DEEPSIDE: PREDICTING DRUG SIDE EFFECTS WITH DEEP LEARNING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

By
Onur Can ÜNER
September 2019

DEEPSIDE: PREDICTING DRUG SIDE EFFECTS WITH DEEP LEARNING
By Onur Can ÜNER
September 2019

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

A. Ercüment Çicek(Advisor)

Öznur Taştan Okan(Co-Advisor)

Özlen Konu

R. Gökberk Cinbiş

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

DEEPSIDE: PREDICTING DRUG SIDE EFFECTS WITH DEEP LEARNING

Onur Can ÜNER
M.S. in Computer Engineering
Advisor: A. Ercüment Çicek
Co-Advisor: Öznur Taştan Okan
September 2019

Drug failures due to unforeseen adverse effects at clinical trials pose health risks for the participants and cause substantial financial losses. Side effect prediction algorithms, on the other hand, have the potential to guide the drug design process. LINCS L1000 dataset provides a vast resource of gene expression profiles across different cell lines that are induced with different dosages taken at different time points. The state-of-the-art approach in the literature relies on high-quality experiments in LINCS L1000 and discard a large portion of the recorded experiments. In this study, we investigate whether more information can be extracted from this remaining set of experiments with a deep learning-based approach. We experiment with 6 different deep learning architectures that use (i) gene expression data from the LINCS L1000 project, (ii) chemical structure fingerprints of drugs, (iii) SMILES string representation of drug structure, and (iv) the atomic structure of the drug molecules. The multilayer perceptron (MLP) based model which uses chemical structures and gene expression features achieve 88% micro-AUC and 79% macro-AUC, thus offering better performance in comparison to the state-of-the-art studies on side effect prediction. We observe that the chemical structure is more predictive than the gene expression profiles despite the fact that the features are extracted with different deep learning models. Finally, the convolutional neural network-based model that uses only SMILES strings of the drugs provides 82% macro-AUC, and 88%micro-AUC improvements, better performing than the models that use gene expression and chemical structure features simultaneously.

Keywords: Machine learning, deep learning, molecular biology, ADR, side effects, drugs, CNN, MLP .

ÖZET

DEEPSIDE: ILAÇLARIN TERS VE YAN ETKILERINI TAHMIN ETMEK IÇIN DERIN ÖĞRENME

Onur Can ÜNER
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: A. Ercüment Çicek
İkinci Tez Danışmanı: Öznur Taştan Okan
Eylül 2019

Klinik denemelerde ortaya çıkan ve öngörülemeyen yan etkilerden kaynaklanan ilaç başarısızlıkları, katılımcılar için ciddi sağlık riskleri oluşturur ve önemli finansal kayıplara neden olmaktadır. Öte yandan yan etki tahmin algoritmaların ilaç tasarım sürecini yönlendirme potensiyeli vardır. LINCS L1000 veri kümesi, farklı hücre hatlarında farkı zamana noklarında bir çok ilacın farklı dozda uygulandığı, geniş bir gen ifade veri seti içermektedir. Literatürdeki en güncel yaklaşım, sadece LINCS L1000'deki yüksek kaliteli deneylere dayanmakta ve farklı deneylerin büyük bir kısmını model dışında bırakmaktadır. Bu çalışmada, derin öğrenme temelli bir yaklaşımla geriye kalan kayıtlı deneylere ait verilerden daha fazla bilginin elde edilip edilemeyeceğini araştırmaktayız. (i) LINCS L1000 projesinden gen ifade verilerini, (ii) ilaçların kimyasal yapı parmak izlerini, (iii) SMILES ilaç yapısının dizi gösterimini ve (iv) ilaçların atom çözünürlüğündeki moleküler yapılarını kullanan 6 farklı derin öğrenme mimarisi ile deneyler yapmaktayız. İlaç moleküler yapısı, kimyasal yapıları ve gen ifade özelliklerini kullanan çok katmanlı algılayıcı (MLP) tabanlı model, %88'lik mikro-AUC ve %79'luk makro-AUC'ye ulaşabilmekte ve bu sayede daha yüksek yan etki tahmini imkânı sunmaktadır. Her ne kadar farklı derin öğrenme modelleri ile güçlü gösterimler çıkarılmış olsa da, kimyasal yapının gen ifade profillerinden daha iyi bir tahmin gücü olduğunu da gözlemlemekteyiz. Son olarak, sadece ilaçların SMILES dizilerini kullanan evrişimli sinir ağı tabanlı model, hem gen ifadesi hem de kimyasal yapı özelliklerini aynı anda kullanan modellerden daha iyi bir performans sunmaktadır.

Anahtar sözcükler: Makine öğrenmesi, derin öğrenme, moleküler biyoloji, ADR, ilaç, yan etki, CNN, MLP .

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my thesis advisors Asst. Prof. Öznur Taştan and Asst. Prof. A. Ercüment Çicek for their guidance, support, and patience throughout my master's studies. Especially I would like to thank Öznur Taştan for accepting me as her master student and giving me the opportunity to work with her. I will never forget the support she gave me during the most difficult times in my master's studies. Asst. Prof. A. Ercüment Çicek and Asst. Prof. R. Gökberk Cinbiş gave always valuable feedback and guidance that helped to complete this thesis. It was a great honor and a chance for me to work with Öznur Taştan, A. Ercüment Çicek, and R. Gökberk Cinbiş.

I would also like to thank my jury members Assoc. Prof. Özlen Konu and Asst. Prof. R. Gökberk Cinbiş for reading my thesis and kindly accepting to be in my thesis jurry.

Finally, I would like to express my gratitude to my parents and my brother for their continuous support. They were always there for me, whenever I need motivation and support. I would also like to thank Gizem Aluç, who is someone very special for me, for her immense support and understanding during my thesis studies. I would like to thank Tayfun Ateş for reading my thesis and giving constructive comments.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Today, the estimated R&D cost for a new drug is ∼\$2.6 billion. Pharmaceutical research is a high-risk domain as the probability of a drug passing clinical trials, and entering into the market is only 11.8% [2]. The high failure rate (70%) in early trial phases is perhaps justifiable. However, even the failure rate of Phase III trials is typically 50% [3, 4, 5]. Failed drugs result in a substantial financial loss for the companies. One example is Vintafolide, a platinum-resistant ovarian cancer drug, whose failure due to the recommendation by an independent safety board at Phase III, led to a 62% slump in the Endocyte's share price [5].

The primary drivers of drug failures are safety and unforeseen adverse drug reactions (ADR), which constitute 17%-30% of all failures at Phase III [6, 7]. This is a major bottleneck in the drug discovery pipeline. More importantly, failures due to ADRs jeopardizes the well being of the participants. For instance, in the Phase III trials of Pembrolizumab, a trial drug targeting multiple myeloma, the fatality rate was higher for the treated group of patients in comparison to the control group [8]. This is not an isolated case, either. The number of patients who participated in the clinical trials that failed at Phase III during the 2012 - 2015 period was 150k [5]. Thus, the risk is substantial.

Computational methods, on the other hand, can provide valuable guidance *in silico* for predicting possible ADRs before going into the clinical trials. This would spare the health risks and can warn researchers and companies before focusing on a specific candidate. Several learning based methods have been proposed for predicting the side effects of drugs based on various features such as: chemical structures of drugs [9, 10, 11, 12, 13, 14, 15, 16, 17], drug-protein interactions [18, 19, 12, 13, 14, 15, 20, 17], protein-protein interactions (PPI) [12, 16], activity in metabolic networks [21, 22], pathways, phenotype information and gene annotations [12].

While these methods have proven useful for predicting ADRs, the features they use are solely based on external knowledge about the drugs (i.e., drug-protein interactions, etc.). Thus, their predictions are not cell or condition (i.e., dosage) specific. To address this issue, [23] utilized the data from LINCS L1000 project, which profiled 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, [23] provided the first comprehensive, unbiased, and cost-effective prediction of ADRs for compounds whose gene expression profiles can be measured. The authors [23] formulated the problem as a multi-label classification task. They [23] report that the *extra trees*-based method performs the best among various classifiers they tried. Even though the LINCS dataset contains a total of 473,647 experiments for 20,338 compounds, the method only utilized the highest quality experiment for each drug to minimize noise. However, this also means that most of the expression data were not used, suggesting a room for improvement in the prediction performance.

In this thesis, we propose a deep learning framework which is called DeepSide for ADR prediction which uses *in vitro* gene expression profiling experiments along with the chemical structures of the drug compounds. Our models operate on the full LINCS L1000 dataset and uses the SIDER dataset as the ground truth for drug ADR pairs' label [24]. We propose four different multi-layer perceptron based neural network architectures: (i) a multi-layer perceptron (MLP), (ii) MLP with residual connections (ResMLP), (iii) multi-modal neural network (MMNN), and finally, (iv) multi-task neural network (MNN). In addition to these, we propose two convolutional neural network (CNN)-based architectures: (i) SMILES convolutional neural network (SMILESConv) and (ii) Edge Convolutional Neural Network (EdgeConv).

GEX dataset contains drug experiments based on conditions and these conditions take the main role to define the drug's side effects. Drug experiments that are performed under different conditions, even for the same drug, may cause different side effects. This fact summarizes the difficulty of using the GEX dataset for the side effect prediction. Extensive experiments with various parameters of these architectures were performed to compare the impact of the CS, GEX (including META data for some experiments) datasets to side effect predictions. Our experiments show that the MLP model using chemical structure(CS) is a robust predictor of side effects. MLP model, which uses CS features as input, produces 10.5% macro-AUC and 12% micro-AUC improvement compared to the state-of-the-art result. Using GEX features along with CS fingerprints provides performance improvement by just using MMNN architecture which is the best model among MLP-based architectures. MMNN model, which uses CS, GEX and META features, achieves 0.79 macro-AUC and 0.877 micro-AUC while the MLP model with CS features achieves 0.784 macro-AUC and 0.866 micro-AUC. These results between MLP models show that using pertinent chemical structure features (CS) along with the context-related features (gene expression) can increase the performance of the classifier just by using the MMNN architecture.

