Hindawi
Journal of Chemistry
Volume 2022, Article ID 4148443, 11 pages
https://doi.org/10.1155/2022/4148443

Research Article

Deep Learning-Based Prediction of Physical Stability considering Class Imbalance for Amorphous Solid Dispersions

Hanbyul Lee,[1] Junghyun Kim ,[1,2] Suyeon Kim,[1] Jimin Yoo,[1] Guang J. Choi,[2,3] and Young-Seob Jeong[4]

[1] *Department of Bigdata Engineering, Soonchunhyang University, Asan-si 31538, Republic of Korea*
[2] *Department of Medical Sciences, Soonchunhyang University, Asan-si 31538, Republic of Korea*
[3] *Department of Pharmaceutical Engineering, Soonchunhyang University, Asan-si 31538, Republic of Korea*
[4] *Department of Computer Engineering, Chungbuk National University, Cheongju 28644, Republic of Korea*

Correspondence should be addressed to Junghyun Kim; kimjh@sch.ac.kr

Received 5 November 2021; Revised 20 February 2022; Accepted 9 March 2022; Published 18 March 2022

Academic Editor: Robert Zalesny

This research is aimed at predicting the physical stability for amorphous solid dispersion by utilizing deep learning methods. We propose a prediction model that effectively learns from a small dataset that is imbalanced in terms of class. In order to overcome the imbalance problem, our model performs a hybrid sampling which combines synthetic minority oversampling technique (SMOTE) algorithm with edited nearest neighbor (ENN) algorithm and reduces the dimensionality of the dataset using principal component analysis (PCA) algorithm during data preprocessing. After the preprocessing, it performs the learning process using a carefully designed neural network of simple but effective structure. Experimental results show that the proposed model has faster training convergence speed and better test performance compared to the existing DNN model. Furthermore, it significantly reduces the computational complexity of both training and test processes.

1. Introduction

Amorphous solid dispersion (ASD) has been widely used in pharmaceutical industry for the enhancement of solubility and bioavailability of poorly water-soluble drugs. ASD relies on the concept of dispersing drug molecules into polymer carriers through constraints to form homogeneous amorphous systems. Since drugs exist in the state of separated molecules in this dispersion system, lattice energy to be overcome during decomposition can be completely avoided, and thus the solubility can be increased. Despite the utility of ASD, physical stability remains a challenging issue for formulation scientists.

Unraveling the chemical properties of substances is a key problem and challenge for drug development. Current pharmaceutical formulation development still strongly relies on the traditional trial-and-error methods of pharmaceutical scientists. This approach is a time-consuming and costly process. Indeed, to ensure both patient safety and drug effectiveness, prospective drugs must undergo a competitive and long procedure. For instance, the physical stability test of amorphous solid dispersion (ASD) needs at least three months to six months by trial-and-error experiments. Moreover, the mechanism of physical stability of ASD is still poorly understood and the theoretical approaches need large amount of physicochemical information of each component and plenty of professional knowledge.

Machine learning has the potential to facilitate data-driven decision making, accelerates processes, and reduces failure rates. For this reason, machine learning has been used in many pharmaceutical applications from drug discovery to drug development. In the drug discovery, one of the early areas in which machine learning is applied is quantitative structure activity relationship (QSAR). QSAR is a strategy based on the idea that when we change a structure of a molecule then also the activity or property of the substance will be modified. Since QSAR research involves complex and nonlinear characteristics, various machine learning tools such as artificial neural network (ANN), support vector machine (SVM), decision tree (DT), random forest (RF), radial basis function neural network (RBFNN), and *k*-nearest neighbors (KNN) are widely used [1, 2]. The process of drug development can be divided into pre-formulation and formulation stage. In the preformulation stage, the physicochemical properties of a drug substance are assessed. Determining the physicochemical properties of a drug substance is very importance because it governs various parameters, such as its solubility, stability, interaction with excipients, and bioavailability [3]. In this area, important progress has been achieved in utilizing the emerging machine learning techniques such as ANN for solubility prediction [4] and transfer learning and multitask learning for pharmacokinetic parameter prediction [5]. In the formulation stage, pure drug substances are formulated into drug products to be administered by patients. Neural networks including the deep neural network (DNN) have gained significant interest in this area. As an example, a DNN architecture was proposed for predicting the disintegrating time of oral disintegrating films and oral disintegrating tablets [6].

For poorly soluble orally administered drugs, the absorption rate is often controlled by the dissolution rate of the drug in the gastrointestinal tract. Various techniques have been used to improve the dissolution rate of sparingly soluble drugs in water. Among them, ASD technique is widely used to obtain the amorphous state of drug and improves the dissolution rate of drugs, hence increasing bioavailability. However, amorphous drug is generally unstable and easily crystallized. Thus, the stability of ASD has become the key issue to hinder the commercialization of this technique. The physical stability test for ASD needs at least three months to six months by trial-and-error experiments, which is a time-consuming and costly process. In addition, the mechanism of physical stability of ASD is still poorly understood [7]. In recent years, several theories about the ASD stability were discussed in [8]. However, these theoretical models need large amount of physicochemical information of each component and plenty of professional knowledge. Moreover, the prediction capability of these models was quite limited with the uncontrolled error due to the mathematic hypothesis.

