Received: 25 October 2020 | Revised: 26 January 2021 | Accepted: 3 February 2021

DOI: 10.1002/int.22389

RESEARCH ARTICLE

WILEY

Descriptive prediction of drug side-effects using a hybrid deep learning model

Chun Yen Lee | **Yi-Ping Phoebe Chen**

Department of Computer Science and Information Technology, La Trobe University, Melbourne, Australia

Correspondence
Yi-Ping Phoebe Chen, Department of Computer Science and Information Technology, La Trobe University, Melbourne 3086, Australia.
Email: phoebe.chen@latrobe.edu.au

Abstract

In this study, we developed a hybrid deep learning (DL) model, which is one of the first interpretable hybrid DL models with Inception modules, to give a descriptive prediction of drug side-effects. The model consists of a graph convolutional neural network (GCNN) with Inception modules to allow more efficient learning of drug molecular features and bidirectional long short-term memory (BiLSTM) recurrent neural networks to associate drug structure with its associated side effects. The outputs from the two networks (GCNN and BiLSTM) are then concatenated and a fully connected network is used to predict the side effects of drugs. Our model achieves an AUC score of 0.846 irrespective of what classification threshold is chosen. It has a precision score of 0.925 and the Bilingual Evaluation Understudy (BLEU) scores obtained were 0.973, 0.938, 0.927, and 0.318 which show significant achievements despite the fact that a small drug data set is used for adverse drug reaction (ADR) prediction. Moreover, the model is capable of accurately structuring correct words to describe drug side-effects and associates them with its drug name and molecular structure. The predicted drug structure and ADR relation will provide a reference for preclinical safety pharmacology studies and facilitate the identification of ADRs during early phases of drug development. It can also help detect unknown ADRs embedded in

existing drugs, hence contributing significantly to the science of pharmacovigilance.

KEYWORDS
deep learning, drug molecular structure, drug side-effects, pharmacovigilance, prediction

1 | INTRODUCTION

An overwhelming number of drugs enter clinical trials but fail to obtain approval for commercialization. To ensure a successful introduction of new drugs, prediction of adverse drug reactions (ADRs) during the drug discovery process is very important.

ADRs are negative reactions to a consumed medication. Some ADRs can be disastrous, bring great risks to human body, may cause hospitalizations and numerous deaths. Occurrence of ADRs has led to attrition of many drugs either in the preclinical stage or the postmarketing stage. Some of the primary root-causes of incidents can be due to interaction of drugs with off-targets or activation of a receptor by its metabolites to induce undesirable ADRs. For example, the appetite suppressant fenfluramine–phentermine (fen–phen) was withdrawn from the market after the deaths of numerous patients. This was due to activation of the 5-hydroxytryptamine-2B (5-HT2B) receptor by one of its metabolites, norfenfluramine, leading to proliferative valvular heart disease.[1,2] Similarly, antihistamine terfenadine has been withdrawn because it caused arrhythmias and death due to the off-target inhibition of the human ether-a-go-go-related gene (hERG).[3]

Experimental method has been used to identify potential adverse drug side-effects; however, this approach is highly expert-dependent, extremely costly, time-consuming, and laborious. Most Food and Drug Administration approvals for first-in-class drugs usually originate from phenotypic screening. Even though these drugs have gone through phenotypic screening, very often, precise mechanisms of actions or molecular on or off-targets were elaborated much later.[4] For example, it took nearly a century to elucidate the mechanisms of actions and molecular targets of aspirin (acetylsalicylic acid).[4] The costly and long experimental process has caused many researchers and drug manufacturing companies to turn their attention to computational pharmacology which aims at using data to predict and understand how drugs affect human body to support decision making in drug discovery, improve clinical practice and avoid unwanted side effects.[5]

Traditional machine learning approaches use the principle that drugs with similar drug structures and properties tend to share similar target proteins and vice versa. On the basis of this principle, traditional machine learning algorithms have been adopted as a computational prediction engine and the prediction is formulated as a binary classification.[5] Most of these approaches rely heavily on handcrafted features and the process tends to be expensive. Moreover, they give uninterpretable prediction on drug–ADR relationships.

The latest machine learning method, deep learning (DL), outperforms traditional machine learning as it requires minimal feature engineering and it leverages representation learning by learning directly from raw data. Inspired by the achievements of DL in de novo molecular design[6–8] and spurred by the interest in addressing challenges associated with the "Black box" issue of DL approaches and the need to use huge data sets to give accurate prediction, we developed a hybrid DL model to assist drug discovery scientists to descriptively predict drug

side-effects using small data set. The hybrid DL model can accurately generate drug molecular structure captioned with drug name and its side effects, thus unravel the "Black box" challenge and improves the interpretable ability of DL. With this new finding, we hope this approach will improve the drug discovery process and hence facilitate the design of new drugs with reduced side effects.

In this paper, we developed the first hybrid DL model, as shown in Figure 1, to predict side effects of drugs. The contribution of this paper can be summarized as follows: (a) this is the first hybrid DL model that combines two different deep neural networks (DNNs) to predict drug side-effects. The model consists of a graph convolutional neural network (GCNN) with Inception modules and a recurrent bidirectional long short-term memory (BiLSTM) neural network; (b) it is the first model that is able to describe drug side-effects in interpretable descriptive language and relate them to drug names as well as their associated structures. The GCNN network extracts features of drugs from drug structures, while the BiLSTM network accurately structures vocabulary for describing drug side-effects and relates the generated keywords to its drug name and drug structure; (c) despite the fact that DL approaches require a large amount of training data to be successful in capturing features and that a small data set can impede the learning, our model is able to mitigate the small data set issue and predict sides effects.

