there is no peer review system. While it is harder to evaluate false negatives, we find that rather than (i) doxycycline causing premenstrual syndrome, and (ii) cyproterone causing leiomyoma; they are used in the treatment of these conditions [43], [44]. For the rest of the findings we see that there are indications in the literature and commercial online resources that these compounds cause corresponding side effects.

TABLE 5
The top-10 False Positive (Top Table) and Top-10 False Negative Predictions (Bottom Table) of the SMILESConv WBCE Model

Perturbation ID	Compound Name	Side Effect	# Pos. Samples
BRD-A37630846	daunorubicin	Anaemia	326
BRD-K13926615	vardenafil	Anaemia	326
BRD-K10670311	sulfasalazine	Vomiting	476
BRD-K19352500	prochlorperazine	Vomiting	476
BRD-K28029915	dolasetron	Vomiting	476
BRD-K32164935	tolazamide	Vomiting	476
BRD-K71451869	halcinonide	Hypertension	293
BRD-K81709173	halcinonide	Hypertension	293
BRD-K81774264	flumethasone	Pain	475
BRD-K81925854	clocortolone	Pain	475
BRD-A39290993	cyproterone	Leiomyoma	11
BRD-A51294525	cyproterone	Leiomyoma	11
BRD-K05395900	nicotine	Nasal ulcer	14
BRD-K11196887	norfloxacin	Metabolic acidosis	16
BRD-A73635141	hydrocortisone	Menstrual disorder	53
BRD-A73635141	hydrocortisone	Application site reaction	26
BRD-A74980173	gatifloxacin	Panic attack	9
BRD-A79479878	testosterone	Sleep apnea syndrome	10
BRD-A88774919	doxycycline	Osteopenia	12
BRD-A88774919	doxycycline	Premenstrual syndrome	13

These are the most confident predictions by the model that are contradicting with the ground truth. For the listed false positive pairs, predicted probabilities are > 0.9995. For the false negative pairs, predicted probability scores are < 0.0005. Note that a drug (name) might have multiple perturbagen id that correspond to different SMILES strings. In that case drug name - side effect pairs are listed multiple times.

The LINCS L1000 dataset is a useful resource for predicting condition specific side effects. In our experiments though, we find the GEX does not improve the results substantially (see Tables 1 and 3) and the best performing model that relies on the drug structure and surpasses the state-of-the-art performance [21] (see Tables 1 and 3). One reason for not being able to leverage condition specific GEX information despite employing various deep learning architectures could be the absence of the condition specific ground truth labels. Since the available side effect labels are per drug but not per condition-drug pairs (i.e., dosage - drug), we suspect the model cannot make use of the LINCS dataset as effectively as it could. On the other hand, deep learning based architecture can leverage the chemical structure information well and can surpass state of the art result, which uses chemical structure and gene expression features in combination with gene ontology [21].

We investigate whether the performance is better for test examples which are structurally similar to drugs in the training set. We find a test drug's most similar top-3 drugs in the train set, and calculate the average cosine similarity of this test example with these drugs. We investigate the relationship of the average cosine similarities versus log-loss values of the test samples.

Fig. 5 displays the cosine similarities of the test sample to the training examples and the log-loss computed on this

[1] http://toxnet.nlm.nih.gov/cgi-bin/sis/htmlgen?HSDB
[2] http://toxnet.nlm.nih.gov/cgi-bin/sis/search2/r?dbs+hsdb:@term+@rn+@rel+20830-81-3, accessed on Oct 30, 2019

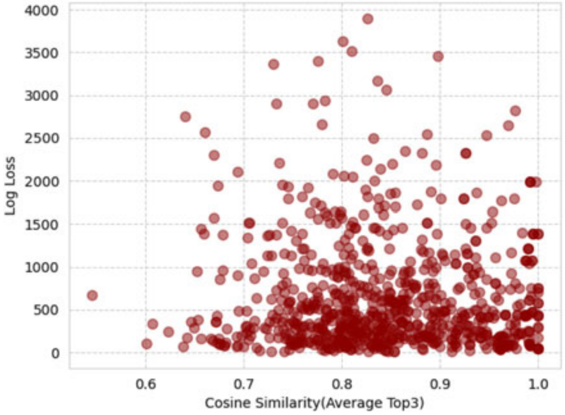


Fig. 5. Average of the cosine similarities of the test sample and its most similar top-3 drugs in the training set versus Log-Loss value of test sample.

