Drug-Target Interaction Prediction in Coronavirus Disease 2019 Case Using Deep Semi-Supervised Learning Model

Faldi Sulistiawan
Department of Computer Science
Faculty of Mathematics and Natural Science, IPB University
Bogor, Indonesia
faldi_sulistiawan@apps.ipb.ac.id

Wisnu Ananta Kusuma
Department of Computer Science and Tropical Biopharmaca
Research Center
IPB University
Bogor, Indonesia
ananta@apps.ipb.ac.id

Nabila Sekar Ramadhanti
Department of Computer Science
Faculty of Mathematics and Natural Science, IPB University
Bogor, Indonesia
nabila.s.ramadhanti@gmail.com

Aryo Tedjo
Department of Medical Chemistry
Faculty of Medicine, Universitas Indonesia
Jakarta, Indonesia
aryo.tedjo@gmail.com

***Abstract*— Coronavirus disease 2019 (COVID-19) is an infectious disease of the respiratory system that caused a pandemic in 2020. There is still not any effective special treatment to cure it. Drug repositioning is used to find an effective drug for curing new diseases by finding new efficacy of registered drug. The new efficacy can be conducted by elaborating the interactions between compounds and proteins (DTI). Deep Semi-Supervised Learning (DSSL) is used to overcome the lack of DTI information. DSSL utilizes unsupervised learning algorithms such as Stacked Auto Encoder (SAE) as pre-training for initializing weights on the Deep Neural Network (DNN). This study uses DSSL with a feature-based chemogenomics approach on the data resulted from the exploration of potential anti-coronavirus treatment. This study finds that the use of fingerprints for compound features and Dipeptide Composition (DC) for protein features gives the best results on accuracy (0.94), recall (0.83), precision (0.817), F-measure (0.822), and AUROC (0.97). From the test data predictions, 1766 and 929 positive interactions are found on the test data and herbal compounds, respectively.**

Keywords— coronavirus disease 2019, drug repositioning, deep semi-supervised learning, stacked autoencoder, deep neural network

I. Introduction

Coronavirus disease 2019 (COVID-19) is an infectious disease located in the respiratory system and is caused by Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) [1]. This disease causes significant health problems such as fever, dry cough, difficulty breathing, pneumonia, multiorgan failure, and even death [2], [3]. Based on the historical data collected by the Worldometer reference website, as of 7 June 2020, there were at least 7008898 total COVID-19 cases that have been reported in 213 countries.

This situations make researchers spend their efforts in drug discovery and development. Researchers conducted drug repositioning for reducing cost. Drug repositioning or drug repurposing is carried out by looking for new efficacy of registered drug compounds. Drug repositioning are typically done by observing the interaction of compounds drugs with proteins which related to the diseases (Drug-Target Interaction or DTI), then predict new DTIs in which the interactions are previously unknown [4], [5]. Conducting drug repositioning can be done on conventional medicines and herbal medicines. Herbal medicines tend to be more easily accessed and used by the Indonesian people in the foreseeable future. Therefore, the development of herbal medicines is necessary [6].

Searching valuable DTI data on COVID-19 can be done by exploring the potential of anti-coronavirus treatment. This data will be utilized for drug repositioning research on COVID-19. This data consists of positive and negative interaction between drug compounds and target proteins. The number of positive interactions is very small when compared to the negative interactions. This imbalance can cause inaccurate classification and prediction models [7]. Therefore, additional handling of imbalanced data is required. Modelling based on deep semi-supervised learning (DSSL) utilizes unsupervised learning algorithms as pre-training on unbalanced data to improve the accuracy of predictive models based on supervised learning [8].

In [9], DSSL method for DTI (DSSL-DTI) prediction is proposed on handling imbalanced interaction data of drug compounds and target proteins. Stacked Autoencoder (SAE) is used as an unsupervised pre-training to initialize the weights of the Deep Neural Network (DNN) classification model. This implementation causes better weight initialization on DNN rather than randomly initializing it. Therefore, the model used was able to achieve better convergence and classification. This study yielded very good results with an accuracy score (AR) of 98.68% and area under the curve score (AUC) of 99.8%. This study uses a feature-based chemogenomics approach on the Yamanishi's golden standard dataset. The chemogenomics approach predicts interactions between compounds and proteins by utilizing information about features of drug compounds and target proteins for predictions [10].