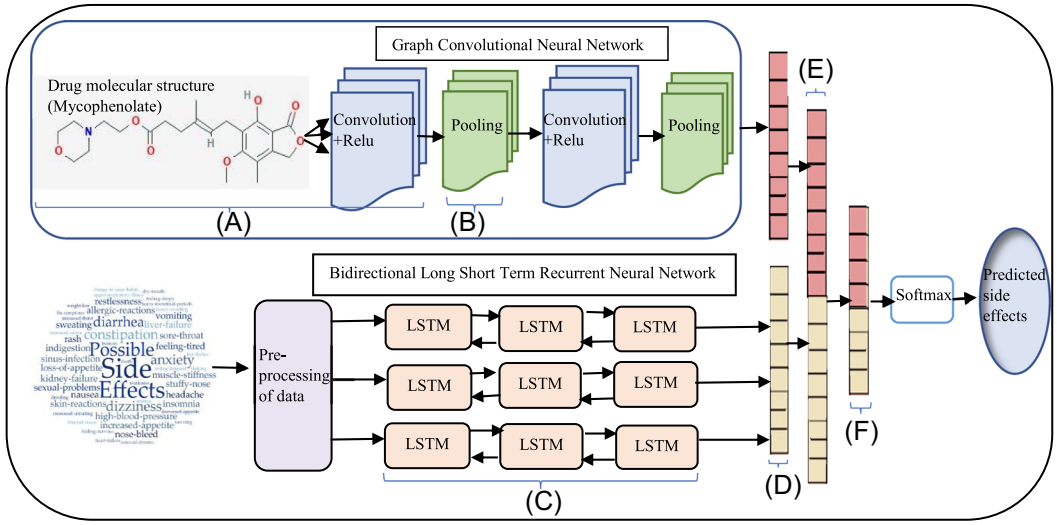

FIGURE 1 The proposed architecture of the hybrid model consists of deep GCNN and BiLSTM networks. (A) SMILES strings are converted into a graph drug molecular structure. Both graph drug molecular structures and drug names are fed directly to the deep CNN incorporated with Inception modules. The GCNN with Inception modules consists of increasing number of filters of different sizes to capture local features of drugs. The number and size of filters in the GCNN increase with the sequence of the layers. (B) This is followed by a pooling layer which focuses on learning the most important drug features. (C) BiLSTM is used because of its ability to consider information in an input sentence and memorize long-term information. BiLSTM receives information from the previous hidden state and current input vector. On the basis of the information received, the input gate, forget gate, and output gate update the memory and emit an output which is passed on to subsequent layers.[9] (D) The outputs from both GCNN and BiLSTM are flattened, (E) concatenated, and (F) fed into fully connected layers and softmax to predict drug side-effects induced by single drugs. BiLSTM, bidirectional long short-term memory; GCNN, graph convolutional neural network; LSTM, long short-term memory; SMILES, simplified molecular input line entry system [Color figure can be viewed at wileyonlinelibrary.com]

With these achievements, we believe our model can detect unknown side effects of existing drugs which have not been previously known and it also has the capability to predict ADRs of new drugs in early stage of the drug development pipeline. Comparatively, the level of dependency on experts for our DL model is much lower because advanced drug molecular structure and knowledge may not be a prerequisite for ADR prediction. Moreover, with the inclusion of Inception modules, the processing time is much shorter, hence it can help reduce the long development process, enhance drug safety, and reduce financial costs.

2 | RELATED WORKS

In the past few years, prominent DL methods, such as convolutional neural networks (CNNs), variants of recurrent neural networks (RNNs), like, long short-term memory (LSTM) neural networks, and gated recurrent units (GRUs), have been widely employed for detection and extraction of adverse side-effects from social media and medical resources with great success.

The variants of RNNs were used to extract medical events and their associated attributes from unstructured text in electronic health records,[10] classify relations from clinical notes,[11] and process text in social media posts.[12] Enhanced CNNs models were used to evaluate Twitter data sets[13] and classify adverse drug events (ADE) in tweets.[14] The multihop self-attention mechanism (MSAM) model which employed multiple vectors for sentence representation was used to detect ADRs from Twitter corpus, PubMed corpus, and PubMed abstracts.[15] In more recent studies, weighted online recurrent extreme learning machine (WOR-ELM)-based method was deployed for ADR extraction from online literature.[16] Separately, Bidirectional Encoder Representations from Transformers (BERT)-based models were developed for ADE extraction from social media data sets.[17]

This new wave has prompted many other researchers to start using DL approaches to predict side effects caused by single drugs[18–20] and induced by drug–drug interaction.[5,21] DL models that utilized chemical, biological, and biomedical information of drugs were used to detect ADRs and predict possible ADRs of new drug.[20] These models, to a certain extent, do not completely take into consideration the three steps which include (a) representing drug molecules in a suitable vector based on drug molecular structures, (b) applying machine learning algorithm on feature space to predict ADRs, and (c) identifying drug molecular structure that is associated with ADRs, hence they are unable to resolve the Black box nature of DL.

Recognizing the importance of using various health data sets to predict ADRs and link ADRs to drug molecular structure, DeepTox, one of the first DNNs, was developed for computational toxicity prediction.[19] The CNN-based model using the simplified molecular input line entry system (SMILES) representation of compounds was also developed for classification of chemical compounds and extraction of chemical motifs (structures).[22] Dey et al.[18] leveraged CNN with "attention" framework to integrate the feature creation and prediction stages into a single system to identify chemical substructures that were associated with ADRs. Their model produced an F1 score of 0.520 and an area under the ROC curve (AUC) of 0.590 for back pain-related ADRs. Even though their DL framework was able to predict drug side-effects based on drug's chemical structure, some of the F1 scores and precision scores obtained were relatively low. To the best of our knowledge, except for these few aforementioned studies, so far, no other studies use DL approaches to identify and associate drug molecular structures with drug side-effects.

3 | ARCHITECTURE OF THE PROPOSED MODEL

3.1 | DL methods

DL has made headlines on healthcare horizon. A CNN is a machine learning model that can detect relevant patterns in data classification and regression.[22,23] It was originally invented for computer vision and was trained to extract images, identify abnormalities in images, and point to areas that need improvement. In recent years, it has shown great success in natural language processing tasks and social media analysis tasks. It has also shown effectiveness in extracting sentence semantics and word information.[24]

An RNN is a powerful model for processing serialized input of an arbitrary length.[25] LSTM is a special RNN structure used to address the exploding or vanishing gradients problem as it possesses a memory cell which is able to store previous information over a long period of time.[25] In the de novo drug design, RNNs containing LSTM cells are able to capture syntax of molecular representation in terms of SMILES strings with good accuracy.[7,26] LSTM networks trained with randomized SMILES were able to generate at least double the number of novel molecules with the same distribution of properties compared with the one trained with canonical SMILES.[27]

Several works have demonstrated the effectiveness of CNN, RNN (with its variants) and ensemble of different kinds of neural networks in computational biology for prediction and regression tasks. CNNs are good at reducing frequency variations; LSTMs are good at temporal modeling, and DNNs are appropriate for mapping features to a more separable space.[28] LSTM has advantages over CNNs because it is able to carry information through infinitely long sequences via memory and it can process sequences of widely varying lengths.[29] DNN, CNN, and LSTM models all have their own unique features, but CNN and LSTM have shown improvements over DNNs across a wide variety of speech recognition tasks.[28] In view of the many interesting properties that one can get from combining CNNs and RNNs and that the combined models make use of both spatial and temporal worlds, in this paper, we present an end-to-end CNN and LSTM model to predict side effects of single drugs.

3.2 | Architecture of our proposed hybrid model

Both CNNs and LSTM have unique characteristics and many studies have proven their effectiveness for prediction. Kwon and Yoon[39] successfully showed that the two-CNN model used for chemical–chemical interaction prediction achieved outstanding performance. The two-CNN model can automatically detect important features without any human supervision. It has been proven that the performance of LSTM could be improved by augmenting it with CNNs for the time series classification problem and image captioning and that the LSTM–CNN model achieved state-of-the-art performance compared with other baseline methods.[29–31] Inspired by their achievements, we propose a hybrid neural network model to predict interpretable drug side-effects.

In this paper, the proposed GCNN–BiLSTM model shown in Figure 1 was trained on only a small data set. The model can learn directly from low-level representations, encodes and compresses them into latent representations that enable DNNs to "understand" molecular representations in drugs, and predict drug side-effects. The architecture of the GCNN–BiLSTM model is elaborated in the following sections.