Recently, to improve efficiency and accuracy of ASD formulation development, an intelligent system for the stability prediction of ASD by machine learning approaches was proposed in [9]. The outcomes suggested that the DNN model has the best performance among their machine learning models. However, for small and imbalanced data, the DNN model may cause overfitting problem and the test performance deterioration due to its inefficient structure.

In this paper, we investigate deep learning methods for stability prediction of ASD and propose a new architecture of prediction model. One of the main contributions is that the proposed model can effectively learn from small dataset with imbalanced input space due to the limited experimental data. Another contribution is that our model is suitable for avoiding overfitting. As a result, our model shows better test performance with fewer training parameters and epochs than the existing DNN model in [9].

2. Methodology

2.1. Dataset. An open dataset [10] is used for fair performance comparison. The data were collected from the "Web of Science" database and regenerated by extracting only samples with the same features. The dataset contains four parts: formulations, process parameters, experimental conditions, and stability results. The experimental conditions include the environmental temperature and relative humidity. The dataset is split into the training set and test set. The training set has 103 data samples with 15 features such as molecular weight, melting point, XLogP3, hydrogen bond donor count, hydrogen bond acceptor count, rotatable bond count, topological polar surface area, heavy atom count, complexity, logS, polymers, drug loading ratio, hot melt extrusion, temperature, and relative humidity. The test set has 20 data samples with the same features above. The stability results are individually given for two labels of 3 months and 6 months. Each label is represented as the numbers "1" for stable and "0" for unstable until the corresponding period.

Unfortunately, the mechanism of physical stability of ASD is not yet well understood, and how the relationship between features affects the stability prediction has not been theoretically established. Therefore, we need an effective model design that enables the prediction by extracting linear as well as non-linear characteristics from the given input data.

2.2. Data Preprocessing. It is worth noting that the stability results were classified into four in [9]. Dual binary classification work was performed through multitask learning to predict the stability after 3 and 6 months. However, it is practically impossible to become unstable after 3 months and stable after 6 months, so considering the corresponding class leads to the performance degradation. In this paper, we reformulate the stability results as three classes: (1) a class that the stability is not maintained until 3 months, (2) a class that the stability is maintained until 3 months, but not until 6 months, and (3) a class that the stability is maintained until 6 months.

In the dataset, the number of samples is not the same for all the classes. The data imbalance problem could be one of the main causes that degrade the performance of classification task. To address this problem, an oversampling technique may be used to generate samples of a class with a relatively insufficient number of samples. Synthetic Minority Oversampling TEchnique (SMOTE) [11] and ADAptive SYNthetic sampling (ADASYN) [12] are used as representative oversampling techniques. Using these techniques, we obtained a dataset containing about 200 samples for each class. In addition, we considered undersampling techniques, Tomek Links (TL) [13], one-sided selection (OSS) [14], and edited nearest neighbor (ENN) [15], and condensed nearest neighbor (CNN) [16], for removing a few samples at class boundaries to further improve classification performance. To obtain both the advantages of oversampling and undersampling, we also considered hybrid sampling techniques, SMOTE+ENN [17] and SMOTE+TL [18]. Finally, we selected a hybrid sampling technique, SMOTE+ENN, that

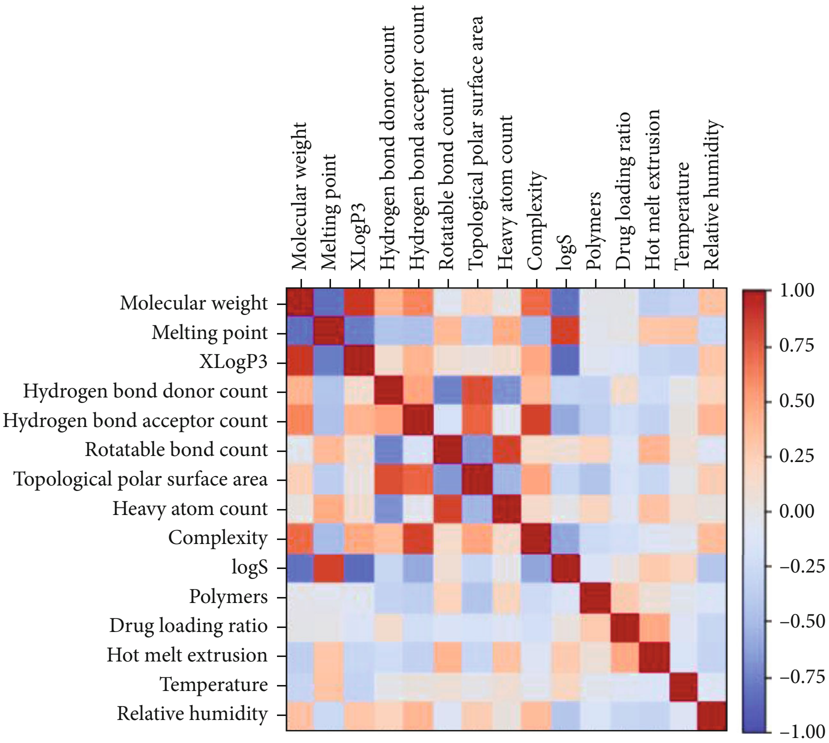

Figure 1: The correlation of the features in the training set.