The success of DSSL-DTI method proposed by [9] inspired this paper to implement it in predicting DTI in COVID-19 case. DSSL-DTI method with a feature-based chemogenomics approach will be implemented in the exploration of COVID-19 coronavirus treatment. This paper utilized DSSL-DTI method in predicting interactions between herbal compounds and target proteins in COVID-19 cases.

II. Deep Semi-Supervised Learning for DTI

Research on drug repositioning is based on the fact that most drug compounds can activate or inhibit the biological functions of the target protein. This creates the needs to develop a DTI identification system [11]. DTI identification

978-1-7281-9279-6/20/$31.00 © 2020 IEEE

can be done by building a classification model using DTI data. The data used as input are Drug-Target Pairs (DTP), and the output is a predicted interaction between DTP [9].

A. Deep Semi-Supervised Learning

Deep semi-supervised learning is a deep learning model that adopts semi-supervised learning techniques, which within the training process. This algorithm consists of two stages: unsupervised pre-training and supervised fine-tuning. Greedy layer-wise unsupervised pre-training helps in achieving the minimum cost function value by initializing the initial parameters and acts as a regulation in increasing the power of data generalization [8]. After that, supervised fine-tuning is done to minimize prediction errors within the training process [9].

B. Stacked Autoencoder

The purpose of AE is to learn new representations of data by reconstructing the input data. An encoder that maps inputs into representations at the bottleneck layer and a decoder that maps representations for the reconstruction of the original input. Stacked Autoencoder (SAE) consist of many stacks of AE that can learn efficient coding of unsupervised data [12]. The example of SAE architecture is shown in Fig. 1.

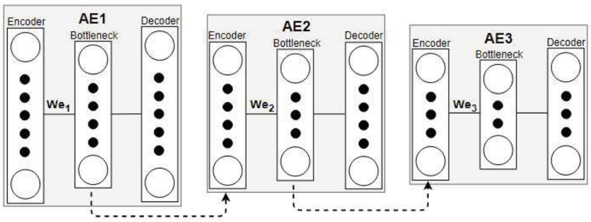

Fig. 1. Stacked Autoencoder

The input data enters the SAE network from the encoder layer on AE1. For example, if the input data are denoted as *x*, the representation *h* can be calculated by (1).

$$h = f_e(x) = s_e(W_e x + b_e) \tag{1}$$

where s_e, W_e, and b_e are the activation function, the weight matrix, and the bias vector of the encoder, respectively. After that, the decoder maps the representation of the bottleneck layer to the output layer (x_r) using (2).

$$x_r = f_d(h) = s_d(W_d h + b_d) \tag{2}$$

In (2), $s_d, W_d, and\ b_d$ are the activation function, a weight matrix, and a bias vector of the decoder, respectively. SAE with N layers and $P = \{P^i \mid i \epsilon \{1,2, \dots N\}\}$, with $P^i = \{W_e^i, W_d^i, b_e^i, b_e^i\}$ can be formulated by (3), (4), and (5).

$$h^i = f_e^i(h^{i-1}) = s_e^i(W_e^i h^{i-1} + b_e^i) \tag{3}$$

$$h_r^i = f_d^i(h_r^{i-1}) = s_d^i(W_d^i h_r^{i+1} + b_d^i) \tag{4}$$

$$h^0 = x \tag{5}$$

where *i* is the parameter in the *i*-th AE. The bottleneck layer at the AE (*i-1*) is used as the input layer at the *i*-th AE [9], [13].

C. Deep Neural Network

. The purpose of DNN is to find the right mathematical manipulation to convert inputs into outputs, and determine whether the relationship is linear or non-linear [14]. The example of DNN architecture is shown in Fig. 2.