The other way to represent the chemical structure of drugs is to use SMILES strings format. We apply a character-level convolutional neural network to SMILES string for the side effect prediction. This model is called SMILESConv and achieves better performances in all metrics compared to all other architectures including MMNN.

The other proposed model is EdgeConv. EdgeConv is a convolutional neural network that uses the atomic relation dataset which is created by us. The atomic relation dataset contains properties of atoms and bonds in the drugs. EdgeConv is a novel method that is open to improvement, although performance results of this method are behind the other architectures.

Chapter 2

Background Information

In this section, we provide a detailed account of the data collection processes and deep neural network (DNN) approaches that are used in this study.

We start by noting that despite the subtle distinction between the terms side effects and adverse drug reactions these phrases are generally used interchangeably, and we adopt this view as well. Side effects are generally defined as undesired yet previously noted and mostly foreseen consequences of taking a prescribed medication. Adverse drug reactions (ADRs), on the other hand, include all the foreseen and unforeseen side effects and just like the side effects they can be dependent on dose and patient drug intolerance.

To compare our ADR prediction performance, we follow [23] and we adopt their data collection and process pipelines.

2.1 Data Collection

We use multiple datasets on drugs. The LINCS L1000 dataset contains gene expression (GE) profiles of different cell lines when treated with different small-molecule compounds. It covers 20,413 small-molecule compounds. There are different experiments, wherein each of these small-molecule compounds are administered with a different dose on different cell lines and gene expression levels of 978 landmark genes are recorded.

The LINCS L1000 dataset incorporates two development phases: Phase 1 and Phase two. Phase-1 refers to the drugs whose studies have been fully completed whereas Phase-2 incorporates drugs which are at an experimental stage in development. [23] uses the Phase1 of the LINCS L1000 set. We also focus on Phase 1 compounds. The authors of [23] 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 dataset that with this feature set (GO+CS) contains 791 drugs and 1052 side effects. To be able to compare our results with the state-of-the-art, we use these 791 drugs to build our models. In total, there are 18,832 experiments for these 791 drugs in LINCS 1000 dataset. We use 3-fold cross-validation to evaluate our models as also done in [23].

Side effect information is obtained from the SIDER Database [24]. SIDER contains data for 1430 drugs (whether they cause 5868 side effects). We use 1052 side effects that are also used in [22] for a fair comparison, as explained above. Note that [23] ends up with 1052 side effects by removing the side effects which have less than 10 positive samples - side effects that occur for less than 10 drugs.

Chemical structure features (CS) are computed with OpenBabel Chemistry Toolbox [25] to create a 166-bit MACCS chemical fingerprint matrix for each drug (a binary vector of length 166).

A SMILES string is an efficient data format which represents the 2D molecular graph of a drug/small molecule as a 1D string. In this simplified representation, the lack of some chemical descriptors such as chemical bond angles means that different drugs/small molecules can be represented with the same string. The SMILES strings for used drugs are downloaded from PubChem. The SMILES strings are used for two purposes in this study: (i) to create atomic relation tensor to used as input for the EdgeConv method, and (ii) to create chemical fingerprints of the drugs for 1D convolution used in SMILESConv method.

2.2 Side Effect Prediction in the Literature

The emergence of the unintended side effects of a new drug is a crucial factor that affects its discovery and development processes. Side effects of a new drug may occur at the end of these long processes and may still exist some hidden side effects. These facts make drug side effect prediction with computational methods important. There exist many computational approaches to predict side effects in the literature. In this section, side effect prediction approaches that use machine learning are reviewed.

DrugClust [26] is the one of the side effect prediction approaches. This method is an R package. DrugClust has different benchmark datasets including chemical and biological features. This method clusters drugs according to the similarity of their features. Probability scores of each side effects are calculated by using the Bayesian approach for each cluster. After drug-side effect probability scores obtained, the enrichment analysis is performed to validate clusters and investigate interactions between similar side effects and similar biological pathways.

The other side effect prediction approach that uses several machine learning methods is [27]. This approach gathers the chemical and biological properties of drugs to create drug representation features as an input of the classifiers. Chemical substructures, target proteins and therapeutic indications of the drugs are used to form drug representation features. In this approach, statistical-based, distance-based and ensemble learning methods are employed. Side effect labels are obtained from SIDER [24]. Since side effect labels are an imbalanced dataset, they split side effects into 3 splits according to the number of associated drugs and the different prediction method is used for each different split.

Drug similarity can give an intuition to predict side effects since similar drugs tend to have the same side effects. Most of the existing side effect prediction approaches including [26, 27] are based on this assumption. The inverse of that assumption can be also meaningful for side effect prediction since dissimilar drugs probably have different side effects. One of the other side effect prediction approaches [28] is built based on that perspective. They use four different drug properties that are chemical structures, drug substituents, target proteins, and drug therapeutic information. They use machine learning methods to classify drugs. Each side effect has its own binary classifier. There are different similarity measurements for each type of feature. According to the obtained similarity score, they extract highly-reliable negative samples for each side effect. Thanks to that way they avoid class imbalance issues since they have highly-reliable negative samples that have an equal number of the positive samples for each side effect classifier.

Deep learning models are also employed to predict side effects in recent studies [29, 30]. [30] uses chemical structure as drug feature. ECFP [31] algorithm is used to extract chemical fingerprints from the chemical structure. Their deep network architecture contains 1D-convolutional layers and attention mechanism. They [30] compare their neural fingerprints that are extracted from the deep network with the chemical fingerprints that are calculated from different algorithms to evaluate side effect prediction performance. They [30] also interpret the side effect predictions of the network to map chemical substructures with the groups of related ADRs.

The other deep learning approach [29] uses multilayer perceptron architecture to predict side effects by using biological, chemical and semantic information as combined features. This study enriches drug features by using clinical notes and case reports of the drugs. Word2Vec [32] embedding method is employed to extract semantic features from the biomedical literature text dataset.

2.3 Deep Neural Networks

2.3.1 Multilayer Perceptron

Multilayer perceptrons are feedforward networks that consist of several linear layers of three categories: the input layer, the hidden layers, and the output layers [33]. To extract non-linear features, activation functions are applied between linear layers. The output of each layer is used as input for the next layer. Feedforward operation of the single layer with matrix notation is represented in Equation 2.1. W, X and, b notations represent weight matrix, input matrix, and bias term vector, respectively. σ is an activation function for the Equation 2.1. An example of a multilayer perceptron network is shown in Figure 2.1.

Multilayer perceptron (MLP) is the stack of several succisive linear layers [33]. By increasing the number of linear layers, the network learns more complex features. MLP is also a feedforward network. In the forward propagation phase of the training, the input features of the neural network pass from the input layer through the hidden layers to the output layer. The output layer uses the intermediate features obtained from the hidden layers to make a prediction. The predictions of the network are compared against the ground truth values in calculating the loss value in accordance with a cost function. In the backward propagation phase of the training, the weights of the neural network are updated according to the previously calculated loss value. The network updates its weights with a suitable gradient-based optimization algorithm such as the stochastic gradient descent algorithm.

$$Z = \sigma\left(f\left(W^{T}X\right) + b\right) \tag{2.1}$$

2.3.2 Convolutional Neural Network

CNN is the most common method to analyze image-based contents. It consists of convolutional layers which learn local features from the input by applying several

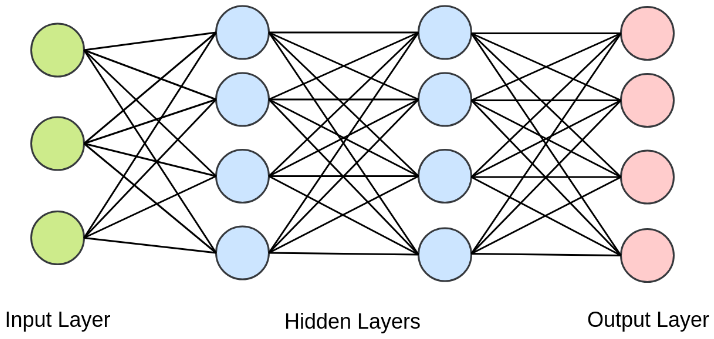


Figure 2.1: Multilayer perceptron with two hidden layers.

filters [34]. In recent works, CNNs are also applied to text-based inputs to extract sequential information by using 1-dimensional convolutional operations [35].

For an image processing task, convolutional layer processes images through two dimensions, which are height and width. Convolutional layers are responsible for extracting pixel relations in a local spatial environment. Filters slide over images to learn spatial information from relations between the previously filtered patches. The receptive field of each neuron at the output layer of the convolutional layer indicates a local patch of the previous layer which equals to the kernel size of the convolutional operation. Increasing convolutional layers also increases the complexity of the learned features since the receptive field expands with the number of convolution operations. An illustrative example of the CNN architecture to classify digit images is shown in Figure 2.2.

The convolutional layer has a kernel, stride, and padding parameters. The kernel consists of weights to be learned. The kernel weights are learned by a sliding filter over the input image. The sliding pixel amount of the kernel is a stride parameter of the convolutional layer. Applying the convolution operation, which has larger than one kernel size or 1 stride without padding reduces the spatial dimensions of the image. In some cases, we want to keep the spatial dimension of the image to avoid the information loss from the border pixels. Because of this reason, a suitable padding parameter is used to add pads to the border of the input.

1D and 2D convolutional layers have the same characteristics but only differs

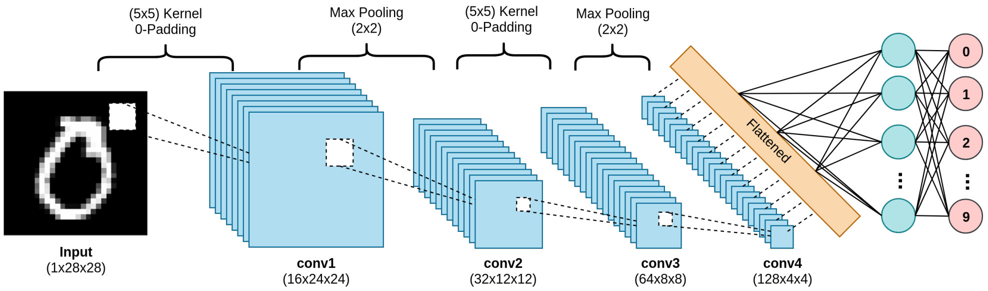


Figure 2.2: Simple CNN architecture to classify handwritten digits.

by the number of dimensions of their inputs. Just like as the 2D convolution layer, a 1D convolution layer is applied to the input by sliding 1D kernel over the input. How 1D convolution works is shown in the Figure 2.3.