3.2.1 | Chemcepterization of SMILES into a molecular graph

Our model allows multimodal embedding learning to mediate between two different goals: prediction of drug side-effects and captioning of drug molecular structure with its associated side effects and drug name. As shown in Figure 2, RDKit was used to encode drug molecule directly from SMILES strings into a two-dimensional (2D) drug molecular graph with four channels, where the four channels were used to learn different drug information from the molecule as follows:

*Layer zero was used to encode information about the bonds and bond order.
*Layers 1–3 were used to encode molecular features, such as atomic number, Gasteiger charges, and hybridization.

3.2.2 | Deep convolutional network with Inception modules

Using Keras functional API, a hybrid model is built to support multiple inputs and mixed data types. We used Keras Conv2D function to create a 2D CNN. The GCNN with Inception modules operates over drug molecular images. The Inception modules used three different sized filters (1×1, 3×3, and 5×5) with a stride of 1 in the x and y directions as shown in Figure 2.[32] The Inception modules were incorporated in CNN to enable more efficient computation and learning of local features at different scales (i.e., 1×1, 3×3, and 5×5). The outputs were then combined to keep the number of parameters in the network low. To encode drug properties, the last layer of the first tower was removed, whereas the others were combined with a prior 1×1 convolutional layer. The GCNN systematically applied learned filters to

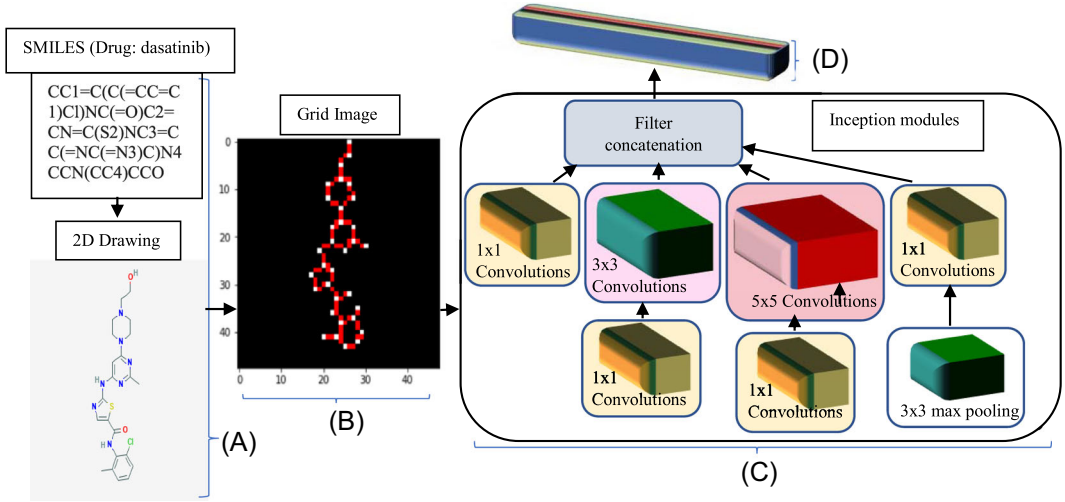

FIGURE 2 The end-to-end workflow which shows (A) SMILES strings are converted into 2D drug molecular graph, (B) which are then mapped onto an input array used to train the GCNN with Inception modules. (C) At every layer, the input data are convolved and pooled as the input of the next layer. (D) Outputs are then concatenated. 2D, two-dimensional; GCNN, graph convolutional neural network; SMILES, simplified molecular input line entry system [Color figure can be viewed at wileyonlinelibrary.com]

input drug structure to create feature maps that summarized the presence of the drug molecular structure and its properties contained in the input drug structure. The first few layers of CNN network learned low-level features of the drug structure while the deeper layers of the neural network learned high-order or more abstract drug properties. The GCNN network views drug molecular images at different spatial levels at the same time by concatenating the data. Different filters focus on learning different features, for example, a particular filter may focus on encoding molecular bonds while other filter may focus on encoding atoms. This architecture enables correlation and development of representations that link the atomic level to the functional group level, then to the fragment level and ultimately to the whole molecule level. All the visual features which carry drug properties are encoded using GCNN. This matrix was run through an activation layer—Rectified Linear Unit (ReLU)—to enable the network to train a lot faster without making a significant difference to accuracy. It also introduced nonlinearity to allow the network to train itself via backpropagation to alleviate the vanishing gradient problem. After the ReLU layer, a pooling layer, also referred to as the down-sampling layer was applied. This layer drastically reduced the spatial dimension (the length and the width change but not the depth) of the input volume and automatically generated a matrix that was much smaller in size than that in the original image. This serves two purposes. The first is that the pooling process further reduces the size of the matrix. This allows the network to train much faster, focusing on processing the most important information in each feature of the drug structure, thus reducing the number of parameters or weights, in return reducing the computation cost. The second is that it controls overfitting to prevent the network from memorizing the training data that it has seen and thus helps the network to generalize the training and validation data well. The MaxPooling2D function was used to add a 2D max-pooling layer with a pooling filter sized 3×3 and a stride of 1 in x and y directions. Max-pooling is used to further reduce the spatial dimensions. We selected the entropy loss function, Adam optimizer, with an automatic adjustable learning rate. As our model was designed for the prediction of drug side-effects, we removed the last layer from the loaded model. The output was then flattened and concatenated before the concatenated layer was connected to a fully connected (FC) layer as shown in Figure 1.

3.2.3 | Recurrent LSTM neural network

RNNs have been widely used for processing long sequences. A variant of RNN, LSTM, is used to overcome the vanishing gradient problem encountered by RNNs.[33–35] Gradient vanishing is a problem in which gradients in the network that is important to change the weights of the network tend to shrink as they backpropagate over time. The BiLSTM network, as shown in Figure 3B, has the ability to resolve the long-term dependencies problem using the gate mechanism and memory cells.

LSTM comprises three different gates and a cell state as shown in Figure 3A. The three different gates are the input gate, forget gate, and output gate; while the cell state is basically the memory of the network which helps overcome the short-term memory of RNNs. When fed into the LSTM network, drug side-effects are transformed into machine-readable vectors and LSTM processes, the sequent and output from the previous hidden state h_{t-1}, are combined to form a vector. The vector goes through tanh activation to regulate the values flowing through the network. The tanh function confines the values to always between -1 and 1. The output is the new hidden state or cell state. The cell state acts as the "memory" of the network and