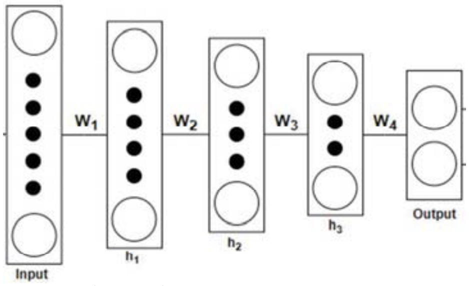

Fig. 2. Deep Neural Network

The inputs move across the network through layers that calculate the output probabilities. Each hidden layer maps the inputs from the previous layer (x_j) to the output (y_j) which will be sent to the subsequent layer as in (6).

$$y_j = f(x_j), \quad x_j = b_j + \Sigma_i y_i w_{ij} \tag{6}$$

The function $f(x_j)$ in (6) is an activation function, b*j* is the bias of unit *j*, *i* is the unit index of the previous layer, and w_{ij} represents the weight that connects *i* and *j*. In the initial construction of the model, the weights and biases in each layer are initialized randomly. DNN can be discriminatively trained with backpropagation that uses cost derivatives (*f '(C)*) to calculate the difference between the target output and actual output. Weights are updated using (7).

$$\Delta w_{ij}(t) = \alpha\, \Delta w_{ij}(t-1) - \epsilon\, f'(C) \tag{7}$$

where Δw_{ij} is the weight update, α is the "momentum" coefficient ($0 < \alpha < 1$) that can smooth the gradient, *t* is the amount of data on the minibatch, and ϵ represent an error. Weights will be updated proportionally using stochastic gradient descent to minimize the error.

III. Materials and Methods

A. Data Acquisition

The data used in this paper came from [15] and [16]. These researches predicted the possibility of approved antiviral drug compounds that will potentially inhibit the development of SARS-CoV-2. In [15], the potential for reusing antiviral agents is investigated based on the therapeutic experience with two infections caused by other coronaviruses. While in [16], proteins encoded by the SARS-CoV-2 gene are systematically analyzed, comparing them with target proteins from other coronaviruses, and predicting their structure using homology modelling. The potential of the antiviral drugs in [15] and [16] are determined by a significant binding affinity score on drug-target interaction. SuperTarget [17] is used to find a list of target proteins from compounds and a list of compounds that interact with target proteins. This is done in order to expand the exploration of drug-target interaction. The output of SuperTarget is a new protein and drug compound target that previously was not mentioned in [15] and [16].

B. Drug-Target Representation

In DTI prediction, numerical representations of proteins and compounds are needed as input data on the classification model. These representations are required so that the machine learning algorithm can understand the DTI data [18]. Compound and protein descriptors will be used to extract the numerical representations.

The compound descriptors are Simplified Molecular-Input Line-Entry System (SMILES). By using SMILES, fingerprint of a chemical structure can be obtained to effectively represent compounds. Fingerprint (FP) is the encoding of a compound into a Boolean FP vector that represents the existence of a substructure within the molecule of the compound. PubChem's FP dictionary will be used because it is considered good in representing the compound molecules [19]. PubChem defines 881 chemical substructures. The retrieval of FP produces feature vector S ($S = [s_1, s_2, s_3, \dots s_{881}]$).

The protein descriptors are protein sequences in FASTA type text files, which are text-based formats to represent nucleotide or amino acids sequences, each of which is represented using a single letter code. Three types of protein features will be extracted by using FASTA, which are Dipeptide Composition (DC), Autocorrelation Descriptors (ACD), and Position-Specific Scoring Matrix (PSSM).

DC is the ratio of dipeptides [18]. Dipeptides are combinations of 2 amino acid components (such as AA, AR, AN, AD, AC) [20]. DC converts protein sequences into 400 features. DC can be defined by (8).

$$X_{dep(i)} = \frac{n_{dep(i)}}{N} \quad (8)$$

where *dep(i)* is the *i*-th dipeptide of 400 dipeptides. $X_{dep(i)}$ represents the ratio of occurrences of *dep(i)*, $n_{dep(i)}$ is the number of occurrences of *dep(i)*, and *N* is the sum of occurrences of all dipeptides.

ACD is defined based on the distribution of amino acid traits along the sequence [21]. The traits of the amino acids were obtained from the AAIndex Database. The amino acid index that were used is the default index on the PROFEAT web [22], namely alpha-CH chemical shifts, hydrophobicity index, and membrane-buried preference parameters. The total features for ACD are 270 features.