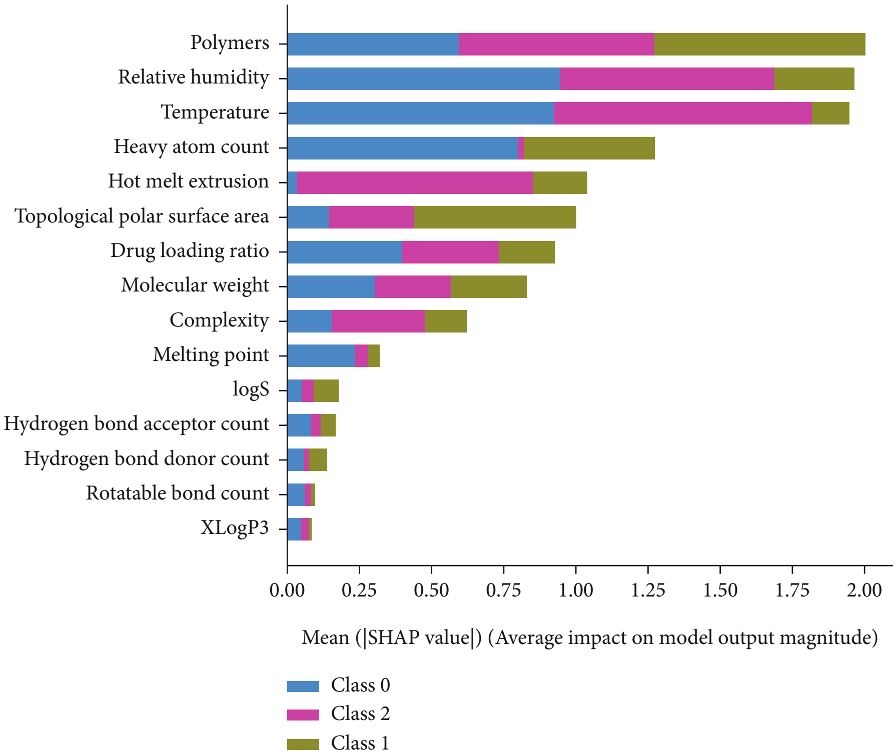

Figure 2: The relative importance of the features obtained by XGBoost.

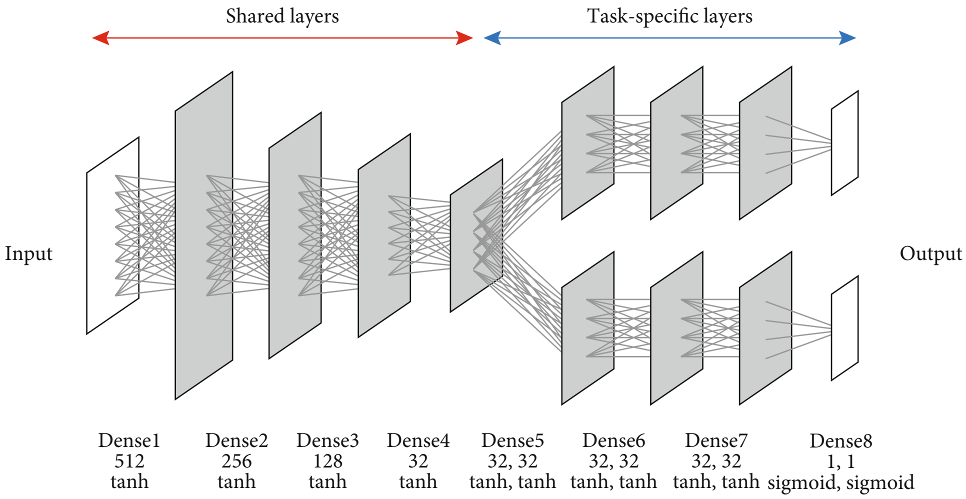

Figure 3: The neural network architecture in [9].

has the best performance. After the sampling, we apply standard scaler to standardize the features so that they are centered around 0 with a standard deviation of 1.

As the next step, we reduce the dimensionality of the features using principal component analysis (PCA) [19]. This algorithm is based on the search of orthogonal directions explaining as much variance of the data as possible. It reduces the dimension of a dataset and increase the speed of the training. In order to determine the amount of the reduction, we calculate the correlation of the features in the training set. It is presented in Figure 1. We also measure the relative importance of the features using eXtreme Gradient Boosting (XGBoost) [20]. XGBoost is an advanced implementation of gradient boosting framework. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. In this process, we can calculate the relative importance of features. The relative importance obtained by XGBoost is presented in Figure 2. From the results, we can see that a feature of low importance is highly correlated with one another. For example, the least important feature, XLogP3, is strongly correlated with molecular weight. Based on these observations, we reduce the dimensionality of the features from 15 to 13 using PCA. As a result, we can not only reduce the computational cost of the classification model but also improve the performance of the model.