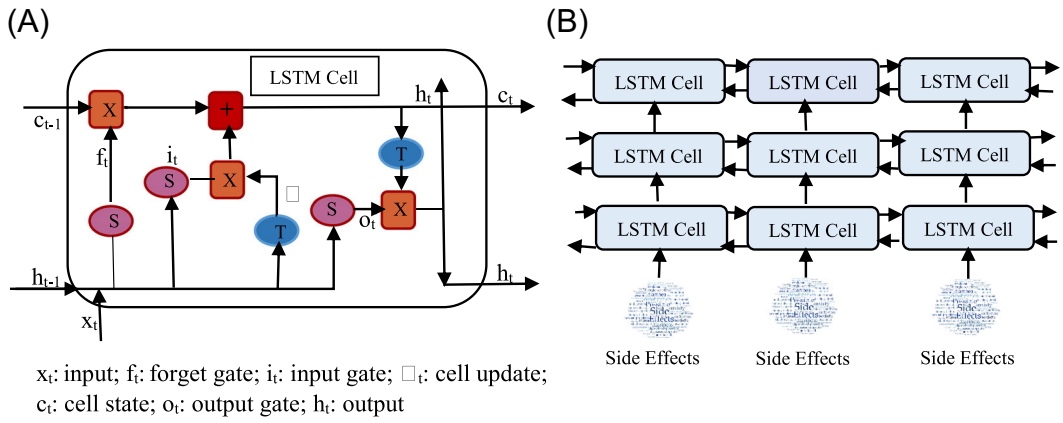

FIGURE 3 (A) The architecture of LSTM cell and its operation. The cell consists of input, forget, and output gates that can regulate the flow of information. These gates learn data in a sequence and decide the importance of the data. They control the decision as to whether to keep or throw away data. In doing so, it can pass relevant information down the long chain of sequences to make predictions. (B) BiLSTM network takes the input in both forward and backward directions [Color figure can be viewed at wileyonlinelibrary.com]

carries relevant information throughout the processing of the sequence. As the cell state goes on its journey, information is added to or removed from the cell state via gates. The gates decide which information is allowed on the cell state.

The input gate has a sigmoid activation that trains the weight that connects the input data to the network nodes. A sigmoid activation is similar to the tanh activation, however, instead of confining the values to between −1 and 1 as in the case for tanh activation, it confines the values between 0 and 1. If the values are close to zero, the input gate will "switch off" certain input data. Conversely, the gate will allow the input data which are close to 1 to pass through it. The tanh output is then multiplied with the sigmoid output. The sigmoid output will decide which information from the tanh output is important to keep. In short, the function of the input gate is to stop new information that flows into the memory cell until it is needed and forget gate will delete the information in the self-recurrent unit, making room for a new memory. The closer the value to 0 means forget, and the closer to 1 means keep. Another gate, known as the output gate, passes the output from the previous hidden state and the current input into a sigmoid function. The newly modified cell state is then passed over to the tanh function where the tanh output is multiplied with the sigmoid output to decide what information the hidden state should carry.

The output is a new cell state and a new hidden state where the values are obtained from combining the outputs of the forget and input gates. The output is carried forward to the next cell. The next cell is obtained by multiplying the previous cell state with the forget vector to drop values in the cell state. Pointwise adding values of the input gate with the cell state, thus generates new essential values of the cell state. The final outputs of BiLSTM and GCNN are then concatenated to predict the side effects of drugs. Mathematically, the processes of BiLSTM can be explained by the following equations:

$$M_t = \tanh(W_{xm}X_t + W_{hm}h_{t-1} + b_m),$$

$$i_t = \sigma(W_{xi}X_t + W_{hi}h_{t-1} + b_i),$$

$$f_t = \sigma(W_{xf}X_t + W_{hf}h_{t-1} + b_f),$$

$$O_t = \sigma(W_{xo}X_t + W_{ho}h_{t-1} + b_o),$$

$$C_t = f_t \theta C_{t-1} + i_t \theta M_t.$$

For each step t, X stands for input vector. The two identifiers, σ and tanh, stand for sigmoid function and hyperbolic tangent function, respectively. On the other hand, θ represents the Hadamard product. The identifiers M_t, i_t, f_t, O_t, C_t, and h_t represent input modulation gate, input gate, forget gate, output gate, cell state, and hidden state, respectively. W_{xm}, W_{xi}, W_{xf}, and W_{xo} represent recurrent weight matrices of the network, while b_m, b_i, b_f, and b_o represent bias vectors.

3.2.4 | Architecture of the GCNN–BiLSTM model

The GCNN–BiLSTM DL model consists of a combination of GCNN and BiLSTM. The networks, as shown in Figure 1, have two inputs, where the graph drug molecular structure and drug names are fed as input data into CNN with Inception modules, while drug names and their side effects are fed as inputs into BiLSTM. Word embedding is used for projecting words with similar meanings into similar representations. To combine the GCNN and BiLSTM streams, the last layer of each stream is flattened and the flattened layers are concatenated. The concatenation fusion method is used because it allows the two networks to have different structures and allows the different neural network designs of the two streams to be customized. The concatenated layer is then connected to an FC layer to further learn the relations between drug molecular structure and their side effects for predicting ADRs as shown in Figure 1. The hybrid model is trained, validated, and the optimum trained hybrid model is then used to generate predicted drug side-effects. Categorical accuracy metrics of multiclass classification of the hybrid model are generated after each epoch. The Bilingual Evaluation Understudy (BLEU) scores can only be generated after obtaining the optimal hybrid model. The predicted side effects are then compared with actual side effects to evaluate the performance of the prediction.

4 | MATERIALS AND METHODS

4.1 | Data sources

We formulated the prediction of drug side-effect problem by extracting drug names, their SMILES strings, and side effects from various popular medical data sources.

4.1.1 | Drug side-effect data sets

We refer to Side Effect Resource (SIDER) database (http://sideeffects.embl.de/) for drug side-effects.[36] (i) The IDs of drug compounds were extracted from PubChem (https://pubchem.ncbi.nlm.nih.gov/) to link drugs to their SMILES. The key publicly available databases used are: DRUG Bank (https://www.drugbank.ca) for a list of drug names, (ii) SIDER (http://sideeffects.embl.de/)

for drug side-effects, (iii) PubChem (https://pubchem.ncbi.nlm.nih.gov/) for compound IDs and SMILES strings. The side effects and drugs are linked by PubChem IDs using SIDER and DRUG Bank associations. A total of 149 SMILES strings are converted into graph drug molecular structures. Drug names and their associated graph drug molecular structures are fed into CNN; while 23,101 drug side-effects together with drug names are fed as input to BiLSTM.

4.1.2 | SMILES strings

SMILES, which uses specific grammar and characters to describe atoms and structure of molecules, is widely recognized and used as a standard representation of a compound for modern chemical information processing.[22] The SMILES format can be encoded into molecular graphs compactly as human-readable strings, like, natural languages. The strings carry identifiers for atoms as well as identifiers denoting topological features, like, bonds, rings, and branches.[37]

It is generally known that chemical structure is closely related to its functions and the relationship between chemical structure and biological activity can be exploited to establish a metabolic pathway and side effects of drugs. Because a considerable number of latent features of chemical compounds exist are represented in SMILES, the SMILES is employed to exploit the relationship between drug structures and drug side-effects.[8] We use canonical SMILES to represent the input of drug compounds. The drug molecules represented by SMILES can be extracted from PubChem (https://pubchem.ncbi.nlm.nih.gov), which is a public repository for information on chemical substances and their biological activities.