test drug. The log-loss metric was chosen since it can be calculated independently of the threshold value selection. The figure shows all the test samples gathered from the 3-folds (only one outlier sample which has cosine similarity less than 0.5 was removed). Even if a test set drug does not have any similar drugs in the test set, that is its average cosine similarity is low, we can achieve a low log loss, indicating a good prediction. The contrary is also observed, where drugs that have high cosine similarities may produce a high log loss. As such, we conclude that the model does not memorize the examples in the training set.

5 Conclusion

The pharmaceutical drug development process is a long and demanding process. Unforeseen ADRs that arise at the drug development process can suspend or restart the whole development pipeline. Therefore, the prior prediction of the side effects of the drug at the design phase is critical.

In our DeepSide method, we use context-related (gene expression) features along with the chemical structure to predict ADRs to account for conditions such as dosing, time interval, and cell line. The proposed MMNN model uses GEX and CS as combined features and achieves better accuracy performance compared to the models that only use the chemical structure (CS) fingerprints. The reported accuracy is noteworthy considering that we do not have condition-independent class labels. The multi-modal architecture learns embeddings for each modality in a separate subnetwork and this combination of the embeddings lead to better prediction performance compared to using raw feature vectors. Finally, SMILESConv model outperforms all other approaches by applying convolution on SMILES representation of drug chemical structure. SMILESConv model learns the latent representations for drug structures directly from the drugs' SMILES strings with the supervision of the loss function. However, other models use a predefined feature set (MACCS) which are calculated only on the structures. The better performance of SMILESConv could be attributed to this supervision which helps the model to better capture side effect and drug specific latent representations. Besides, the ability of the convolutions to leverage sequential feature information using fewer number of parameters compared to the fully connected layers could be factoring in.

References

[1] J. Scheiber *et al.*, "Mapping adverse drug reactions in chemical space," *J. Med. Chem.*, vol. 52, no. 9, pp. 3103–3107, 2009.

[2] N. Atias and R. Sharan, "An algorithmic framework for predicting side effects of drugs," *J. Comput. Biol.*, vol. 18, no. 3, pp. 207–218, 2011.

[3] E. Pauwels, V. Stoven, and Y. Yamanishi, "Predicting drug side-effect profiles: A chemical fragment-based approach," *BMC Bioinf.*, vol. 12, no. 1, 2011, Art. no. 169.

[4] L.-C. Huang, X. Wu, and J. Y. Chen, "Predicting adverse side effects of drugs," *BMC Genomics*, vol. 12, no. 5, 2011, Art. no. S11.

[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] Y. Yamanishi, E. Pauwels, and M. Kotera, "Drug side-effect prediction based on the integration of chemical and biological spaces," *J. Chem. Inf. Model.*, vol. 52, no. 12, pp. 3284–3292, 2012.

[7] M. Liu *et al.*, "Large-scale prediction of adverse drug reactions using chemical, biological, and phenotypic properties of drugs," *J. Amer. Med. Inform. Assoc.*, vol. 19, no. e1, pp. e28–e35, 2012.

[8] L.-C. Huang, X. Wu, and J. Y. Chen, "Predicting adverse drug reaction profiles by integrating protein interaction networks with drug structures," *Proteomics*, vol. 13, no. 2, pp. 313–324, 2013.

[9] E. Bresso *et al.*, "Integrative relational machine-learning for understanding drug side-effect profiles," *BMC Bioinf.*, vol. 14, no. 1, 2013, Art. no. 207.

[10] G. M. Dimitri and P. Lió, "Drugclust: A machine learning approach for drugs side effects prediction," *Comput. Biol. Chem.*, vol. 68, pp. 204–210, 2017.