Pooling layer is a common method for reducing the spatial dimension of input. As the number of spatial dimensions gets smaller, the parameters of the network also become fewer, and this helps to attain improvements in computational performance. Additionally, the pooling layer preserves the translation invariance features, and it helps in avoiding overfitting. The most commonly used pooling layers are max pooling and average pooling layers. For instance; to down sample the input of the pooling layer by 2 times, a 2x2 kernel is applied to the input. Max pooling layer gathers the maximum values of the kernel applied patches from the input while average pooling gathers the average values for each patch. A general description of the pooling layer, and max and average pooling layers are shown in Figure 2.4. To show the difference between max pooling and average pooling, we provide an example in Figure 2.4. The output features are different, although both pooling methods use the same input. When we look at the average pooling example, the average pooling layer returns 4 from each patch since the average values of each patch are equal to 4. However, the max-pooling layer returns different values which are obtained by taking a max operation for each patch.

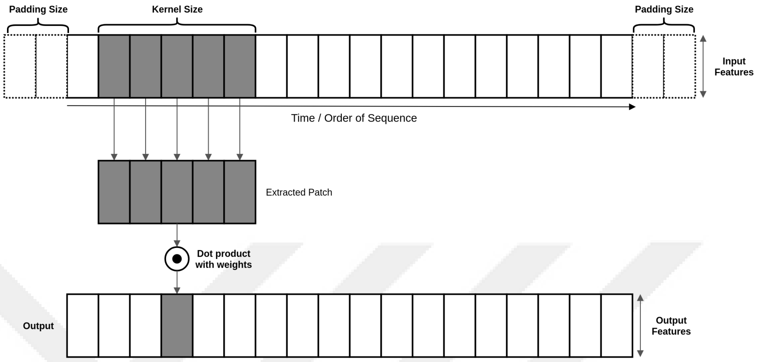
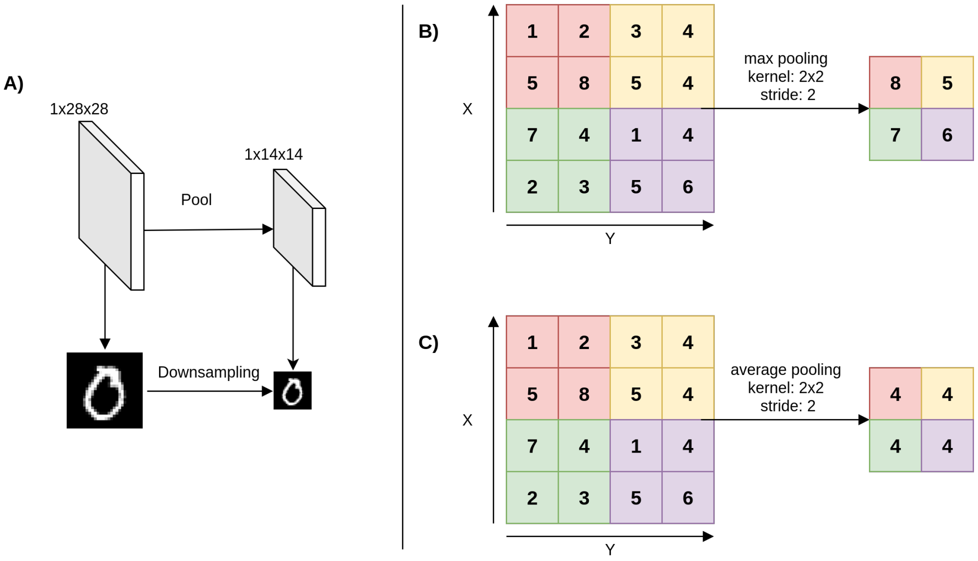


Figure 2.3: A simple example demonstrating how 1D convolution works.



B)

1	2	3	4
5	8	5	4
7	4	1	4
2	3	5	6

8	5
7	6

C)

1	2	3	4
5	8	5	4
7	4	1	4
2	3	5	6

4	4
4	4

Figure 2.4: A) Shows the downsampling of the input image after applying the pooling layer. B) Shows that how works the max-pooling layer. C) Shows that how works the average-pooling layer. This figure is adopted from [1].

2.3.3 Loss Functions

Since the side effect prediction problem is a multilabel classification task, we use a binary cross-entropy loss function. In this section, the concept of binary cross-entropy (BCE) and class weighted version of BCE are explained.

2.3.3.1 Cross Entropy Loss

Multi-label classification means that one sample can belong to many classes. This should not be confused with multi-class classification, where one sample can only belong to one class of the many classes.

The softmax activation function is used to get the posterior probabilities of the classes. The sum of the probabilities of each class equals to 1. The mathematical formulation of the softmax activation function is represented in the Equation 2.2. s_i is the i_{th} class logit value which is obtained from the output layer before the softmax function. C represents the set of output classes.

The network which carries out the multi-label classification uses a sigmoid activation function which is applied to the output layer of the network. The sigmoid activation function returns values bounded between 0 and 1 which represent the probabilities for each. Sigmoid function formula is given in the Equation 2.3.

$$f\left(s_i\right) = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \tag{2.2}$$

$$f\left(s_i\right) = \frac{1}{1 + e^{-s_i}} \tag{2.3}$$

Multi-class classification task can be resolved by using a categorical cross-entropy loss function. Categorical cross-entropy or log loss function is shown in the Equation 2.4. t_i is the ground truth value of the i_{th} class and y_i is the output of the softmax function for that class.

$$CE = -\sum_{i}^{C} t_i \log\left(y_i\right) \tag{2.4}$$

For the binary class prediction case, cross-entropy function transforms to Equation 2.5. This loss function is defined as binary cross-entropy loss function. In our case, there are many side effects all of which have binary labels. The summation of each loss value of the side effects produces the total loss value for the network prediction task.

$$CE = -\sum_{i=1}^{C=2} t_i \log\left(s_i\right) = -t_1 \log\left(s_1\right) - \left(1 - t_1\right) \log\left(1 - s_1\right) \tag{2.5}$$

The Equation 2.6 shows binary cross-entropy loss function, which is used in this study as the loss function. N represents the number of side effects, t_k represents the label value of the class k and y_k shows the prediction probability of the kth class. Since our side effect dataset is imbalanced, we also apply class weights to the binary cross-entropy loss function. w_k represents class weights. Class weights are adjusted as inversely proportion of the class sample count.

$$WeightedBCE = -\frac{1}{k}\sum_{k=1}^{N} w_k \left(t_k \log\left(y_k\right) + \left(1 - t_k\right) \log\left(1 - y_k\right)\right) \tag{2.6}$$

Chapter 3

Methods

3.1 Datasets

The LINCS L1000 project consists of gene expression (GEX) of 978 landmark genes in different cell lines when perturbed by 20,413 small molecule compounds under different conditions such as time and dosages. The LINCS project provides two phases within the L1000 dataset. We confine our analyses to Phase I of the L1000 dataset in order to compare our results with the [23]. Additionally, there are 5 different data process levels in the L1000 dataset. We use Level 5, which contains signature experiments with the preprocessed data amongst its replicates. The metadata is also provided for all experiments in the L1000 project. This metadata comprises the measurement time, cell line, and dosage value information for the gene expressions.

The chemical structures of the drugs/small molecule compounds(CS) are obtained from [23], which converts SMILES represented chemical structures into the binary 166-bit Molecular ACCess System (MACCS) chemical fingerprint format using the Open Babel cheminformatic toolbox [25].

ADR labels for the drugs/small molecule compounds are downloaded from the

SIDER [24]. The dataset contains 5868 side effects, 1430 drugs, and 139,756 drug-ADR relations. To make a meaningful comparison between our results and those in [23], we use the same data content pre-processed in [23], which is a version of SIDER labels.

ADRs are grouped into related groups by using ADR ontology database (ADReCS) [36]. ADReCS is a comprehensive ADR ontology database that contains 6,847 ADR terms in a four-level hierarchical tree.

SMILES is a data format which converts the chemical structure of molecules into strings of ASCII characters [37]. Since the chemical structure encodes valuable characteristic properties of the molecules, it is plausible to use SMILES Strings when investigating the side effects of the drugs. SMILES Strings of the drugs annotated by SIDER are downloaded from PubChem [38]. RDKit Cheminformatics toolbox is used to extract extended SMILES Strings of the drugs [39]. The extended SMILES Strings contain all the primary chemical bonds as well as the hydrogen bonding information explicitly. Adding these data as extra characters into SMILES Strings gives longer text representations for the drugs. Zero-padding is applied to bring all the sequences to the longest sequence size found among the SMILES Strings. In this dataset, the alphabet contains 33 unique characters, including the end of sequence character. We filter out SMILES Strings that have less than 100 characters and more than 400 characters. After this filtering 615 out of 791 are kept. For these drugs, side effects (target classes) are also filtered due to the limitation of positive sample count. As a result, 651 drugs and 1042 side effects are included in SMILES Strings dataset. These side effects form a subset of the 1052 side effects culled from the CS, GE datasets. In this study, character vocabulary of the SMILES strings consists of the following characters: ”_, #, ,, +, -, /, 1, 2, 3, 4, 5, 6, 7, 8, =, C, F, H, I, N, O, P, S, [, \,], c, Cl, Br, n, o, s”. Characters that occur less than 2 among all SMILES strings are removed from the character vocabulary and ”_” character is used instead of them.