4.1.3 | Tokenization and embedding of wordings

For different drug side-effects, the length of side effects is different. To keep the length of side effects consistent and to retain complete information, the side-effect data are prepared by mapping their characters to a one-hot vector matrix and converting them into a fixed-length representation.[4] The length of the longest side-effect word is used to determine the length of a sequence. For example, if the longest sequence is 208 characters in length, then for the length of the side effects which is shorter than 208 characters, the word "startseq" padding is applied to the left while "endseq" is inserted to the right of the sequence to construct a uniform size of 208 characters long. We use the pad_sequences() function in the Keras DL library to pad variable-length sequences to a fixed length. In such a way, all token vectors have the same length.[38]

4.1.4 | Composition of data set

We use a five-fold cross-validation method to find an optimal hyperparameter that shows the best average performance. The data set is randomly divided into six equal parts with 90% of the data used for model selection (training and validation sets) and 10% used for independent testing (test set) as shown in Figure 4. Of the 90% of the data set, 10% of it is used to iteratively cross-validate the data set (validation set). Training and validation processes are carried out independently of the test set to ensure fairness of the independent tests.[23]

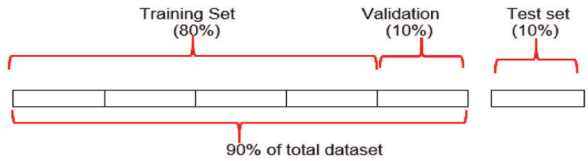

FIGURE 4 Five-fold validation for model selection. The five-fold validation set consists of 80% training data, 10% validation data, and 10% test data. Of the 90% of the training data set, 10% is used for validation. The remaining 10% of the data is used as a test set [Color figure can be viewed at wileyonlinelibrary.com]

5 | EXPERIMENTAL SETTING

In this paper, our hybrid GCNN–BiLSTM model was implemented using the Python 3.7.6 SciPy environment with scikit-learn, Pandas, and Matplotlib installed; while Keras 2.3.1 interfaced with TensorFlow version 2.1.0 was used as the backend. We utilized five-fold validation as described in Figure 4 to train, validate, and test the models.

5.1 | Loading data

There are two input arrays to the hybrid neural network. The side-effect input text was encoded as integers, which was fed to a word embedding layer. The graph drug molecular structure and drug names were fed directly to the CNN for training.

5.2 | Fitting the model

There are no general rules for designing an optimal architecture. We experimented with different architectures and parameters to obtain an optimal combination of the number of filters and filter length for the GCNN network. We tried different types and sizes of network layers for BiLSTM. The list of hyperparameters for this paper is shown in Table 1.

The training process involved feeding training data sets to the networks and optimizing the loss function. The network iterated to update the parameters to reduce loss when more graph drug molecular structures were trained. The model was trained using Adam as the optimization algorithm. During the training, the learning rate was adjusted automatically and continuously at each step of the learning process. Dropout regularization technique was used to reduce overfitting. Early stopping was used during the training to avoid overfitting. Max-pooling was then applied at the end of the process to achieve a better summarization effect.[39] Finally, flattening was adopted to convert the output of the convolutional layers to create a single long feature vector. In other words, we put all the pixel data in one line and made connections with the final layer. To improve the accuracy of our model, we tested the effect of adjusting the learning rate continuously, changed the dropout threshold, batch size, and the number of iterations. We even changed the number of units in our hidden layers to see how different architectures increased or decreased the accuracy of the model.

Experiments were conducted with a wide range of hyperparameters and iteratively validated to fine-tune the model. By comparing the difference in the validation and training set metrics, it allowed us to determine whether the network design was overfitted or not.

TABLE 1 Experiment settings

Hyperparameter of hybrid model	Value
Learning rate	Adjusted automatically 0.00025
Batch size	149
Input drug shape: (image height) x (image width) x (image depth)	48, 48, 4
Word embedding dim	10, 208
Dropout	0.5
Hidden LSTM layers	8
CNN filter size	1×1, 3×3, 5×5
Number of filters	16
Activation function for CNN	RELU
Epoch number	80
Hybrid model loss	Categorical cross entropy
Optimization function	Adam

Abbreviations: CNN, convolutional neural network; LSTM, long short-term memory.

- *Batch size*: represents the number of instances that are used in each step of the training.
- *Embedding dim*: represents the dimensions of the vector space model.
- *Dropout*: A method used to regularize the network. This setting controls the fraction of neurons available during training time.
- *Filter size*: represents the number of words in the sentence matrix to slide over in each filter.
- *Number of filters*: represents the number of filters per each filter size.

We monitored the loss of the validation set until the optimal performing hyperparameters which showed the best average performance was obtained. At the end of the run, we used the model with the best performance on training and validation as our final model.

6 | PERFORMANCE EVALUATION METRICS

Drug names and drug side-effects were fed into BiLSTM for the training of BiLSTM. When outputs of both GCNN and BiLSTM were concatenated, drug names and their side effects were mapped onto visual features of the drug molecular structures as well as textual features generated from the drug structures. To perform quantitative analysis on the data set for text prediction, performance metrics were used. We evaluate the prediction performance using the widely used performance metrics. We used recall (also known as sensitivity [SN] and true-positive rate [TPR]), false-positive rate (FPR), specificity (SP), prediction accuracy (ACC), precision, and F1-measure to evaluate the performance of the model. Their formulations are as follows[40]:

$$\text{Recall} = \text{SN} = \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}, \quad \text{SP} = \frac{\text{TN}}{\text{TN} + \text{FN}},$$

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}},$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

$$\text{F1 measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},$$

respectively, while FP and FN indicate false positive and false negative, respectively. The F1 score is used to evaluate the overall performance because it provides a reasonable combination of both precision and recall.

The performance analysis of the model was also computed by calculating the area under the receiver operating curve (ROC), which is AUC.[41] An ROC curve is a graph which shows the performance of a classification model at all classification thresholds; while AUC measures the entire 2D area underneath the entire ROC curve from $(0, 0)$ to $(1, 1)$. By setting several thresholds for predicting positive samples, a series of TPRs, FPRs, and precisions can be obtained. AUC ranges in value from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0.0. On the other hand, if predictions are 100% correct, then the model has an AUC of 1.0.

In this hybrid model, drug SMILES strings were converted into drug molecular structure using RdKit before feeding into GCNN. Drug features extracted from drug molecular structures were utilized for predicting drug side-effects associated with drug names and drug molecular structures. The proposed drug molecular captioning approach used for captioning drug molecular structure with its drug name and associated side effects was validated by using BLEU metric. BLEU serves as a metric for evaluating a generated sentence to a reference sentence by comparing *n*-grams of a candidate translation of text to *n*-grams of one or more reference translations and counts the number of matches.[42] The more the matches, the better the candidate translation. Although developed for translation, it can be used for language generation, image caption generation, text summarization, speech recognition, and many more. A perfect match results in a score of 1.0, whereas a perfect mismatch results in a score of 0.0. This value indicates how close the prediction is, with values closer to 1 representing excellent prediction.