PSSM gives the probability value for each amino acid in each position in the protein sequence [23]. PSSM is a matrix of *N* x 20. Each element *M(i, j)* can be obtained using (9).

$$M = \begin{pmatrix} \alpha_{1,1} & \cdots & \alpha_{1,20} \\ \vdots & \ddots & \vdots \\ \alpha_{N,1} & \cdots & \alpha_{N,20} \end{pmatrix} \quad (9)$$

where *N* is the length of the protein sequence, and $\alpha_{(i,j)}$ is the probability that the *i*-th protein residue mutates to the *j*-th amino acid in multiple sequence alignment proteins [23]. Auto Cross Covariance (ACC) transforms PSSM into a scale-based descriptor. Therefore, PSSM information of all proteins taken has the same length to build feature vectors [24]. ACC is a combination of auto covariance (AC) of amino acids and cross-covariance (CC) between two amino acids. AC measures the correlation of amino acids in two separate residues by the *lg* distance along the sequence and is formulated by (10).

$$AC = \sum_{j=1}^{L-lg}\left(S_{ij} - \bar{S}_i\right)\left(S_{i,j+lg} - \bar{S}_i\right)(L - lg)^{-1} \quad (10)$$

In (9), i is the residue, *L* is the length of the protein sequence, S_{ij} is the PSSM value of the amino acid *i* in position *j*, $\bar{S}_i$ is the average of the amino acid values along the sequence. The number of AC features is as much as 20 * *lg*. CC measures the correlation between two amino acids in two residues separated by the *lg* distance along the sequence and is formulated by (11).

$$CC = \sum_{j=1}^{L-lg}\left(S_{xj} - \bar{S}_{i1}\right)\left(S_{yj+lg} - \bar{S}_{i2}\right)(L - lg)^{-1} (11)$$

with *x*, *y* are two different amino acids. The number of CC features is 380 * *lg*. The *lg* variable used is 1. The number of ACC features is 400. The retrieval of protein features will produce 3 feature vector P ($P = [p_1, p_2, p_3, \dots p_n]$) where each of them will contain DC, ACD, and PSSM.

The feature vectors of compound and protein are combined into 3 feature vector F ($F = [s_1, s_2, s_3, \dots s_{881,} p_1, p_2, p_3, \dots p_n]$). Each F is a combination of FP and each of the protein features types. The data is transformed by creating every feature vector pairs from unique protein IDs and compound IDs. Pairs of compounds and proteins that are known to have interactions will be labelled as 1, if not, labelled as 0. Data cleaning is done by filling in missing values with median values and normalization. To increase the chance of randomness and overcome class imbalance, a random sample of negative interactions was selected as many as five times the number of positive interactions. This sample is used for training data [19]. Negative interaction data that were not taken by random samples will be used as test data.

C. SAE-DNN Modelling

Before modelling, all transformed data will be used as input to SAE modelling. SAE will be trained with the aim of initializing weights on DNN. Pre-training uses the concept of unsupervised learning. Initialization of weights in DNN modelling is done to produce an optimal model [9]. The unsupervised pre-training SAE process carried out in [9] consisted of the following stages:

- Step 1: Use the entire dataset as input data in the encoder layer to train the initial AE.
- Step 2: After the AE has been trained, save the weight and bias parameters in the encoder layer.
- Step 3: Remove the decoder layer, take the data representation at the AE bottleneck layer as input data for the encoder layer to train AE2.
- Step 4: Repeat steps 2 and 3 to train next AE. Repeat until all AE is trained.

The next step is the supervised fine-tuning process by making a DNN prediction model. The training data is used as input data. Weights and biases from the results of the SAE training will be used as weight and bias parameters in the

DNN architecture. The weight at the output layer is randomly initialized. Because the data label is binary, the sigmoid function is used as the activation function at the output layer and the binary cross-entropy is used as cost function.

To improve the performance of the model, Batch Normalization (BN) and Dropout (DO) are used. BN is used to normalize the input of each layer by ensuring all inputs have a mean close to 0 and standard deviations close to 1. This is done to speed up the training process [25]. DO removes a random percentage of the output node from each layer when training the model to reduce the capacity and complexity of DNN and prevent overfitting [26]. The use of DO after conducting a BN can lead to a stable training process, faster convergence, and better generalization performance [27].

Hyperparameter tuning is done to choose a combination of hyperparameter (*X*) from all the hyperparameter combinations provided (R_D) on a model that optimizes the performance of the SAE-DNN model. Bayesian Optimization is used because it is able to produce optimal results in shorter iterations compared to greedy method. Bayesian Optimization builds a probabilistic model to make decisions on choosing the next *X* selection while eliminating uncertainty [28]. The list of hyperparameters is shown in Table 1.