SMILES Strings are also used to create the atomic bond dataset. Thanks to SMILES Strings, we can extract existing atom and bonding types between each atom. Similar bond properties for the drugs are used in the [40] study as an attention weights of their GCN [41] model. They [40] applied atomic bond features to a different architecture than our proposed architecture. However, they [40] have inspired us to select valuable properties of the atomic bonds. Edge convolutional neural network is built so as to leverage the atomic bond dataset. In this dataset, nodes correspond to atoms in the drug and the edges denote the bonds between the pairs of atoms. The SMILES Strings of 615 drugs contain 12 unique atom types. Accordingly, we have 12 different nodes in all drug graphs. Edges between these atoms are represented as 60-dimensional feature vector. *[begin_atom_properties]* and *[end_atom_properties]* have 26 binary values. *[bond_properties]* has 8 binary values. Concatenation of these three vectors (26+26+8) forms the feature vector (60-dimensional) for given atom pair. If there is a bond between two atoms, then 60-dimensional vector is filled with *[begin_atom_properties][end_atom_properties][bond_properties]* values. If there is no bond between two atoms, the atomic-bond (60D) features are populated with 0 values.

Atomic Properties

The atomic properties vector has a dimension of 26.

Atom number: 12-dimensional one-hot vector to describe the atomic number.

Neighbour Count: 6-dimensional one-hot vector to describe the number of neighbours of an atom (excluding Hydrogen).

Hydrogen Count: 5-dimensional one-hot vector to describe the total number of Hs (explicit and implicit) on the atom. (there exist 5 Hydrogens for each atom in the SMILES String dataset maximum.)

Atomic Charge: Atomic charge as integer value.

Is Atom in Ring: Boolean value to describe whether a target atom is a ring member.

Is Atom Aromatic: Boolean value to describe whether a target atom belongs to an aromatic group.

Bond Properties

The bond properties vector has a dimension of 8.

Bond Type: 4-dimensional one-hot vector to describe the bond type. (Single,

Double, Triple, Aromatic)
Is Bond Aromatic: Boolean value to describe whether a bond belongs to an aromatic group.
Hydrogen Count: Boolean value to describe whether a bond is considered to be conjugated.
Is Atom in Ring: Boolean value to describe whether a bond is a ring member.

3.2 Architectures

3.2.1 Multi-Layer Perceptron - MLP

Multilayer perceptron (MLP) [33] is our baseline architecture. In this architecture, we define several decision blocks which contain linear neural network layers, batch normalization [42], and ReLU activation functions [43], in this order. The number of decision blocks and the number of neurons that exist in the linear layers depend on the features used. Different architectural configurations are created for different feature types. MLP model returns the probability for each side effect by using features that are extracted for each drug. Hence, MLP contains neurons as much as the number of side effects at the output layer. The sigmoid activation function is applied to the output layer of the network. An example of the basic MLP architecture, which uses concatenated CS and GEX features is represented in Figure 3.1.

3.2.2 Residual Multi-Layer Perceptron - ResMLP

Residual multilayer perceptron (ResMLP) possesses essentially the same architecture with the MLP. The only difference is adding skip connections between the hidden layers of the MLP. Although increasing the number of layers enables the network to extract more complex features, it may also cause vanishing gradients.

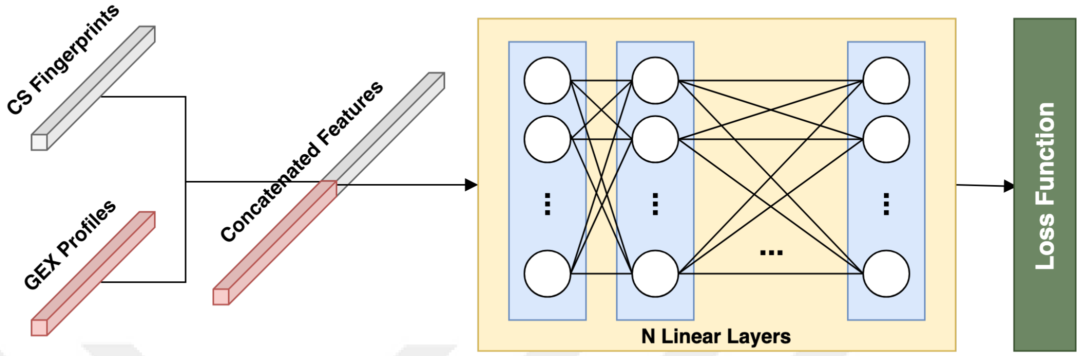


Figure 3.1: Multi-Layer Perceptron architecture developed for the concatenated features of GEX and CS.

Therefore, skip connections are introduced between the hidden layers to overcome this problem [44].

3.2.3 Multi-Modal Neural Network - MMNN

This approach defines two distinct MLP branches to feed two different data types. In our experiments, we use one of them for gene expression dataset (GE - LINCS L1000 Level 5) and the other one for the chemical structures dataset (CS). These two separate networks are responsible for extracting features using their data type. After the features extraction phase, feature fusion and classification phases are followed in the pipeline. We use two different feature fusion methods: feature concatenation and feature summation. Let $B \times N$ and $B \times M$ are two matrices with the same row and different column size. Feature concatenation method merges two $B \times N$ and $B \times M$ features into a $B \times (N + M)$ dimensional feature. Feature summation method gets two features which have identical dimensions $(B \times N)$ and makes elementwise summation over these features to obtain a fused $B \times N$ feature. At the top of the network, there are classification layers which use the fused feature to make side-effect prediction. Classification part of the network is designed as a MLP. The architecture of the multi-modal neural network with two branches is shown in Figure 3.2.

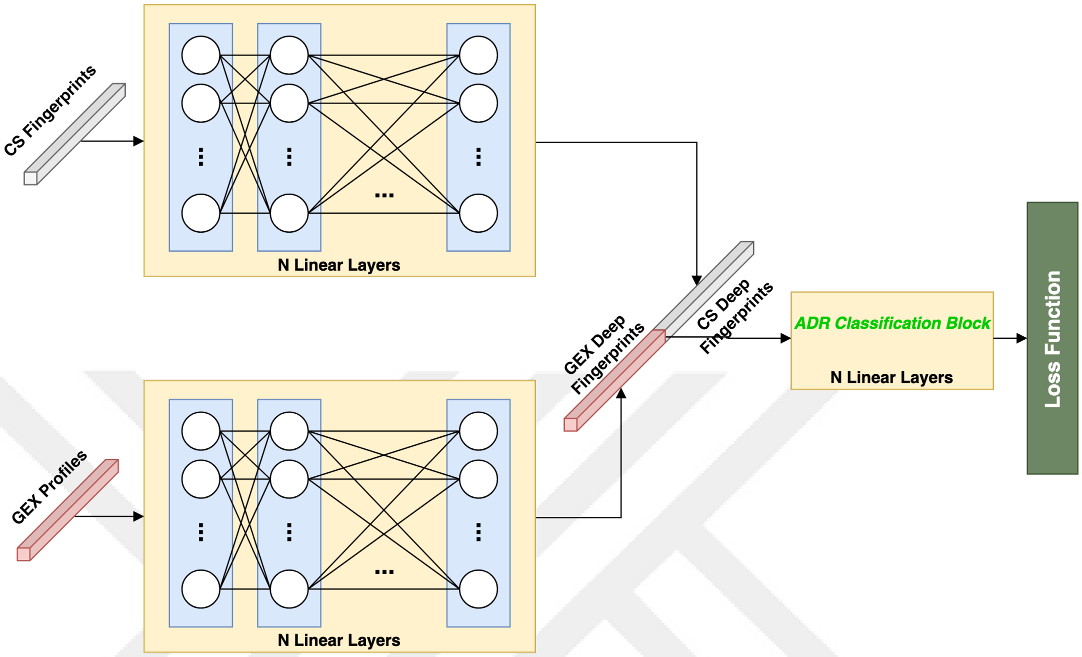


Figure 3.2: Multi-Model Neural Network architecture has two branches to feed network separately with GEX and CS features.

3.2.4 Multitask Neural Network - MNN

Drug side effects can be represented in a hierarchical structure. Our multitask learning (MTL) approach splits drug side effects into the related groups by using drug side effects ontology tree, which is obtained from ADReCS database [45]. This approach contains base and task-specific networks. Base and task-specific network architectures are similar to the MLP architecture. The base network is responsible for extracting features by using a combined set of GEX and CS features. Hidden layers of the base network are shared between task-specific networks. Since this approach extracts common features for all task-specific networks, overfitting risk is less pronounced [46]. Architecture details are shown in Figure 3.3.

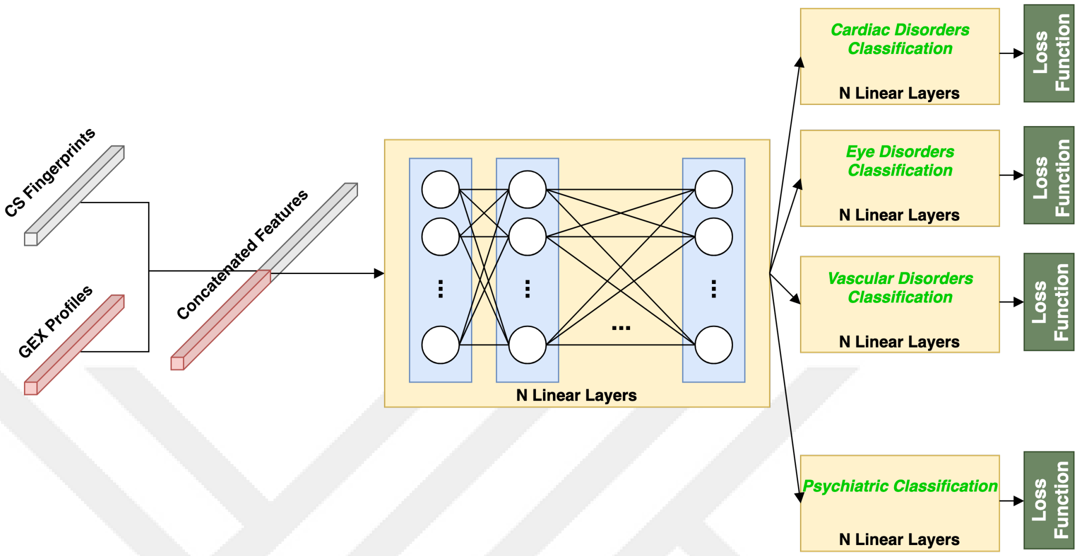


Figure 3.3: Multitask Neural Network architecture that shares weights in common layers and each of the side effect groups has its own classifier.