7 | RESULTS AND DISCUSSION

In this paper, the functions of GCNN and BiLSTM networks are defined and built independently using the functional API of Keras which allows more flexibility in connecting multiple inputs with different modalities. The two GCNN and BiLSTM networks operate independently of each other until they are concatenated.

In the GCNN network, the images are flattened to convert the pooled drug features to a single column (line 1) and then passed to an FC layer (line 2) to deliver an image output (line 3). This FC layer is called the "transfer layer."

1. $x = \text{Flatten}()(x)$,
2. $x = \text{Dense}(100, \text{activation} = \text{'relu'})(x)$,
3. $\text{image\_model} = \text{Model}(\text{inputs} = \text{input\_img}, \text{outputs} = x)$.

The layers of the GCNN are examined (line 4). The last layer normally used for classification is removed and we generate a new model called image_model_transfer (line 5).

1. image_model.layers.pop(),
2. image_model_transfer = Model(inputs = input_img, outputs = image_model.layers[-1].output).

Transfer layer is the layer which is responsible for transferring the image output of GCNN to the newly created image_model_transfer (line 6). Keras back-end is used to map vector dimension of drug molecular image from GCNN model with sequences of integer-tokens that can be converted to text from BiLSTM (line 7, where K is TensorFlow Keras back-end). The transfer layer will output a vector of transfer value which maps the vector dimension of output from BiLSTM for concatenation (line 8).

4. transfer_layer = image_model.get_layer('fully_connected_layer'),
5. transfer_values_size = K.int_shape(transfer_layer.output),
6. transfer_values=\image_model_transfer.predict(),

We create a "transfer-feature dictionary" to map drug molecular structure with the transfer value (line 9).

7. transferfeaturesdict = dict(zip(df['drugname'], transfer_values)).

The Keras functional API enables the two subnetworks with different input modalities of the model to ultimately concatenate. After concatenation, the final step to process the combined input is to define a Keras Model object which accepts two inputs of different modalities and define the output as the final set of FC layers. The FC layers of the model which serves as "decoder" are trained. The fully connected layer for the combined input is activated by ReLU activation function. The network is optimized with Adam optimizer and the last FC layer has Softmax as an activation function to perform multiclass classification. The effectiveness of our hybrid model was evaluated using AUC, F1 score, precision, and recall metrics as well as BLEU scores.

7.1 | Analysis of prediction performance evaluation metrics

Figure 5 and Table 2 indicate that the hybrid model achieved accuracy, recall, precision, F1, and AUC scores of 0.514, 0.518, 0.925, 0.664, and 0.846, respectively. From Figure 5A, we can see that the model achieves a categorical accuracy score of 0.514. As accuracy score is a reliable measurement only when we have symmetric data sets, and the score can be regarded as an effective indicator for prediction of side effects only if the data set used is balanced, in this case, the presence of rare side effects from some drugs has made the data set imbalanced, thus causing the accuracy score to be lower.

Figure 5B shows that the recall score of 0.518 was achieved by our proposed model. The recall score is not high because a small data set was fed to the machine. However, the recall score shows that our hybrid model has the capability to predict drug side-effects even though only 149 SMILES strings were graphed as input to CNN. To improve the accuracy and recall scores, larger data set can be used as larger data set may help reduce the imbalance of data set and consequently improve the model performance. Additional features and different model hyperparameters need to be investigated to further enhance the accuracy and recall scores of the model.

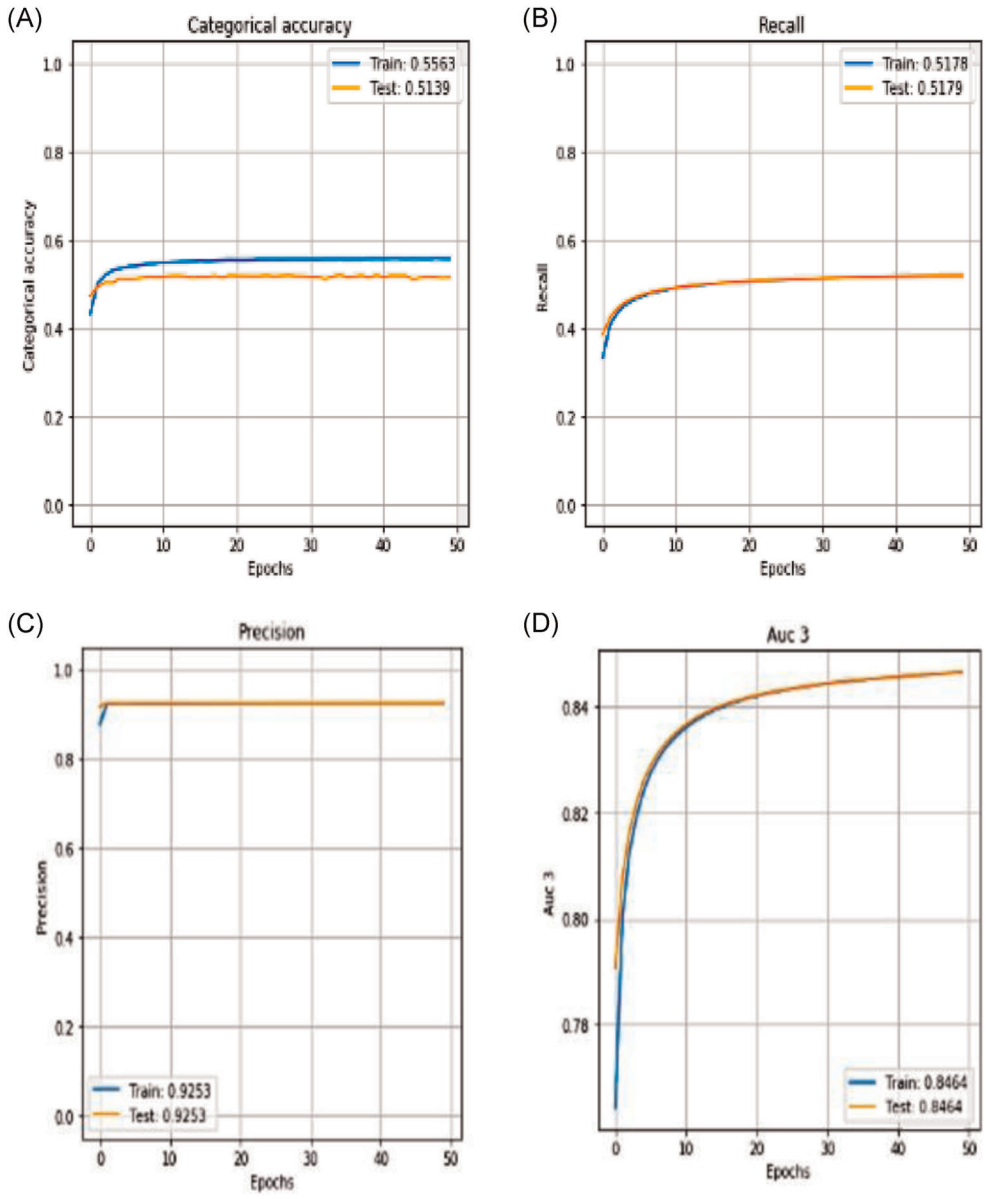

FIGURE 5 Performance of the hybrid GCNN–BiLSTM model on small drug data set. A high AUC score indicates that the proposed model achieves good prediction. AUC, area under the ROC curve; BiLSTM, bidirectional long short-term memory; GCNN, graph convolutional neural network; ROC, receiver operating curve [Color figure can be viewed at wileyonlinelibrary.com]

The high precision score of 0.925 in Figure 5C suggests that our hybrid model can make accurate side-effect predictions. As ADRs are composed of short text fragments, graph CNN with Inception modules can capture the necessary information on drug molecular properties and link them to drug side-effects.