TABLE 1. HYPERPARAMETERS FOR TUNING

Params	Values
Hidden node 0 (HN_0)	300, 500, 700, 1 000
Hidden node i (HNi)	0.5 x HN_{i-1}, 0.66 x HN_{i-1}, 0.75 x HN_{i-1}
Hidden layer	2, 3, 4
Optimizer	adam, sgd, adagrad, rmsprop
Learning rate	0.01, 0.001, 0.0001
Activation function	relu, tanh, sigmoid
Dropout rate	0.1, 0.3, 0.5

D. Model Evaluation

Stratified *k*-fold cross-validation is used to test the model. This method divides the data into *k* parts according to the proportion of its class, followed by making a model using *k-1* parts of data and tested using 1 part of data. Training and testing are done k times on different pieces of data. Evaluation is done by paying attention to the prediction label with the actual label. The metrics are defined as follows:

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100 \qquad (12)$$

$$Recall = \frac{TP}{(TP+FN)} \qquad (13)$$

$$Precision = \frac{TP}{(TP+FP)} \qquad (14)$$

$$F - Measure = \frac{(2 \times Precision \times Recall)}{(Precision + Recall)} \qquad (15)$$

E. Predicting New Interactions

The best model based on the selection of protein features and hyperparameter tuning will be used to predict interactions on the test data and herbal compound data. The selection of protein features is done by comparing the performance of the SAE-DNN model on three different data, which are *FP*-DC, *FP*-ACD, and *FP*-PSSM. Test data is the negative interactions that were not taken by random samples in making training data. Herbal compound data comes from Indonesian Herbal Database (HerbalDB). The herbal compounds dataset includes the name of the compound and its PubChem FP. All combinations of herbal and protein pairs are made by combining the FP of the herbal compound with the selected protein features.

IV. RESULTS AND DISCUSSION

A total of 78 virus-based compounds, 16 human-based compounds, 15 host-based proteins, 7 virus-based proteins along with the DTIs are obtained from [15] and [16]. The data gathered from SuperTarget adds 29 new compounds and 303 new proteins. After the data is combined, pre-processing continues with the verification of compound IDs and protein entry names, and eliminating duplicates. The final dataset consists of 39975 interactions, with 712 positive interactions and 39263 negative interactions. FP-DC and FP-PSSM have 1281 columns, while FP-ACD has 1151 columns. After the data is ready to be used, SAE-DNN hyperparameter tuning is done on those three datasets. The results of this process are shown in Table 2.

TABLE 2. HYPERPARAMETERS TUNING RESULTS

Params	Values		
	FP-DC	FP-ACD	FP-PSSM
Hidden node 0 (HN_0)	300	1 000	700
Hidden node i (HNi)	½ * HNin-1	½ * HNin-1	½ * HNin-1
Hidden layer	2	4	3
Optimizer	adam	adam	rmsprop
Learning rate	0.01	0.01	0.01
Activation function	relu	relu	relu
Dropout rate	0.5	0.5	0.5

The test was performed using stratified k-fold cross-validation with k = 10 on the training data from three datasets, namely FP-DC, FP-ACD, and FP-PSSM. The test results of cross-validation are represented by the average value as well as the standard deviation of the metrics used. These results are presented in Table 3.

TABLE 3. TESTING RESULTS

Metrics	Model		
	FP-DC	FP-ACD	FP-PSSM
Accuracy	0.940±0.009	0.884±0.009	0.930±0.009
Recall	0.830±0.043	0.423±0.049	0.761±0.040
Precision	0.817±0.050	0.785±0.054	0.815±0.051
F-measure	0.822±0.022	0.548±0.044	0.785±0.025
AUROC	0.970±0.009	0.855±0.023	0.947±0.013

Table 3 shows that the FP-DC model is superior compared to other models, with an average accuracy of 0.94, recall 0.83, precision 0.817, F-measure 0.822, and AUROC 0.97. This proves that the SAE-DNN model that uses this feature vector as training data is good at predicting DTI classes (accurate), accurate in predicting positive classes (high recall), good positive predictions (high precision), good performance on minority class (high F-measure), and confident in distinguishing classes (high AUROC). The smaller the standard deviation, it means that the metric is more stable in each fold of the CV. The FP-DC model has the smallest standard deviation value, except for the recall metric that is outperformed by the FP-PSSM model.