Each task-specific network is designed as a MLP. Each task is defined by using drug's parent group in the ADReCS ontology tree [45]. Thus, similar side effects are grouped together in the same tasks. There are 24 parent groups that are obtained from ADReCS[45] according to 1052 side effects. Similar side effects, intuitively, needs almost similar features. Task-specific MLP networks use the common feature which is extracted from the base network to predict side effects between similar ones. Some of side effects can be grouped under different parent groups like nausea. While stomach disorders can be the cause of nausea, nausea can be also observed because of dizziness. For such cases, our model predicts more than one probability for the same side effect from different tasks. Our final predictions for the side effects are calculated by taking the maximum probability for the same side effect, which can be observed from different tasks.

3.2.5 SMILES Convolutional Network - SMILESConv

Convolutional neural networks(CNN) work well for the computer vision problems which can be solved by learning internal features from 2D images [34]. 2D Convolutional kernels aim to extract relations between pixels by moving over neighbor pixels. 1D convolutional neural networks follow the same rule. Each character is represented as 1D vector which can be one-hot vector or learned embedding vector and 1D convolutional kernels are moved over these vectors. In this way, kernels learn features from the string by seeking relations between consecutive characters. SMILESConv architecture is shown in Figure 3.4.

Our network contains 200 1D-convolutional layers which have different kernel sizes between 1 and 200 sequentially. Each convolutional layer has 32 output channels. After each convolutional layer [34], batch normalization [42], ReLU activation functions [43], and max-pooling are applied. The size of the pooling operations is equal to that of the feature map that has been extracted after convolution, batch normalization, and ReLU operations. Max pooling is used to vectorize the feature maps. Each vector is concatenated to pass through linear classification layers. Extracted feature vector has 6400 units which come from 32*200. Before the classification layers, dropout is applied to the extracted feature vectors. Linear layers in the classification block contain 2000 units. Batch normalization and ReLU activation follow each linear layer. The sigmoid activation function is applied to the output layer, which contains 1042 units.

3.2.6 Edge Convolutional Neural Network - EdgeConv

SMILES Strings dataset contains 12 unique atoms. For a single drug and its single atom-atom bond, the atomic relations are encoded in a vector which has a dimension of 60. All the atom-atom relations in a single drug are represented with a $60 \times 12 \times 12$ 3D tensor. When we take image as an input of the convolutional neural networks, convolutional kernel extracts the relation between the center pixel of the kernel and its neighbor pixels. In our case, instead of pixel positions, we encoded drug features according to atoms in drugs since we need to aggregate atom feature information of each atom in drug graph with its neighbor atoms and atom-atom bond features.

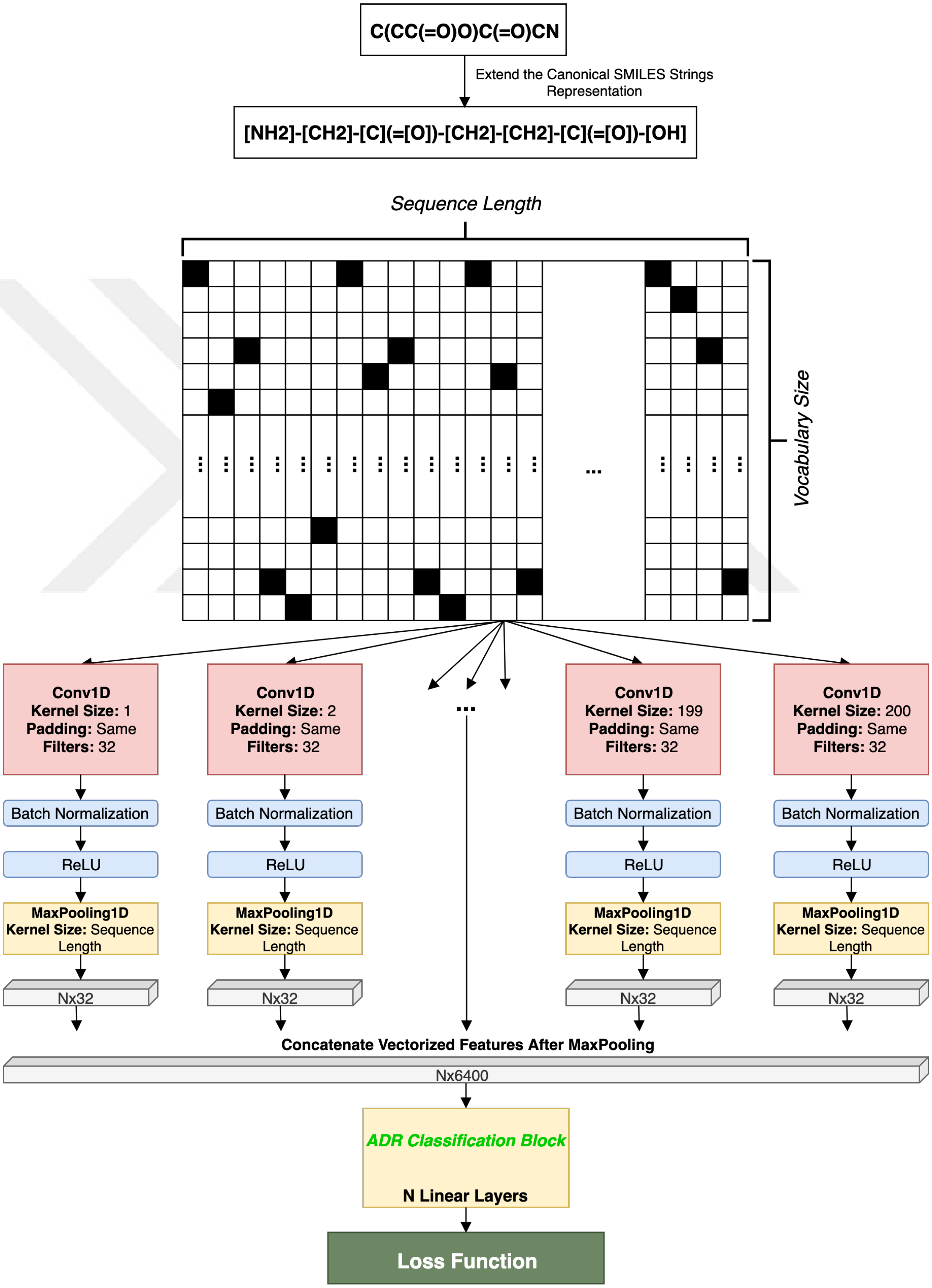

C(CC(=O)O)C(=O)CN
Extend the Canonical SMILES Strings Representation
[NH2]-[CH2]-[C](=[O])-[CH2]-[CH2]-[C](=[O])-[OH]
Sequence Length
Vocabulary Size
Conv1D Kernel Size: 1 Padding: Same Filters: 32
Conv1D Kernel Size: 2 Padding: Same Filters: 32
Conv1D Kernel Size: 199 Padding: Same Filters: 32
Conv1D Kernel Size: 200 Padding: Same Filters: 32
Batch Normalization
ReLU
MaxPooling1D Kernel Size: Sequence Length
Nx32
Concatenate Vectorized Features After MaxPooling
Nx6400
ADR Classification Block
N Linear Layers
Loss Function

Figure 3.4: Architecture figure of the SMILES Convolutional Neural Network.

Since drug molecules contain atoms and bonds between atoms, drugs can be represented with a graph structure. Because of that fact, applying graph convolutional neural network (GCN) is a natural approach. We apply GCN [41] onto the drug graph dataset, which is different from the atomic bond dataset. The drug graph dataset contains atoms and bonds between atoms as nodes and edges, respectively. However, the edges between the nodes do not contain any information about bond properties. Edges are only used to describe neighbor connectivity in a drug graph.

EdgeConv network aggregates atom features with all the neighbor atoms and bonds between them by using a 2D-convolution operator. Since we rely on features between atoms, we call this method as EdgeConv. There are 3 convolutional layers which have kernel sizes of 5x5, 2-padding and a stride of 1 at the beginning of the network. Batch normalization [42] and ReLU activation function [43] follow each convolution layers. After convolutional blocks, an adaptive max pooling is applied to vectorize the extracted features. The vectorized features are then passed through linear layers. The number of input channels of the first convolutional layer is equal to the channel size of the atomic bond dataset, which is 60. Each of the convolutional layers contains 1024 filters. Linear layers at the end of the network contain 4096 units. The output layer has a length of 1042, the number of side effects. EdgeConv architecture is shown in Figure 3.5.

3.3 Implementation Details

Each neural network model is evaluated with a 3-fold cross validation. We eliminate ADRs which has less than 10 drug associations in training and validation sets for each fold. 1052 ADRs are left after this preprocessing step. We use the same 1052 ADRs for each different dataset type. The neural network models that are trained with GEX dataset, are evaluated with two different validation sets. One of these validation sets is GE profile replicates of 6k drugs/small molecule

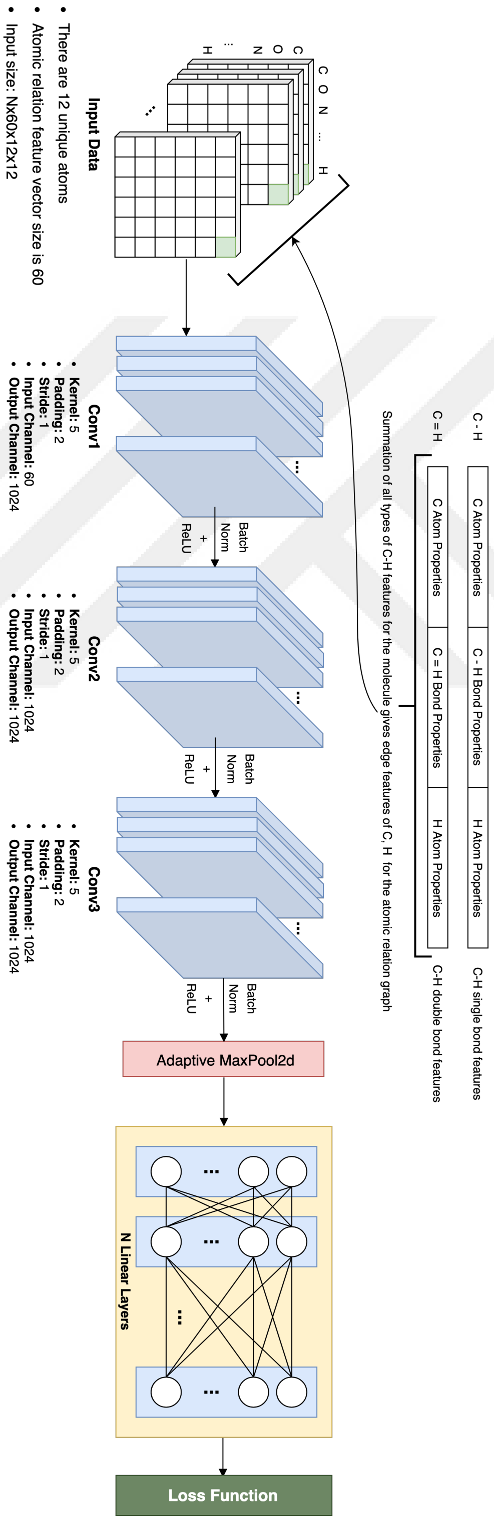


Figure 3.5: Architecture of the Edge Convolutional Neural Network.

compounds in different conditions. The other one contains 263 samples which carry the strongest signatures of GE profiles for the given drugs/small molecule compounds.