[11] H. Zhou *et al.*, "Medicascy: A machine learning approach for predicting small-molecule drug side effects, indications, efficacy, and modes of action," *Mol. Pharmaceutics*, vol. 17, no. 5, pp. 1558–1574, 2020.

[12] L. Yang, L. Xu, and L. He, "A citationrank algorithm inheriting google technology designed to highlight genes responsible for serious adverse drug reaction," *Bioinformatics*, vol. 25, no. 17, pp. 2244–2250, 2009.

[13] L. Xie, J. Li, L. Xie, and P. E. Bourne, "Drug discovery using chemical systems biology: Identification of the protein-ligand binding network to explain the side effects of CETP inhibitors," *PLoS Comput. Biol.*, vol. 5, no. 5, 2009, Art. no. e1000387.

[14] H. Zhou, M. Gao, and J. Skolnick, "Comprehensive prediction of drug-protein interactions and side effects for the human proteome," *Sci. Rep.*, vol. 5, 2015, Art. no. 11090.

[15] W.-P. Lee, J.-Y. Huang, H.-H. Chang, K.-T. Lee, and C.-T. Lai, "Predicting drug side effects using data analytics and the integration of multiple data sources," *IEEE Access*, vol. 5, pp. 20449–20462, 2017.

[16] Y. Zheng, H. Peng, S. Ghosh, C. Lan, and J. Li, "Inverse similarity and reliable negative samples for drug side-effect prediction," *BMC Bioinf.*, vol. 19, pp. 91–104, 2018.

[17] D. C. Zielinski *et al.*, "Pharmacogenomic and clinical data link non-pharmacokinetic metabolic dysregulation to drug side effect pathogenesis," *Nature Commun.*, vol. 6, 2015, Art. no. 7101.

[18] 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.

[19] C.-S. Wang, P.-J. Lin, C.-L. Cheng, S.-H. Tai, Y.-H. Kao Yang, and J.-H. Chiang, "Detecting potential adverse drug reactions using a deep neural network model," *JMIR Med. Inform.*, vol. 21, no. 2, 2018, Art. no. e11016.

[20] S. Dey, H. Luo, A. Fokoue, J. Hu, and P. Zhang, "Predicting adverse drug reactions through interpretable deep learning framework," *BMC Bioinf.*, vol. 19, 2018, Art. no. 476.

[21] Z. Wang, N. R. Clark, and A. Maayan, "Drug-induced adverse events prediction with the LINCS l1000 data," *Bioinformatics*, vol. 32, no. 15, pp. 2338–2345, 2016.

[22] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006.

[23] 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, 2015.

[24] A. Subramanian *et al.*, "A next generation connectivity map: L1000 platform and the first 1,000,000 profiles," *Cell*, vol. 171, no. 6, pp. 1437–1452, 2017.

[25] M.-C. Cai *et al.*, "ADReCS: An ontology database for aiding standardization and hierarchical classification of adverse drug reaction terms," *Nucleic Acids Res.*, vol. 43, no. D1, pp. D907–D913, 2014.

[26] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison, "Open babel: An open chemical toolbox," *J. Cheminformatics*, vol. 3, no. 1, 2011, Art. no. 33.

[27] S. Kim *et al.*, "PubChem substance and compound databases," *Nucleic Acids Res.*, vol. 44, no. D1, pp. D1202–D1213, 2015.

[28] G. Landrum *et al.*, "RDKit: Open-source cheminformatics," 2006. [Online]. Available: https://scholar.google.com/citations?view_op=view_citation&hl=en&user=xr9paY0AAAAJ&citation_for_view=xr9paY0AAAAJ:J_g5lzvAfSwC

[29] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 683–697, Sep. Sep. 1992.

[30] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[31] Y. Li and Y. Yuan, "Convergence analysis of two-layer neural networks with ReLU activation," in *Proc. 31st Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, pp. 597–607.

[32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[35] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Processi. Syst.*, 2017, pp. 5998–6008.

[36] G. B. Goh, N. O. Hodas, C. Siegel, and A. Vishnu, "SMILES2Vec: An interpretable general-purpose deep neural network for predicting chemical properties," 2017, *arXiv:1712.02034*.