For drug side-effect prediction, it is undesirable to optimize or prioritize any performance evaluation metric over another, hence the F1 score, which is the weighted average of precision and recall, was measured as it takes both false positives and false negatives into account for the

TABLE 2 Experiment results

Evaluation metrics	Value
Accuracy	0.514
Recall	0.518
Precision	0.925
F1 score	0.664

imbalanced side-effects distribution of a given drug data set. The F1 score recorded is 0.664, while AUC score achieved is 0.846, as shown in Figure 5D. The low recall score lowers the F1 score. However, the high AUC score shows that despite using a small drug data set, our hybrid DL model is able to produce accurate ADR prediction results irrespective of what classification threshold is chosen.

7.2 | BLEU score measurement

In addition to the above performance metrics, we also used the metric BLEU to evaluate the predicted side effects of drugs. The BLEU scores obtained were 0.973, 0.938, 0.927, and 0.318,

TABLE 3 Comparison of our hybrid model with other state-of-the-art method

Model	Our hybrid model	DeepTox	1-dim CNN
Type of deep neural network	Hybrid two-dimensional graph CNN–LSTM model which consists of Inception modules	DNN	1-dim CNN
Input feature	Graph drug molecular structure and drug side-effect data	ECFP features	SMILES matrix
Size of data set	Small data set 149, SMILES strings converted into graph molecular structure	Bigger data set: 12,000 chemicals and drugs	Bigger data set: 12,000 chemicals and drugs
AUC score	0.846	0.768	0.813
BLEU scores	BLEU-1: 0.973; BLEU-2: 0.938; BLEU-3: 0.927; BLEU-4: 0.318	N/A	N/A
Output	Predicted drug side-effects and captioning of drug molecular structure with its side effects	Detected the presence of toxiphores in drug substructure using different layers	Detected chemical motif (structure), such as protein-binding sites and structures of unknown functional groups

Abbreviations: AUC, area under the ROC curve; BLEU, Bilingual Evaluation Understudy; CNN, graph convolutional neural network; DNN, deep neural network; ECFP, extended-connectivity fingerprint; LSTM, long short-term memory; ROC, receiver operating curve; SMILES, simplified molecular input line entry system.

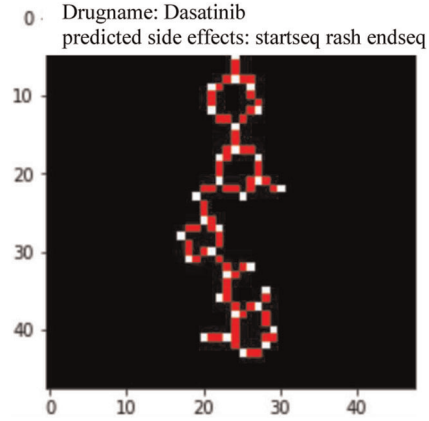

FIGURE 6 The hybrid model generated the drug name "dasatinib," its drug structure and side effects, "rash," when an embedding word dimension of 10 was used [Color figure can be viewed at wileyonlinelibrary.com]

respectively, as shown in Table 3. The high BLEU scores show the ability of the hybrid model to generate descriptive sentences with proper wordings of drug side-effects that are associated with drug structures and their names as shown in Figure 6. Upon physically checking the side effects listed in SIDER, the printed side-effects data show a direct word-to-word side-effect which is identical to the side effects listed in SIDER. The high correlation between BLEU scores and human inspection shows that the GCNN–BiLSTM network can accurately structure correct drug side-effects wordings and relate them to drug names and drug structures.

7.3 | Comparison with other baseline methods

To demonstrate the effectiveness of our proposed hybrid 2D GCNN–BiLSTM model for prediction of drug side-effects, we compare the experimental results of our proposed hybrid model with other state-of-the-art methods, such as the CNN model and DeepTox, which is one of the pioneering methods used for prediction of toxicity using DNNs. Table 3 shows that our hybrid GCNN–BiLSTM model has the highest AUC score (0.846) despite the fact that only a small data set was used for training the model. The highest AUC score confirms that our hybrid model is a superior model for the prediction of drug side-effects. It has the ability to capture the necessary information for ADR classification and prediction. Moreover, the predicted side effects can be printed out in descriptive language.

8 | CONCLUSION

In this paper, we propose a hybrid DL model that consists of GCNN with Inception modules and BiLSTM for the prediction of drug side-effects. It is the first hybrid model with Inception modules run on a small data set of SMILES strings graphed into drug molecular structures. Our model indicates that combination of two types of DNNs can build the capability of DL models to autonomously extract drug molecular properties, and to relate the properties to drug names as well their structures. Our model is one of the first to be able to generate drug side-effects in the correct flow of interpretable descriptive language. Comparisons with other baseline models show that our hybrid model achieves superior AUC prediction score and robustness. In the future, to improve the

accuracy of ADRs prediction, the proposed model can be run on much larger data sets of different sizes to test the scalability of the model. Other types of drug information, such as chemical–protein binding and drug effectiveness, can be incorporated to enhance the robustness of the model. Different configurations of the model can be further investigated to improve the prediction accuracy, which may contribute significantly to the science of pharmacovigilance and hence expedite the drug discovery process.