MLP [33] models comprise blocks which contain linear layer, batch normalization layer [42], ReLU activation [43], and dropout layers laid out in the said order. The probability of an element to be zeroed for the dropout layers [47] is set to 0.2. The number of hidden layers and the number of neurons that exist in hidden layers affect the performance of the model directly. These numbers are described in Table 4.1. GE, CS, GE+CS, and GE+CS+META datasets are applied to the MLP model. The best performance result for the MLP model is obtained by using the CS dataset for the 263 validation samples with strongest signatures.

MMNN and MTL models have the same linear blocks as in the MLP models but deployed differently when building up the network architecture as explained previously in sections 3.2.1 and 3.2.3.

We use binary cross entropy loss functions and an Adam optimizer for training the neural networks. Initial learning rate for Adam optimizer differs depending on the models and datasets. We decrease the initial learning rate by a factor of 0.1 at the steps 300 and 400.

Chapter 4

Results

The predictive performance of our methods is tested with different performance measurements. We compare our proposed models against the work of [23], which predicts the same side effects from the same drugs of our working dataset. The best method among our proposals is also trained with the weighted loss to mitigate the imbalanced dataset problem. All of the comparisons and the pertinent results are detailed below in this section.

4.1 Performance Evaluation Metrics

3-fold cross-validation is performed to test our proposed models as in the base model [23] for holding meaningful comparisons. Our generated fold splits are different than the base model [23] since they did not provide their splits. Nevertheless, each proposed model is trained with the same splits for comparative fairness among our models. Side effect prediction task is multi-label classification problem since more than one side effects can be positive for the same drug at the same time. Micro-averaged Area Under Curve(AUC), micro-averaged mean Average Precision(mAP) and Hamming Loss consider all predictions and labels globally, hence the reason that these metrics are used to calculate the performance of the classifiers based on individual prediction for each of drugs and its side effect label. On the other hand, macro-averaged Area Under Curve(AUC) and macro-averaged mean Average Precision(mAP) are employed to calculate side effect based performance. Since these metrics calculate each of the classes performances independently we take the average of them.

As we discussed in the Dataset section 3.1, CNN-based models are trained with the subset of the drugs which is used to train Multi-Layer Perceptron based models since some of the drugs are pruned according to the dimension of the SMILES String representation's characters. Therefore Multi-Layer Perceptron based models are compared with each other. The best model among the MLP models is selected according to its performance. This model is trained from scratch with the dataset which is used to train the CNN based models to hold a fair comparison between the MLP and CNN based models.

Some of the side effects in this dataset are rare while some of them are commonly observed for many drugs. Presence of such an imbalance of labels is one of the major challenges in this research. The data imbalance renders the learning of more common side effects easier in comparison to the rare side effects by the neural network since the more common side effects have more associated drugs than the rare side effects. Class weighted loss function is used to train the neural network to mitigate the imbalanced dataset problem. The class weighted loss function is used to train only the best model among all of MLP and CNN based models to circumvent the imbalanced dataset problem and the performance results obtained from both the class weighted and the unweighted trainings are presented.

4.2 Performances of Multi-Layer Perceptron Based Architectures

Multi-Layer Perceptron (MLP), Residual Multi-Layer Perceptron (ResMLP), Multi-Modal Neural Network (MMNN) and Multitask Neural Network (MNN) structures are built based on the Multi-Layer Perceptron architecture that contains only linear layers. These models are trained with CS and GEX datasets. These datasets contain 791 drugs and 1052 side effects. In the GEX dataset, each of drugs has more than one experiment which is treated on different cells at different times with different dosages. In table 4.1, results obtained from MLP based models are represented. The GEX-18K training set consists of all experiments of the 791 drugs. The GEX-791 training set contains only the strongest experiments for each of the drugs which are defined for the meta-data provided in LINCS L1000 dataset.

Using the CS fingerprints of the 791 drugs with basic MLP produces significant predictive performance even though it is a basic network architecture. MLP model which is trained with CS-791 dataset yields better performance than our base results which are reported by [23]. To improve over this method, we incorporate more complex features by adding GEX profiles along with their experimental META information to the CS fingerprints. According to our results, multi-modal learning approach with two branches, one of which is for the concatenated GEX profiles and META information and the other branch is for the CS fingerprints, produces best predictive performance for all the tested metrics except the micro mean average precision (micro mAP).

4.3 Performances of Convolutional Neural Network Based Architectures

As discussed previously, convolutional models are trained with the subset of the 791 drugs since there exist limitations on the character length of the SMILES string representations. After the necessary pruning processes, 615 drugs and 1042 side effects are retained. The datasets' statistics are represented in Figure 4.1. To compare MLP and Conv based architectures, models are trained from scratch but this time by using the 615 drugs and trying to predict 1042 side effects. Table 4.2 shows that using SMILESConv increases performance significantly in all the tested metrics compared to the all other types of network architecture. EdgeConv method produces the worst results. However, we keep in mind that EdgeConv is a novel architecture which relies on basic features and has a simple convolutional neural network architecture. Nevertheless, this improvable method can achieve average success when compared to the other types of architectures. Finally, the graph neural network-based models produce nearly random prediction performance. Since the main disadvantage of this method is that it does not account for the properties of the bonds between atoms, it is not possible to get high predictive performance from GCN [41].

4.4 Performances of Class Weighted Training

Since a single drug can have more than a single side effect at the same time, binary cross-entropy loss function is used to compute the cost for the wrong predictions. However, the label set is imbalanced and this makes it more difficult to learn the rarer side effects than the more common ones. To balance the importance of the classes, we deploy a class-weighted binary cross entropy(WBCE) as the loss function. SMILESConv is trained with class-weighted binary cross-entropy loss function to investigate the effectiveness of the imbalanced labels on the final

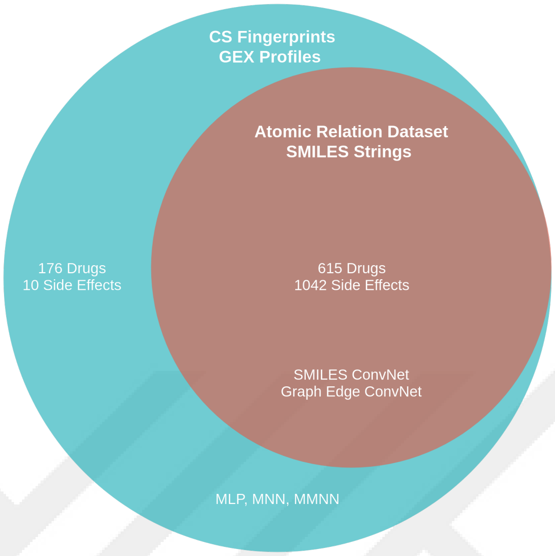


Figure 4.1: The number of samples in datasets according to method used.

performance. Slight performance increase is observed with class-weighted training in all the metrics except for the hamming loss. Comparison of with/without class-weighted training performances are represented as shown in Table 4.3

Model	Dataset	# train	# test	GEX - CS neurons	layers	Macro AUC	Micro AUC	Macro mAP	Micro mAP	Hamming
Ma'ayan [23]	GO+CS	528	263	-	-	0.679	0.854	-	-	0.083
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	CS	528	263	800	3 - 1	0.784	0.866	**0.457**	0.578	0.072
ResMLP	CS	528	263	800	101 - 1	0.768	0.843	0.428	0.520	0.077
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
MTL	[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	0.542	0.079
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
MMSum	[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**
MMSum	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
MMSum	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

Table 4.1: Performance comparison between MLP models by using GEX, CS and META datasets. $X\&Y$ 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.

Model	**Dataset**	**# 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
MMSum	[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
EdgeConv	Atomic Relation Graph	410	205	0.776	0.847	0.467	0.559	0.084
SMILESConv	SMILES String	410	205	**0.821**	**0.881**	**0.496**	**0.596**	**0.075**

Table 4.2: Performance comparison between MLP and Conv models which are trained with 615 drugs for the 1042 side effects. $[X, Y]$ 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.

Model	**Dataset**	**# train**	**# test**	**Macro AUC**	**Micro AUC**	**Macro mAP**	**Micro mAP**	**Hamming**
SMILESConv without class weights	SMILES String	410	205	0.821	0.881	0.496	0.596	**0.075**
SMILESConv with class weights	SMILES String	410	205	**0.826**	**0.891**	**0.512**	**0.605**	0.085

Table 4.3: Performance impact of the class-weighted training for the best model.