[37] M. Hirohara, Y. Saito, Y. Koda, K. Sato, and Y. Sakakibara, "Convolutional neural network based on smiles representation of compounds for detecting chemical motif," *BMC Bioinf.*, vol. 19, no. 19, 2018, Art. no. 526.

[38] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst. 30*, 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[39] L. S. Shapley, *17. A value for n-person games*. Princeton, NJ, USA: Princeton Univ. Press, 2016.

[40] J. E. Groopman and L. M. Itri, "Chemotherapy-induced anemia in adults: incidence and treatment," *J. Nat. Cancer Inst.*, vol. 91, no. 19, pp. 1616–1634, 1999.

[41] S. Okubo, K. Nakatani, and K. Nishiya, "Gastrointestinal symptoms associated with enteric-coated sulfasalazine (azulfidine en tablets)," *Modern Rheumatol.*, vol. 12, no. 3, pp. 0226–0229, 2002.

[42] C. E. Lopes, G. Langoski, T. Klein, P. C. Ferrari, and P. V. Farago, "A simple HPLC method for the determination of halcinonide in lipid nanoparticles: development, validation, encapsulation efficiency, and in vitro drug permeation," *Brazilian J. Pharm. Sci.*, vol. 53, no. 2, 2017.

[43] A. Toth, M. Lesser, G. Naus, C. Brooks, and D. Adams, "Effect of doxycycline on pre-menstrual syndrome: A double-blind randomized clinical trial," *J. Int. Med. Res.*, vol. 16, no. 4, pp. 270–279, 1988.

[44] F. Polatti, F. Viazzo, R. Colleoni, and R. E. Nappi, "Uterine myoma in postmenopause: A comparison between two therapeutic schedules of HRT," *Maturitas*, vol. 37, no. 1, pp. 27–32, 2000.

Onur Can Uner received the BS degree in computer engineering from Middle East Technical University, Ankara, Turkey, in 2016, and the MS degree in computer engineering from Bilkent University, Ankara, Turkey, in 2019.

Halil Ibrahim Kuru received the BS and MS degrees in computer engineering from Bilkent University, Ankara, Turkey, in 2016 and 2019, respectively. He is currently working toward the PhD degree in the Computer Engineering Department, Bilkent University, Ankara, Turkey. He is also a software engineer with ASELSAN.

R. Gokberk Cinbis received the BS degree in computer engineering from Bilkent University, Ankara, Turkey, in 2008, and the MS degree from Boston University, Boston, Massachusetts, in 2010. He was a doctoral researcher with LEAR (now THOTH) Team, INRIA Grenoble, France, from 2010 until 2014. He was an assistant professor with Bilkent Univeristy from 2016 to 2017. Since then, he is an assistant professor with Computer Engineering Department, Middle East Technical University. His research interests include machine learning and computer vision, with special interest in data-efficient deep learning via minimal supervision (zero-shot, few-shot, weakly-supervised, self-supervised learning), learning to learn (meta learning), vision and language integration, and large-scale image/video understanding.

Oznur Tastan received the BS degree in biological sciences and bioengineering from Sabanci University, Istanbul, Turkey, in 2004, and the MS and PhD degrees from Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, in 2007 and 2011, respectively. She worked as a postdoctoral researcher with Microsoft Research New England. She was an assistant professor with Computer Engineering Department, Bilkent University. Currrently, she is an assistant professor with the Faculty of Natural Sciences of Sabanci University, affiliated with the Computer Science and Engineering and Molecular Biology Genetics and Bioengineering Programs.

A. Ercument Cicek received the BS and MS degrees in computer science and engineering from Sabanci University, Istanbul, Turkey, in 2007 and 2009, respectively, and the PhD degree in computer science from Case Western Reserve University, Cleveland, Ohio, in 2013. Then, he worked as a lane fellow in computational biology with Carnegie Mellon University till 2015. Since then, he is an assistant professor with the Computer Engineering Department, Bilkent University and is an adjunct faculty member with Computational Biology Department, Carnegie Mellon University.