2.3. Model Architecture. The neural network architecture used in [9] is shown in Figure 3. A multitask learning technique was adopted to extract the common information in the low-level features among the 3-month and 6-month tasks. This network contained two parts: the first part was the shared layers, and the second part was the task layers. Four shared layers with 512, 256, 128, and 32 neurons were implemented to extract the common features from the raw data. The task layers with two branches were implemented to extract the specific features for the 3- and 6-month stability predictions, respectively. Each subpart of the task layers contained 3 layers of 32 neurons. For the layers, weights are initialized with Glorot uniform distribution [21] and biases are initialized to zero. In the network, hyperbolic tangent function was chosen as the activation function in the hidden layers and sigmoid in the output layers. The loss weights for two tasks were 0.5. The Adam optimizer [22] was adopted with parameters, $\alpha = 0.00001$, $\beta 1 = 0.9$, and β $2 = 0.999$. The batch gradient descent took 1400 epochs.

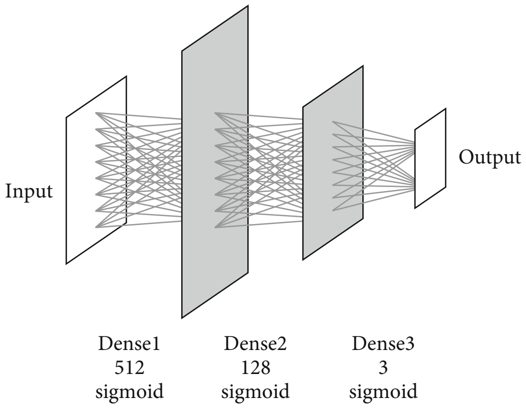

Figure 4: The proposed neural network architecture.

The proposed neural network architecture is shown in Figure 4. The neural network included two hidden layers; the first layer containing 512 neurons and the second layer containing 128 neurons. For the layers, both weights and biases are initialized randomly with standard normal distribution. The two hidden layers used Sigmoid function as the activate function, while the last output layer used Softmax function for the multiclass classification. The Adam optimizer was adopted with parameters, $\alpha = 0.0001$, $\beta 1 = 0.9$, and $\beta 2 = 0.999$. The batch gradient descent took 100 epochs. Note that the number of epochs in the proposed model is only 7.14% of the number of epochs in the model used in [9], and the number of training parameters is also significantly reduced.

3. Results and Discussion

3.1. Model Performance Criterion. The common evaluation metrics for machine learning models are accuracy, precision,

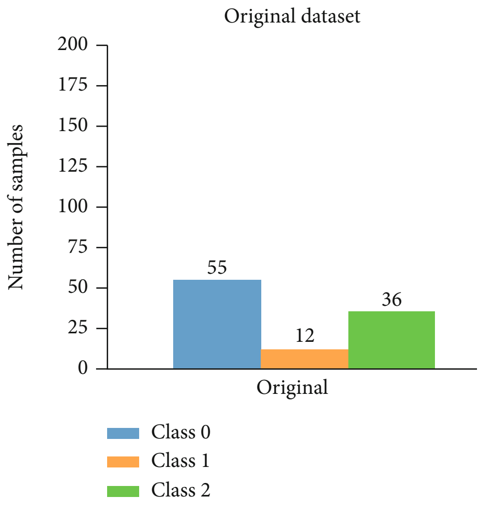

Figure 5: Number of samples per class in the original dataset.

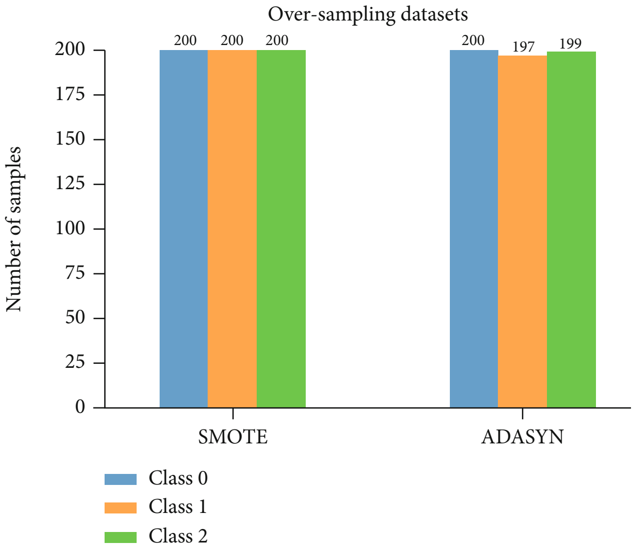

Figure 6: Number of samples per class in the datasets which oversampling SMOTE and ADASYN are applied.