4.5 The Most and The Least Predictable ADRs

According to our results of the experiments, the most predictive features are obtained from the chemical structure of the drugs and the most predictive model is SMILESConv. Because of these facts, we use predictions of SMILESConv and MLP models to make a class-based comparison. In addition to that, we visualize the effect of the using weighted binary cross-entropy(WBCE) for the SMILESConv model. Figures 4.2, 4.3, and 4.4 show the changes in the AUC scores according to the number of positive samples of the side-effects. Blue lines in these figures represent the function of the regression model that fitted by using positive number count and auc scores. These regression functions indicate that the AUC score decreases as the number of positive samples increases. The slope of regression which is calculated for the SMILESConv model trained with weighted loss is bigger than the slope of the regression function of SMILESConv trained with unweighted loss. This fact shows that when we train our classifier by using class weights, then prediction performance is increasing for the rare classes. Moreover, when we look at the mean AUC score in the Figure 4.2 and Figure 4.3, although the patterns of the indicative points of the side-effects in these Figures are similar, we see that the SMILESConv model raises side-effect prediction.

In Table 4.4 and Table 4.5, we present top-10 side effects that are predicted easily and hardly by MLP, SMILESConv trained with BCE and SMILESConv trained with Weighted-BCE models.

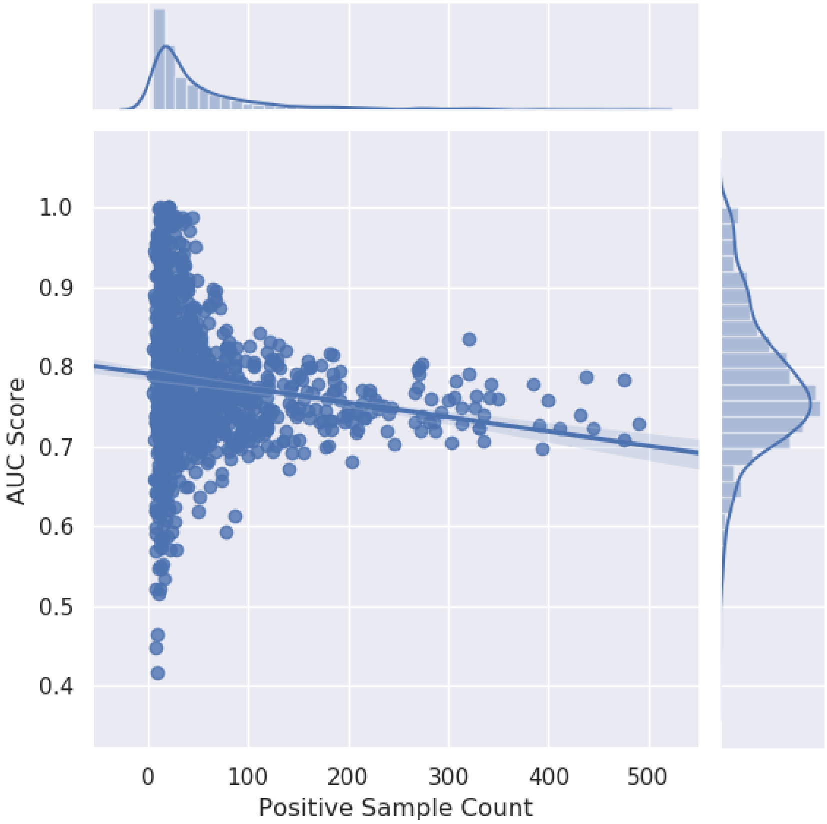


Figure 4.2: Side effect AUC scores distribution based on the number of positive samples. The model is the MLP model that uses CS features.

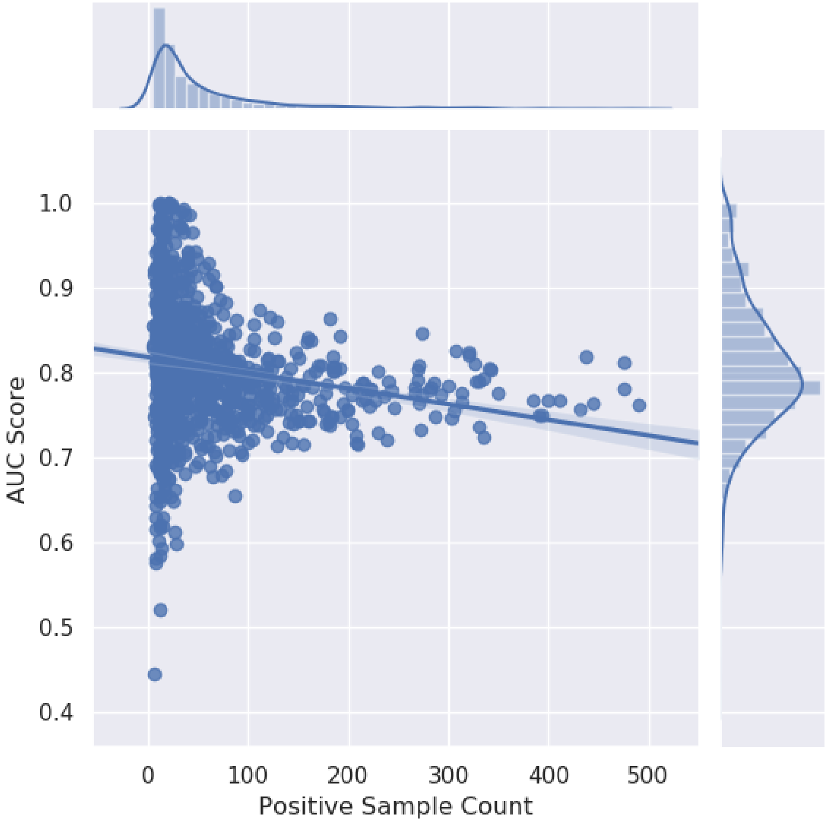


Figure 4.3: Side effect AUC scores distribution based on the number of positive samples. The model is the SMILESConv model trained with binary-cross-entropy(BCE).

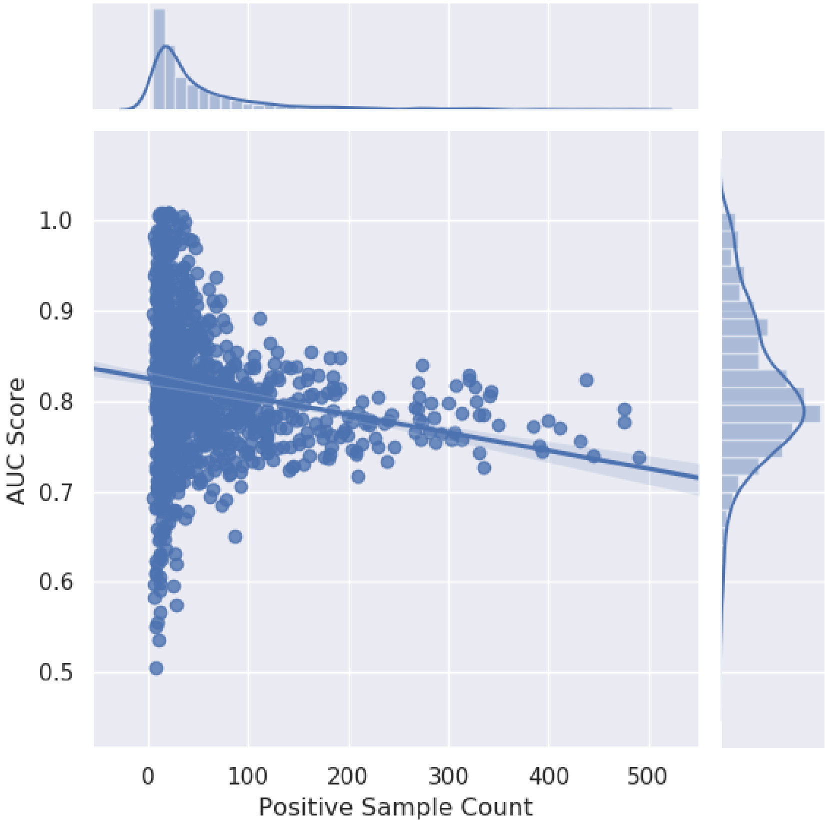


Figure 4.4: Side effect AUC scores distribution based on the number of positive samples. The model is the SMILESConv model trained with weighted binary cross entropy(WBCE).

4.6 Model Performance Comparison Based on Proposed Drug-SE Relations

In this section we compare MLP, MMSum and SMILESConv models with respect to their true positive and false positive predictions. These models are trained and tested with the same dataset which contains 615 drugs. From 615 drugs and 1042 side effects, there exist only 62579 positive drug-SE relations in the dataset. The number of positive drug-SE relation indicates how the dataset is imbalanced since only 10% of the dataset have positive relations. 4.6 shows the number of TP and FP predictions of the MLP, MMSum and SMILESConv models. When we analyze results, it is obvious that the SMILESConv model can predict better than the other architectures with respect to the number of TP and FP predictions. Training with weighted binary cross-entropy is increasing TP performance while it is decreasing FP performance.

Side Effect	# positive samples	MLP AUC	SMILESConv_bce AUC	SMILESConv_weighted_bce AUC
Skin test positive	21	1.00	1.00	1.00
Cushing's syndrome	12	1.00	1.00	1.00
Myocardial rupture	19	1.00	1.00	1.00
Alkalosis hypokalaemic	21	1.00	1.00	1.00
Fat embolism	15	1.00	1.00	1.00
Muscle mass	21	1.00	1.00	1.00
Coombs direct test positive	10	1.00	1.00	1.00
Paraplegia	17	1.00	0.99	1.00
Lupus miliaris disseminatus faciei	23	0.99	1.00	1.00
Nitrogen balance	23	0.99	0.99	1.00

Table 4.4: The top 10-side effects predicted with the highest performance. Number of positive samples column indicates the number of drugs annotated with the side effect.

Side Effect	# positive samples	MLP AUC	SMILESConv_bce AUC	SMILESConv_weighted_bce AUC
Skin burning sensation	7	0.45	0.64	0.54
Panic attack	9	0.47	0.66	0.55
Tachypnoea	10	0.52	0.60	0.64
Sensory disturbance	8	0.52	0.57	0.50
Hepatitis fulminant	11	0.57	0.65	0.58
Ear disorder	28	0.57	0.60	0.57
Arrhythmia supraventricular	15	0.59	0.62	0.65
Respiratory disorder	87	0.61	0.65	0.64
Personality disorder	26	0.61	0.61	0.62
Congenital eye disorder	11	0.62	0.65	0.62

Table 4.5: Side effects with the lowest performance. Number of positive samples column indicates the number of drugs annotated with the side effect.