Another measurement done is by creating the Receiver Operating Characteristic (ROC) which are obtained by making a graph between the True Positive Rate (TPR) and the False Positive Rate (FPR) of several threshold values. Area Under ROC Curve (AUROC) value can be calculated as a numerical representation of ROC curve. To compare the ROC curve, the ROC curve on the fold is used to produce the best AUROC in each model and can be seen in Fig. 3.

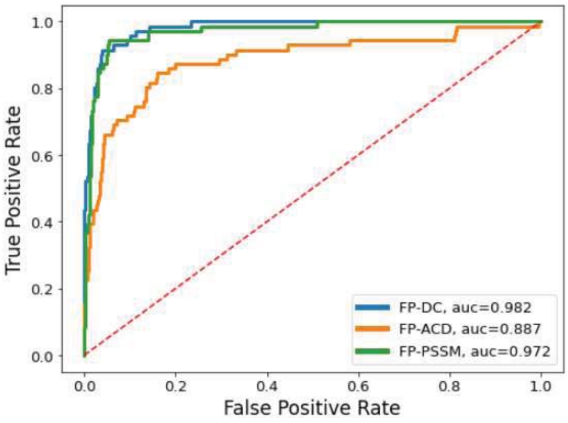

Fig. 3. Receiver operating characteristic curve of the FP-DC model

The ROC curve shows the trade-off between sensitivity (or TPR) and specificity (1 - FPR). Based on the graph in Fig. 3, it can be observed that the FP-DC model has a wider ROC curve area compared to other models. A more expanded area causes a higher AUC. AUC explains that the model has a 98.2% chance of being able to distinguish between positive and negative classes.

Further comparison on the models is done by analyzing confusion matrix results on the fold that produce the best AUROC in each model. Metrics on each model is also shown. This comparison can be seen in Table 4.

TABLE 4. CONFUSION MATRIX AND METRICS ON THE FOLD THAT PRODUCE THE BEST AUROC IN EACH MODEL.

Metrics	Model		
	FP-DC	FP-ACD	FP-PSSM
True Positive (TP)	65	28	53
True Negative (TN)	340	350	349
False Positive (FP)	16	6	18
False Negative (FN)	6	43	7
Accuracy	0.948	0.885	0.941
Recall	0.915	0.394	0.746
Precision	0.802	0.824	0.883
F-measure	0.855	0.533	0.809
AUROC	0.983	0.887	0.972

Based on the test results, the FP-DC model provides better metrics than other models. For this reason, the FP-DC model will be used to predict the test data and the herbal compounds data. Test data has 35703 interactions. Prediction results resulted in 1766 positive interactions. There are 403 herbal compounds used for the herbal compound data. Herbal compound data consists of all pairs of herbal compounds and target proteins. Herbal compounds data have 130975 interactions. The results of the prediction of test data showed that there were 929 positive interactions. Ten interactions of herbal compounds with the COVID-19 target protein in [16] with the highest probability according to the model are presented in Table 4.

TABLE 5. TEN POSITIVE COVID-19 DTIS WERE PREDICTED FROM THE FP-DC MODEL ON THE HERBAL COMPOUNDS DATA WITH THE HIGHEST PROBABILITY

Compound	Protein	Probability
Y-mangostin	RdRp	0.994717
Stigmatellin	RdRp	0.994717
Berberine2	RdRp	0.994717
3-Eicosyne	RdRp	0.994650
Enzacamene	3CLPro	0.994538
Brazilein	3CLPro	0.994439
Brazilin	3CLPro	0.994385
Eugenol	3CLPro	0.994145
1,4-Dimethoxy-6,7,8,9-tetrahydro-5-benzocycloheptenone	RdRp	0.993844
Eugenol	RdRp	0.992982

For the research on drug repositioning in handling COVID-19, positive prediction interactions between compounds and proteins associated with SARS-CoV-2 were collected. The number of predicted positive interactions between drug compounds and virus-based target proteins associated with COVID-19 according to [16] on the test data and herbal compound data is shown in Table 5.