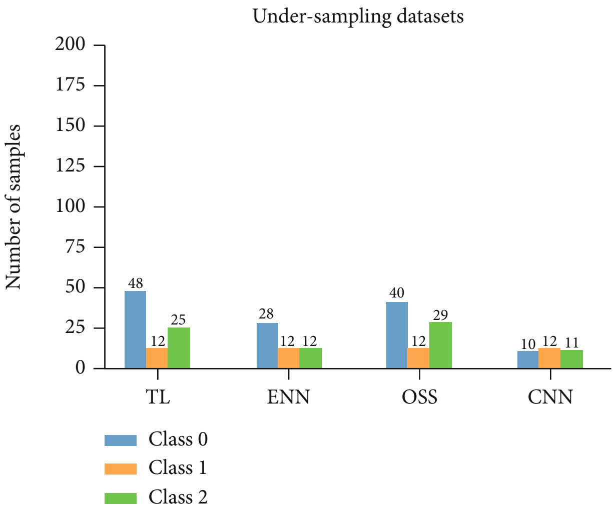

Figure 7: Number of samples per class in the datasets which undersampling TL, ENN, OSS, and CNN are applied.

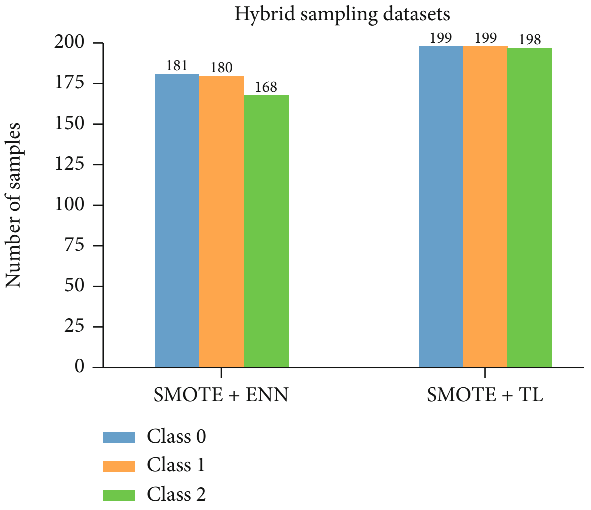

Figure 8: Number of samples per class in the datasets which hybrid sampling SMOTE+ENN and SMOTE+TL are applied.

recall, and $F1$-score. The metrics derive from four types of outcomes for prediction: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). TP means that data samples labelled as positive are actually positive. FP means that data samples labelled as positive are actually negative. TN means that data samples labelled as negative are actually negative. FN means that data samples labelled as negative are actually positive.

Accuracy is a ratio of corrected predictions to total predictions. It is defined as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{1}$$

If a dataset is asymmetric, that is, the numbers of positive and negative samples are different, it is proper to consider other metrics, such as precision, recall, and $F1$-score, as well.

Precision is a ratio of corrected positive predictions to total positive predictions. Therefore, precision measures how many predictions are correct among all samples labelled as positive. The definition of precision is as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{2}$$

Recall is a ratio of corrected positive predictions to total actual samples. Therefore, recall measures how many actual samples the model can label. The definition of recall is as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{3}$$

Table 1: The performance comparison of models applying various sampling techniques to the proposed neural net architecture.

		Without sampling	Oversampling		Undersampling				Hybrid sampling	
			SMOTE	ADASYN	TL	ENN	OSS	CNN	Smote+ENN	Smote+TL
Accuracy	Training set	0.864	0.998	0.99	1.0	1.0	1.0	1.0	1.0	1.0
	5-fold cross-validation	0.864 ± 0.063	0.957 ± 0.027	0.924 ± 0.036	0.871 ± 0.087	0.8 ± 0.041	0.877 ± 0.077	0.8 ± 0.078	0.971 ± 0.019	0.958 ± 0.025
	Test set	0.85	0.9	0.8	0.9	0.8	0.8	0.65	**0.95**	0.9
Precision	Training set	0.865	0.998	0.99	1.0	1.0	1.0	1.0	1.0	1.0
	5-fold cross-validation	0.88 ± 0.071	0.959 ± 0.026	0.933 ± 0.028	0.904 ± 0.062	0.651 ± 0.059	0.908 ± 0.049	0.814 ± 0.17	0.974 ± 0.016	0.96 ± 0.024
	Test set	0.864	0.86	0.864	0.86	0.887	0.826	0.725	**0.96**	0.86
Recall	Training set	0.864	0.998	0.99	1.0	1.0	1.0	1.0	1.0	1.0
	5-fold cross-validation	0.864 ± 0.063	0.957 ± 0.027	0.924 ± 0.036	0.871 ± 0.087	0.8 ± 0.041	0.877 ± 0.077	0.8 ± 0.078	0.971 ± 0.019	0.958 ± 0.025
	Test set	0.85	0.9	0.8	0.9	0.8	0.8	0.65	**0.95**	0.9
*F*1-score	Training set	0.86	0.998	0.99	1.0	1.0	1.0	1.0	1.0	0.998
	5-fold cross-validation	0.857 ± 0.071	0.957 ± 0.027	0.921 ± 0.038	0.869 ± 0.082	0.715 ± 0.054	0.867 ± 0.093	0.765 ± 0.116	0.971 ± 0.019	0.958 ± 0.025
	Test set	0.842	0.878	0.812	0.878	0.825	0.803	0.676	0.952	0.878