Model	# true positive samples	# false positive samples
MLP_BCE	24730	19400
MLP_WBCE	26951	20762
MMSum_BCE	25535	18235
MMSum_WBCE	28535	21512
SMILESConv_BCE	27435	12652
SMILESConv_WBCE	32026	18237

Table 4.6: TP and FP prediction results of the MLP, MMSum and SMILESConv models trained with BCE (binary cross-entropy) and WBCE (weighted binary cross-entropy). Probability threshold: 0.5

Chapter 5

Discussion

The pharmaceutical drug development process is a long and demanding process. Unforeseen ADRs that arise at the drug development process can stop or restart the whole development pipeline. Therefore, the a priori prediction of the ADRs/side effects of the drug at the design phase is critical. At this point, our proposed method SMILEConv can predict ADRs/side effects with high accuracy by using the SMILES data structure which is a string representation of the drug chemical structure.

ADRs can also be seen due to some patient related conditions such as drug intolerance or the dosing. In this case, 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 context-related features are obtained from the LINCS L1000 project which contains hundreds of thousands treatment experiments for the thousands of the existing drugs/small molecules in varying conditions. SIDER dataset is used as our side effect label set. Since the SIDER contains approved side effects without depending on patient-related conditions, we treat GE samples which are obtained from different treatment experiments of the same drug as if they had the same side effects. While the LINCS L1000 dataset is large-scale and contains many condition-dependent examples, our proposed model (MMNN) uses GE and CS features as combined features and achieves better accuracy performance compared to the models that use the only chemical structure (CS) fingerprints. The reported accuracy is noteworthy considering that we are only trying to estimate the condition-independent side effects.

The other contribution of this study is that we propose a novel approach based on a Edge Convolutional Neural Network for the side effect prediction by using atom-atom relations including the bond type properties between the neighbor atoms. Extracting the atom-atom relation matrix and applying a convolutional neural network on this dataset is a new approach. EdgeConv is not the model that produces the best result between other models. Nevertheless, this method is open to further improvements in many ways.

There are many things that can be done as future work to attain better accuracy. The most important one of them is to enlarge the data set by digging the other side effects datasets and adding the Phase-2 portion of the LINCS L1000 project. For the SMILESConv method, adding a trained embedding which is trained for the SMILES string character prediction task at the beginning of the network can also increase the performance of the SMILESConv method. As we discussed before, the EdgeConv method is open to development in many directions such as creating a richer and denser atom-atom relation matrix and deploying a more complex convolutional network.

Bibliography

[1] A. Karpathy, "Convolutional neural networks." `http://cs231n.github.io/convolutional-networks`. Accessed: 2019-09-01.

[2] J. A. DiMasi, H. G. Grabowski, and R. W. Hansen, "Innovation in the pharmaceutical industry: new estimates of r&d costs," *Journal of health economics*, vol. 47, pp. 20–33, 2016.

[3] E. Stopke and J. Burns, "New drug and biologic r&d success rates, 2004-2014," *PAREXEL?s Bio/Pharmaceutical R&D Statistical Sourcebook*, vol. 2016, 2015.

[4] L. V. Sacks, H. H. Shamsuddin, Y. I. Yasinskaya, K. Bouri, M. L. Lanthier, and R. E. Sherman, "Scientific and regulatory reasons for delay and denial of fda approval of initial applications for new drugs, 2000-2012," *Jama*, vol. 311, no. 4, pp. 378–384, 2014.

[5] A. G. P. S. Pretorius, "Phase iii trial failures: Costly, but preventable," *Applied Clinical Trials*, vol. 25, no. 8, 2016.

[6] J. DiMasi, "Causes of clinical failures vary widely by therapeutic class, phase of study," *Tufts CSDD Impact Report*, vol. 15, no. 5, pp. 1–4, 2013.

[7] T. J. Hwang, D. Carpenter, J. C. Lauffenburger, B. Wang, J. M. Franklin, and A. S. Kesselheim, "Failure of investigational drugs in late-stage clinical development and publication of trial results," *JAMA internal medicine*, vol. 176, no. 12, pp. 1826–1833, 2016.

[8] C. Schmidt, "The struggle to do no harm in clinical trials.," *Nature*, vol. 552, no. 7685, pp. S74–S75, 2017.

[9] J. Scheiber, J. L. Jenkins, S. C. K. Sukuru, A. Bender, D. Mikhailov, M. Milik, K. Azzaoui, S. Whitebread, J. Hamon, L. Urban, *et al.*, "Mapping adverse drug reactions in chemical space," *Journal of medicinal chemistry*, vol. 52, no. 9, pp. 3103–3107, 2009.

[10] N. Atias and R. Sharan, "An algorithmic framework for predicting side effects of drugs," *Journal of Computational Biology*, vol. 18, no. 3, pp. 207–218, 2011.

[11] E. Pauwels, V. Stoven, and Y. Yamanishi, "Predicting drug side-effect profiles: a chemical fragment-based approach," *BMC bioinformatics*, vol. 12, no. 1, p. 169, 2011.

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

[13] 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.

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

[15] M. Liu, Y. Wu, Y. Chen, J. Sun, Z. Zhao, X.-w. Chen, M. E. Matheny, and H. Xu, "Large-scale prediction of adverse drug reactions using chemical, biological, and phenotypic properties of drugs," *Journal of the American Medical Informatics Association*, vol. 19, no. e1, pp. e28–e35, 2012.

[16] 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.

[17] E. Bresso, R. Grisoni, G. Marchetti, A. S. Karaboga, M. Souchet, M.-D. Devignes, and M. Smaïl-Tabbone, "Integrative relational machine-learning

for understanding drug side-effect profiles," *BMC bioinformatics*, vol. 14, no. 1, p. 207, 2013.

[18] 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.

[19] 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 computational biology*, vol. 5, no. 5, p. e1000387, 2009.

[20] H. Zhou, M. Gao, and J. Skolnick, "Comprehensive prediction of drug-protein interactions and side effects for the human proteome," *Scientific reports*, vol. 5, p. 11090, 2015.

[21] D. C. Zielinski, F. V. Filipp, A. Bordbar, K. Jensen, J. W. Smith, M. J. Herrgard, M. L. Mo, and B. O. Palsson, "Pharmacogenomic and clinical data link non-pharmacokinetic metabolic dysregulation to drug side effect pathogenesis," *Nature communications*, vol. 6, p. 7101, 2015.

[22] I. Shaked, M. A. Oberhardt, N. Atias, R. Sharan, and E. Ruppin, "Metabolic network prediction of drug side effects," *Cell systems*, vol. 2, no. 3, pp. 209–213, 2016.

[23] 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.

[24] M. Kuhn, I. Letunic, L. J. Jensen, and P. Bork, "The sider database of drugs and side effects," *Nucleic acids research*, vol. 44, no. D1, pp. D1075–D1079, 2015.

[25] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison, "Open babel: An open chemical toolbox," *Journal of Cheminformatics*, vol. 3, p. 33, Oct 2011.

[26] G. M. Dimitri and P. Lió, "Drugclust: A machine learning approach for drugs side effects prediction," *Computational Biology and Chemistry*, vol. 68, pp. 204 – 210, 2017.

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

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

[29] 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 Medical Informatics*, vol. 21, 05 2018.

[30] S. Dey, H. Luo, A. Fokoue, J. Hu, and P. Zhang, "Predicting adverse drug reactions through interpretable deep learning framework," *BMC Bioinformatics*, vol. 19, 12 2018.

[31] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *Journal of Chemical Information and Modeling*, vol. 50, no. 5, pp. 742–754, 2010. PMID: 20426451.

[32] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pp. II–1188–II–1196, JMLR.org, 2014.

[33] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Transactions on Neural Networks*, vol. 3, pp. 683–697, Sep. 1992.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, (USA), pp. 1097–1105, Curran Associates Inc., 2012.

[35] X. Zhang, J. J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” *CoRR*, vol. abs/1509.01626, 2015.

[36] M.-C. Cai, Q. Xu, Y.-J. Pan, W. Pan, N. Ji, Y.-B. Li, H.-J. Jin, K. Liu, and Z.-L. Ji, “Adrecs: an ontology database for aiding standardization and hierarchical classification of adverse drug reaction terms,” *Nucleic Acids Research*, vol. 43, no. D1, pp. D907–D913, 2015.

[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 Bioinformatics*, vol. 19, no. 19, p. 526, 2018.

[38] S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker, P. A. Thiessen, B. Yu, L. Zaslavsky, J. Zhang, and E. E. Bolton, “Pubchem 2019 update: improved access to chemical data,” *Nucleic acids research*, vol. 47, pp. D1102–D1109, Jan 2019. 30371825[pmid].

[39] G. Landrum, “Rdkit: open-source cheminformatics.” `http://www.rdkit.org/`. Accessed: 2019-09-01.

[40] C. Shang, Q. Liu, K.-S. Chen, J. Sun, J. Lu, J. Yi, and J. Bi, “Edge attention-based multi-relational graph convolutional networks,” *ArXiv*, vol. abs/1802.04944, 2018.

[41] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv e-prints*, p. arXiv:1609.02907, Sep 2016.

[42] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *CoRR*, vol. abs/1502.03167, 2015.

[43] Y. Li and Y. Yuan, “Convergence analysis of two-layer neural networks with relu activation,” in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 597–607, Curran Associates, Inc., 2017.

[44] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[45] M. Cai, Q. Xu, Y.-J. Pan, W. Pan, N. Ji, Y.-B. Li, H. Jin, K. Liu, and Z.-L. Ji, “Adrecs: An ontology database for aiding standardization and hierarchical classification of adverse drug reaction terms,” *Nucleic acids research*, vol. 43, 10 2014.

[46] S. Ruder, “An overview of multi-task learning in deep neural networks,” *CoRR*, vol. abs/1706.05098, 2017.

[47] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.