TABLE 6. NUMBER OF POSITIVELY PREDICTED INTERACTIONS BETWEEN DRUG COMPOUNDS AND VIRUS-BASED TARGET PROTEINS ASSOCIATED WITH COVID-19

Protein	Identifier	Test data	Herbal
PLPro	PLPro_SARS-CoV-2	92	113
3CLPro	6LU7:A	115	93
RdRp	yp_009725307.1	118	300
Spike-ACE2	6M0J:A	5	0
	6M0J:E	4	0
	6LZG:A	5	0
	6VSB:A	7	0

V. CONCLUSION AND FUTURE WORK

The use of the fingerprint feature as a compound representation and the Dipeptide Composition (DC) feature as a protein representation yielded good average metric value of cross-validation with an accuracy of 0.94, recall 0.84, precision 0.817, F-measure 0.822, and AUROC 0.97. In the test data, 1766 DTIs are predicted to be positive by the model, with 346 of them are associated with COVID-19. For the herbal compound data, it is predicted that there are 929 interactions of the herbal compounds with the target protein, where 506 of them are associated with COVID-19. As for future work, new information on DTI related to COVID-19 might give better training. We hope to be able to utilize a more advanced architecture and wider search space on hyperparameter tuning for a possible increase in performance.

ACKNOWLEDGEMENT

This research is supported by Ministry of Research, Technology and Higher Education, Indonesia, under Competitive Research Grant from Directorate of Higher Education, Indonesia, 2020, contract no. 4020/IT3.L1/PN/2020.

References

[1] A. E. Gorbalenya *et al.*, “Severe acute respiratory syndrome-related coronavirus: classifying 2019-nCoV and naming it SARS-CoV-2,” *Nat. Microbiol.*, vol. 5, no. 4, pp. 536–544, Apr. 2020, doi: 10.1038/s41564-020-0695-z.

[2] N. Chen *et al.*, “Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in Wuhan, China: a descriptive study,” *Lancet*, vol. 395, no. 10223, pp. 507–513, Feb. 2020, doi: 10.1016/S0140-6736(20)30211-7.

[3] D. S. Hui *et al.*, “The continuing 2019-nCoV epidemic threat of novel coronaviruses to global health — The latest 2019 novel coronavirus outbreak in Wuhan, China,” *International Journal of Infectious Diseases*, vol. 91. Elsevier B.V., pp. 264–266, Feb. 01, 2020, doi: 10.1016/j.ijid.2020.01.009.

[4] T. T. Ashburn and K. B. Thor, “Drug repositioning: Identifying and developing new uses for existing drugs,” *Nature Reviews Drug Discovery*, vol. 3, no. 8. Nature Publishing Group, pp. 673–683, 2004, doi: 10.1038/nrd1468.

[5] N. Novac, “Challenges and opportunities of drug repositioning,” *Trends Pharmacol. Sci.*, vol. 34, no. 5, pp. 267–272, May 2013, doi: 10.1016/j.tips.2013.03.004.

[6] L. Erlina *et al.*, “Virtual Screening on Indonesian Herbal Compounds as COVID-19 Supportive Therapy: Machine Learning and Pharmacophore Modeling Approaches,” *BMC Med. Inform. Decis. Mak.*, 2020, doi: 10.21203/RS.3.RS-29119/V1.

[7] V. S. Spelmen and R. Porkodi, “A Review on Handling Imbalanced Data,” in *Proceedings of the 2018 International Conference on Current Trends towards Converging Technologies, ICCTCT 2018*, Nov. 2018, doi: 10.1109/ICCTCT.2018.8551020.

[8] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, “Why does unsupervised pre-training help deep learning?,” in *Journal of Machine Learning Research*, 2010, vol. 9, pp. 201–208.

[9] M. Bahi and M. Batouche, “Drug-Target Interaction Prediction in Drug Repositioning Based on Deep Semi-Supervised Learning,” in *IFIP Advances in Information and Communication Technology*, 2018, vol. 522, pp. 302–313, doi: 10.1007/978-3-319-89743-1_27.

[10] Y. Yamanishi, “Chemogenomic Approaches to Infer Drug–Target Interaction Networks,” in *Data Mining for Systems Biology 2013*, Humana Press, Totowa, NJ, 2013, pp. 97–113.

[11] G. R. Zimmermann, J. Lehár, and C. T. Keith, “Multi-target therapeutics: when the whole is greater than the sum of the parts,” *Drug Discovery Today*, vol. 12, no. 1–2. Elsevier Current Trends, pp. 34–42, Jan. 01, 2007, doi: 10.1016/j.drudis.2006.11.008.