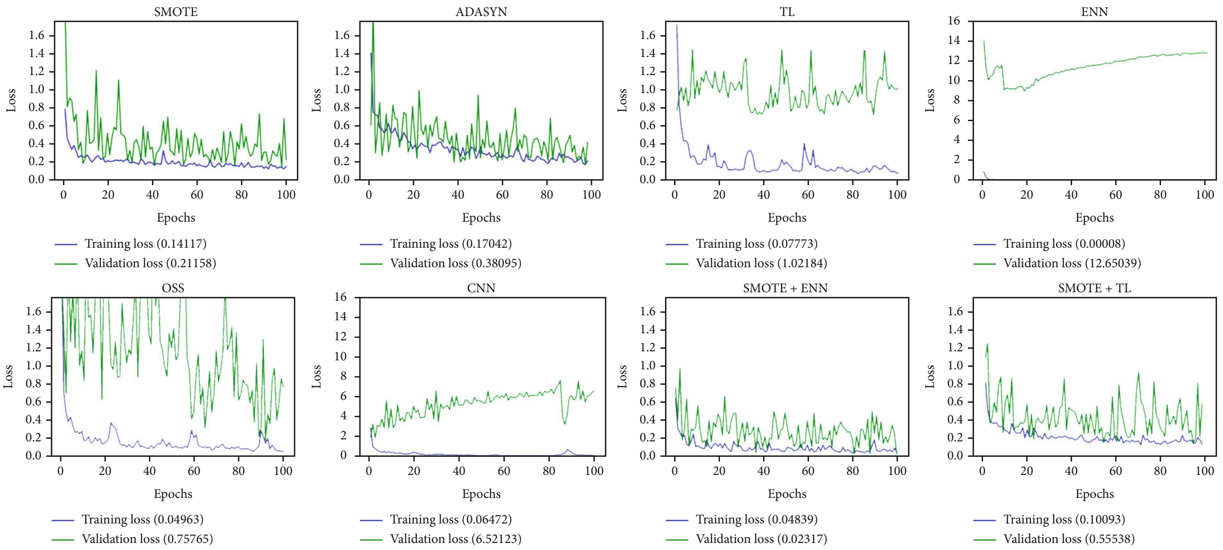

FIGURE 9: Training and validation loss versus epochs for each sampling technique.

$F1$-score is the harmonic mean of precision and recall. It is defined as follows:

$$F1\text{-score} = \frac{2 * (\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}}. \tag{4}$$

In multiclass classification, the performance of a classifier is usually evaluated by microaverage and macroaverage. Macroaverage computes a metric independently for each class and then take the average (hence treating all classes equally), whereas microaverage aggregates the contributions of all classes to compute the average metric. Since our dataset is imbalanced, we only use the microaverage for each of the above metrics. Note that micro- and macroaverage values are the same for a balanced dataset.

3.2. Performance Comparison. In this section, we evaluate and compare the performance of the proposed model and the existing models in [9]. The evaluation metrics are precision, recall, $F1$-score, and accuracy. Figures 5–8 show the original data samples and the number of samples generated by each sampling method used in the experiment. Figure 5 confirms that the data imbalance problem is very serious in the original dataset. In addition, it is confirmed through Figures 6–8 that the imbalance problem is alleviated through the sampling technique.

Table 1 shows that the performance comparison of models applying various sampling techniques to the proposed neural network architecture. At first, we divided the training set and the test set at a 103/20 ratio. After the sampling, we performed 5-fold cross-validation for verification and parameter tuning. In this case, we used a stratified k -fold with random shuffling. Comparing oversampling techniques, SMOTE outperforms ADASYN on both the training set and test set. Because undersampling is trained on a relatively small number of samples than oversampling techniques, it is more prone to overfitting problems. However, among them, ENN is showing relatively good performance. However, among them, TL and ENN are showing relatively good performance. Therefore, the performance of SMOTE +ENN combining oversampling SMOTE and undersampling ENN and SMOTE+TL combining oversampling SMOTE and undersampling TL were compared. As a result, it was confirmed that SMOTE+ENN had the best performance.

Figure 9 shows the change in the loss value as the number of epochs increases in the learning process for each sampling technique. In the figure, the blue curve is the result of the training set and the green curve is the result of the validation dataset. Note that the larger the difference between the two curves, the more overfitting problems arise. From the result, we can expect that SMOTE+ENN ensures the most stable performance by avoiding overfitting.