ORCID

Chun Yen Lee https://orcid.org/0000-0001-8995-1176
Yi-Ping Phoebe Chen https://orcid.org/0000-0002-4122-3767

REFERENCES

1. Lounkine E, Keiser MJ, Whitebread S, et al. Large-scale prediction and testing of drug activity on side-effect targets. *Nature*. 2012;486:361-367.
2. Rothman RB, Baumann MH, Savage JE, et al. Evidence for possible involvement of 5-HT2B receptors in the cardiac valvulopathy associated with fenfluramine and other serotonergic medications. *Circulation*. 2000; 102:2836-2841.
3. Roy M, Dumaine R, Brown AM. A primary human ventricular target of the nonsedating antihistamine terfenadine. *Circulation*. 1996;94(4):817-823.
4. Liu P, Li H, Li S, Leung KS. Improving prediction of phenotypic drug response on cancer cell lines using deep convolutional network. *BMC Bioinf*. 2019;20(408):1-14.
5. Zitnik M, Agrawal M, Leskovec J. Modeling polypharmcy side effects with graph convolutional networks. *Bioinformatics*. 2018;34:457-466.
6. Paul A, Jha D, Al-Bahrani, R, Liao W-k, Choudhary A, Agrawal A. CheMixNet: mixed DNN architectures for predicting chemical properties using multiple molecular representations. In: *Proceedings of the 32nd Conference on Neural Information Processing Systems*. Montreal, Canada; 2018:1-13.
7. Goh, GB, Hodas, N, Siegel, C, Vishnu A. SMILES2vec: Predicting chemical properties from text representations. 6th International Conference on Learning Representations. Vancouver, BC, Canada, 2018:1-4.
8. Segler MHS, Kogej T, Tyrchan C, Waller MP. Generating focused molecule libraries for drug discovery with recurrent neural networks. *Am Chem Soc*. 2018;4(1):120-131.
9. Erasian G, Avsec Ž, Gagneur J, Theis FJ. Deep learning: new computational modelling techniques for genomics. *Nat Rev Genet*. 2019;20:389-403.
10. Jagannatha AN, Hong Y. Bidirectional RNN for medical event detection in electronic health records. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics; 2016:473-482.
11. Luo Y. Recurrent neural networks for classifying relations in clinical notes. *J Biomed Inf*. 2017;72:85-95.
12. Cocos A, Fiks AG, Masino AJ. Deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in Twitter posts. *J Am Med Inf Assoc*. 2017;24(4): 813-821.
13. Huynh T, He Y, Willis A, Rueger S. Adverse drug reaction classification with deep neural networks. In: *Proceedings of the 26th Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee; 2016:877-887.
14. Lee K, Qadir A, Hasan SA, et al. Adverse drug event detection in tweets with semi-supervised convolutional neural networks. In: *Proceedings of the 26th International Conference on World Wide Web*. Perth, Australia: published under Creative Commons CC BY 4.0 License. WWW 2017;2017:705-714.
15. Zhang T, Lin H, Ren Y, et al. Adverse drug reaction detection via a multihop self-attention mechanism. *BMC Bioinf*. 2019;20(479):1-11.

16. El-allaly E, Sarrouti M, En-Nahnahi N, Ouatik El Alaoui S. An adverse drug effect mentions extraction method based on weighted online recurrent extreme learning machine. *Comput Methods Programs Biomed*. 2019;176:33-41.
17. Brandon F, Fan W, Smith C, Garner HS. Adverse drug event detection and extraction from open data: a deep learning approach. *Inf Process Manage*. 2020;57:1-14.
18. Dey S, Luo H, Fokoue A, Hu J, Zhang P. Predicting adverse drug reactions through interpretable deep learning framework. *BMC Bioinf*. 2018;19(476):1-13.
19. Mayr A, Klambauer G et al. DeepTox: toxicity prediction using deep learning. *Front Environ Sci*. 2016;3: 1-13.
20. Wang CS, Lin PJ, Cheng CL, Tai SH, Kao Yang YH, Chiang JH. Detecting potential adverse drug reactions using a deep neural network model. *J Med Internet Res*. 2019;21(2):1-22.
21. Ryu JY, Kim HU, Lee SY. Deep learning improves prediction of drug–drug and drug–food interactions. *Proc Natl Acad Sci*. 2018;2018:1-8.
22. Maya H, Saito Y, Koda Y, Sato K, Sakakibara Y. Convolutional neural network based on SMILES representation of compounds for detecting chemical motif. *BMC Bioinf*. 2018;19(526):83-94.
23. Ozturk H, Ozgur A, Ozkirimli E. DeepDTA: deep drug-target binding affinity prediction. *Bioinformatics*. 2018;34:821-829.
24. Zheng S, Hao Y, Lu D, et al. Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing*. 2017;257:59-66.
25. Zhou D, Miao L, He Y. Position-aware deep multi-task learning for drug–drug interaction extraction. *AIME*. 2018;87:1-8.
26. Anvita G, Müller AT, Huisman BJH, Fuchs JA, Schneider P, Schneider G. Generative recurrent networks for de novo drug design. *Mol Inf*. 2018;37:1-9.
27. Arús-Pous J, Johansson SV, Prykhodko O, et al. Randomized SMILES strings improve the quality of molecular generative models. *J Cheminf*. 2019;11(71):1-13.
28. Sainath TN, Vinyals O, Senior A, Sak H. Convolutional, long short-term memory. Fully connected deep neural networks. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, QLD, Australia: IEEE; 2015:4580-4584.
29. Harshitha K, Ajay B. Ensemble learning on deep neural networks for image caption generation. In: *Proceedings of 14th IEEE International Conference on Semantic Computing, ICSC*. San Diego, CA: Institute of Electrical and Electronics Engineers Inc. 2020:61-68.
30. Li P, Mohamed A, Yuan JH. Real-time crash risk prediction on arterials based on LSTM–CNN. *Accid Anal Prev*. 2020;135:1-9.
31. Zhang Y, Lin H, Yang Z, et al. A hybrid model based on neural networks for biomedical relation extraction. *J Biomed Inf*. 2018;81:83-92.
32. Szegedy C, Liu w, Jia Y, et al. Going deeper with convolutions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA: IEEE; 2015;1-9.
33. Hochreiter S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int J Uncertainty Fuzziness Knowl-Based Syst*. 1998;6(2):107-116.
34. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9(8):1735-1780.
35. Schuster M, Paliwal KK. Bidirectional recurrent neural network. *IEEE Trans Signal Process*. 1997;45(11): 2673-2681.
36. Kuhn M, Letunic I, Jensen LJ, Bork P. The SIDER database of drugs and side effects. *Nucleic Acids Res*. 2016;44:1075-1079.
37. Winter R, Montanari F, Noé F, Clevert DA. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chem Sci*. 2019;10(6):1692-1701.
38. Zheng S, Yan X, Gu Q, et al. QBMG:quasi-biogenic molecule generator with deep recurrent neural network. *J Cheminf*. 2019;11(5):1-12.
39. Kwon SY, Yoon S. DeepCCI: end-to-end deep learning for chemical–chemical interaction prediction. *IEEE/ACM TCBB*. 2017;16(5):1436-1447.
40. Liang H, Chen L, Zhao X, Zhang X. Prediction of drug side effects with a refined negative sample selection strategy. *Comput Math Methods Med*. 2020;2020:1-16.

41. Hanley JA, McNeil BJ. The meaning and use of the area under a receiver operating characteristics (ROC) curve. *Radiology*. 1982;143(1):29-36.
42. Papineni K, Roukos S, Ward T, Zhu W-J. BLEU: a method for automatic evaluation of machine translation. In: Isabelle P, (Ed) *Proceedings for the 40th Annual Meeting on Association for Computational Linguistics*. Stroudsburg, PA, United States: Association for Computational Linguistics; 2002:311-318.

How to cite this article: Lee CY, Chen Y-PP. Descriptive prediction of drug side-effects using a hybrid deep learning model. *Int J Intell Syst*. 2021;36:2491–2510. https://doi.org/10.1002/int.22389