[12] M. A. Kramer, “Nonlinear principal component analysis using autoassociative neural networks,” *AIChE J.*, vol. 37, no. 2, pp. 233–243, Feb. 1991, doi: 10.1002/aic.690370209.

[13] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” in *Advances in Neural Information Processing Systems*, 2007, pp. 153–160.

[14] J. Schmidhuber, “Deep Learning in neural networks: An overview,” *Neural Networks*, vol. 61. Elsevier Ltd, pp. 85–117, Jan. 01, 2015, doi: 10.1016/j.neunet.2014.09.003.

[15] G. Li and E. De Clercq, “Therapeutic options for the 2019 novel coronavirus (2019-nCoV),” *Nature reviews. Drug discovery*, vol. 19, no. 3. NLM (Medline), pp. 149–150, Mar. 01, 2020, doi: 10.1038/d41573-020-00016-0.

[16] C. Wu *et al.*, “Analysis of therapeutic targets for SARS-CoV-2 and discovery of potential drugs by computational methods,” *Acta Pharm. Sin. B*, vol. 10, no. 5, pp. 766–788, May 2020, doi: 10.1016/j.apsb.2020.02.008.

[17] S. Günther *et al.*, “SuperTarget and Matador: resources for exploring drug-target relationships,” *Nucleic Acids Res.*, vol. 36, no. suppl_1, pp. D919–D922, 2007, doi: 10.1093/nar/gkm862.

[18] S. Redkar, S. Mondal, A. Joseph, and K. S. Hareesha, “A Machine Learning Approach for Drug□target Interaction Prediction using Wrapper Feature Selection and Class Balancing,” *Mol. Inform.*, vol. 39, no. 5, p. 1900062, May 2020, doi: 10.1002/minf.201900062.

[19] Y. Bin Wang, Z. H. You, S. Yang, H. C. Yi, Z. H. Chen, and K. Zheng, “A deep learning-based method for drug-target interaction prediction based on long short-term memory neural network,” *BMC Med. Inform. Decis. Mak.*, vol. 20, no. S2, p. 49, Mar. 2020, doi: 10.1186/s12911-020-1052-0.

[20] M. Bhasin and G. P. S. Raghava, “Classification of nuclear receptors based on amino acid composition and dipeptide composition,” *J. Biol. Chem.*, vol. 279, no. 22, pp. 23262–23266, May 2004, doi: 10.1074/jbc.M401932200.

[21] S. A. K. Ong, H. H. Lin, Y. Z. Chen, Z. R. Li, and Z. Cao, “Efficacy of different protein descriptors in predicting protein functional families,” *BMC Bioinformatics*, vol. 8, no. 1, p. 300, Aug. 2007, doi: 10.1186/1471-2105-8-300.

[22] Z.-R. Li, H. H. Lin, L. Han, L. Jiang, X. Chen, and Y. Z. Chen, “PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence,” *Nucleic Acids Res.*, vol. 34, no. suppl_2, pp. W32–W37, 2006, doi: 10.1093/nar/gkl305.

[23] L. Wang, Z.-H. You, X. Chen, X. Yan, G. Liu, and W. Zhang, “RFDT: A Rotation Forest-based Predictor for Predicting Drug-Target Interactions Using Drug Structure and Protein Sequence Information,” *Curr. Protein Pept. Sci.*, vol. 19, no. 5, pp. 445–454, Dec. 2018, doi: 10.2174/1389203718666161114111656.

[24] X. Liu, L. Zhao, and Q. Dong, “Protein remote homology detection based on auto-cross covariance transformation,” *Comput. Biol. Med.*, vol. 41, no. 8, pp. 640–647, Aug. 2011, doi: 10.1016/j.compbiomed.2011.05.015.

[25] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *32nd International Conference on Machine Learning, ICML 2015*, Feb. 2015, vol. 1, pp. 448–456.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014, doi: 10.5555/2627435.2670313.

[27] G. Chen, P. Chen, Y. Shi, C.-Y. Hsieh, B. Liao, and S. Zhang, “Rethinking the Usage of Batch Normalization and Dropout in the Training of Deep Neural Networks,” May 2019, Accessed: Jun. 07, 2020. [Online]. Available: http://arxiv.org/abs/1905.05928.

[28] J. Snoek, H. Larochelle, and R. P. Adams, “Practical Bayesian Optimization of Machine Learning Algorithms,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 2951–2959.