Given the small size of the dataset, complex deep learning models are prone to overfitting. To verify the performance of the proposed deep learning model, we compared the performance by applying sampling techniques to ensemble machine learning methods, random forest (RF), and light gradient boosting machine (lightGBM) that showed excellent performance in [9]. Tables 2 and 3 show the performance of RF and lightGBM, respectively. We also compared the performance of the machine learning models without sampling as an ablation study. As a result, it was confirmed that there is a limitation in prediction performance due to the data imbalance when any sampling method is not used. Among the sampling methods, hybrid samplings show the best performance as in the proposed model. However, they still show worse performance than that of the proposed model.

Table 4 shows the performance comparison of the final proposed model and the existing DNN model [9]. From the results summarized in Table 4, it is confirmed that the proposed model significantly outperforms the existing DNN model [9] for all the four evaluation metrics introduced above. In particular, the proposed model obtained 100% accuracy in the training set and 95% accuracy even in the test set.

Table 2: The performance comparison of models applying various sampling techniques to the RF [9].

		Without sampling	Oversampling		Undersampling				Hybrid sampling	
			SMOTE	ADASYN	TL	ENN	OSS	CNN	Smote+ENN	Smote+TL
	Training set	1.0	0.985	0.973	1.0	1.0	1.0	1.0	1.0	0.987
Accuracy	5-fold cross-validation	0.691 ± 0.095	0.947 ± 0.015	0.931 ± 0.023	0.8 ± 0.089	0.864 ± 0.092	0.801 ± 0.15	0.543 ± 0.108	0.985 ± 0.013	0.953 ± 0.019
	Test set	0.85	0.9	0.85	0.8	0.65	0.6	0.5	0.9	0.9
	Training set	1.0	0.985	0.975	1.0	1.0	1.0	1.0	1.0	0.987
Precision	5-fold cross-validation	0.724 ± 0.083	0.948 ± 0.014	0.978 ± 0.02	0.846 ± 0.078	0.917 ± 0.022	0.83 ± 0.141	0.459 ± 0.139	0.985 ± 0.013	0.956 ± 0.019
	Test set	0.83	0.933	0.83	0.77	0.842	0.739	0.647	0.933	0.933
	Training set	1.0	0.985	0.973	1.0	1.0	1.0	1.0	1.0	0.987
Recall	5-fold cross-validation	0.691 ± 0.095	0.947 ± 0.015	0.931 ± 0.023	0.8 ± 0.089	0.864 ± 0.092	0.801 ± 0.15	0.543 ± 0.108	0.985 ± 0.013	0.953 ± 0.019
	Test set	0.85	0.9	0.85	0.8	0.65	0.6	0.5	0.9	0.9
	Training set	1.0	0.985	0.973	1.0	1.0	1.0	1.0	1.0	0.987
F1-score	5-fold cross-validation	0.691 ± 0.09	0.947 ± 0.015	0.932 ± 0.023	0.797 ± 0.0939	0.866 ± 0.078	0.789 ± 0.157	0.476 ± 0.115	0.985 ± 0.013	0.953 ± 0.019
	Test set	0.832	0.906	0.832	0.783	0.688	0.607	0.559	0.906	0.906

Table 3: The performance comparison of models applying various sampling techniques to the lightGBM [9].

		Without sampling	Oversampling		Undersampling				Hybrid sampling	
			SMOTE	ADASYN	TL	ENN	OSS	CNN	Smote+ENN	Smote+TL
Accuracy	Training set	0.99	1.0	1.0	1.0	1.0	1.0	0.364	1.0	1.0
	5-fold cross-validation	0.815 ± 0.024	0.953 ± 0.013	0.951 ± 0.028	0.777 ± 0.077	0.522 ± 0.121	0.802 ± 0.053	0.362 ± 0.064	0.98 ± 0.01	0.953 ± 0.023
	Test set	0.9	0.9	0.85	0.75	0.6	0.8	0.05	0.85	0.85
Precision	Training set	0.991	1.0	1.0	1.0	1.0	1.0	0.132	1.0	1.0
	5-fold cross-validation	0.843 ± 0.042	0.956 ± 0.01	0.953 ± 0.028	0.815 ± 0.079	0.398 ± 0.117	0.823 ± 0.045	0.134 ± 0.047	0.982 ± 0.009	0.954 ± 0.022
	Test set	0.86	0.86	0.83	0.709	0.734	0.807	0.002	0.866	0.83
Recall	Training set	0.99	1.0	1.0	1.0	1.0	1.0	0.364	1.0	1.0
	5-fold cross-validation	0.815 ± 0.024	0.953 ± 0.013	0.951 ± 0.028	0.777 ± 0.077	0.522 ± 0.121	0.802 ± 0.053	0.362 ± 0.064	0.981 ± 0.01	0.953 ± 0.023
	Test set	0.9	0.9	0.85	0.75	0.6	0.8	0.05	0.85	0.85
F1-score	Training set	0.99	1.0	1.0	1.0	1.0	1.0	0.194	1.0	1.0
	5-fold cross-validation	0.812 ± 0.032	0.953 ± 0.012	0.951 ± 0.028	0.772 ± 0.07	0.438 ± 0.111	0.798 ± 0.054	0.195 ± 0.059	0.981 ± 0.01	0.952 ± 0.023
	Test set	0.878	0.878	0.832	0.729	0.653	0.788	0.005	0.856	0.832

Table 4: The performance comparison of the existing DNN model [9] and the proposed model.

		The existing DNN model [9]	The proposed model (SMOTE+ENN)
Accuracy	Training set	0.99	1.0
	5-fold cross-validation	0.894 ± 0.07	0.971 ± 0.019
	Test set	0.85	0.95
Precision	Training set	0.991	1.0
	5-fold cross-validation	0.906 ± 0.0813	0.974 ± 0.016
	Test set	0.856	0.96
Recall	Training set	0.99	1.0
	5-fold cross-validation	0.894 ± 0.07	0.971 ± 0.019
	Test set	0.85	0.95
F1-score	Training set	0.99	1.0
	5-fold cross-validation	0.888 ± 0.076	0.971 ± 0.019
	Test set	0.85	0.952

4. Conclusions

In this paper, we proposed a deep learning-based stability prediction model that can replace the conventional ASD stability test that takes a long time and requires expensive costs. The proposed model guarantees superior performance and low complexity compared to the existing technique. For efficient model design, unnecessary class was removed, and the data imbalance problem was solved through the sampling technique. Moreover, remarkable performance was finally obtained through reducing the dimensionality of features through the correlation analysis and PCA technique and designing effective neural network architecture. The proposed technique is expected to be utilized in predicting the properties of various substances for the development of new drugs in the future.

Data Availability

The data used to support the findings of this study are cited at relevant places within the text as references.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-01108, Big data-based development of novel solid forms for P-CAB drugs and establishment of dedicated AI platforms) and the Soonchunhyang University Research Fund.

References

[1] R. Sheikhpour, M. Sarram, M. Rezaeian, and E. Sheikhpour, "QSAR modelling using combined simple competitive learning networks and RBF neural networks," *SAR and QSAR in Environmental Research*, vol. 29, no. 4, pp. 257–276, 2018.

[2] S. Yousefinejad, M. Mahboubifar, and R. Eskandari, "Quantitative structure–activity relationship to predict the antimalarial activity in a set of new imidazolopiperazines based on artificial neural networks," *Malaria Journal*, vol. 18, no. 1, pp. 1–17, 2019.

[3] S. Gaisford and M. Saunders, *Essentials of Pharmaceutical Preformulation*, John Wiley & Sons, Ltd, 2012.

[4] G. Chen, X. Luo, H. Zhang et al., "Artificial neural network models for the prediction of CO_2 solubility in aqueous amine solutions," *International Journal of Greenhouse Gas Control*, vol. 39, pp. 174–184, 2015.

[5] Z. Ye, Y. Yang, X. Li, D. Cao, and D. Ouyang, "An integrated transfer learning and multitask learning approach for pharmacokinetic parameter prediction," *Molecular Pharmaceutics*, vol. 16, no. 2, pp. 533–541, 2019.

[6] Y. Yang, Z. Ye, Y. Su, Q. Zhao, X. Li, and D. Ouyang, "Deep learning for in vitro prediction of pharmaceutical formulations," *Acta Pharmaceotica Sinica B*, vol. 9, no. 1, pp. 177–185, 2019.

[7] F. Qian, J. Huang, and M. A. Hussain, "Drug-polymer solubility and miscibility: stability consideration and practical challenges in amorphous solid dispersion development," *Journal of Pharmaceutical Sciences*, vol. 997, pp. 2941–2947, 2010.

[8] J. Lu, K. Cuellar, N. I. Hammer et al., "Solid-state characterization of Felodipine-Soluplus amorphous solid dispersions," *Drug Development and Industrial Pharmacy*, vol. 42, no. 3, pp. 485–496, 2016.

[9] R. Han, H. Xiong, Z. Ye et al., "Predicting physical stability of solid dispersions by machine learning techniques," *Journal of Controlled Release*, vol. 311-312, pp. 16–25, 2019.

[10] "An open dataset for solid dispersion," https://github.com/yylonly/DeepPharm-InVitro.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[12] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: adaptive synthetic sampling approach for imbalanced learning," in *In Proceeding. IEEE International Joint Conference on Neural Networks*, pp. 1322–1328, Hong Kong, 2008.

[13] I. Tomek, "Two modifications of CNN," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 6, no. 11, pp. 769–772, 1976.

[14] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, pp. 408–421, 1972.

[15] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," in *In Proceeding International Conference on Machine Learning*, pp. 179–186, Nashville, USA, 1997.

[16] P. Hart, "The condensed nearest neighbor rule (Corresp)," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 515-516, 1968.

[17] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine

learning training data," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[18] G. E. A. P. A. Batista, A. L. C. Bazzan, and M. C. Monard, "Balancing training data for automated annotation of keywords: a case study," *WOB*, pp. 10–18, 2003.

[19] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[20] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," in *In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, New York, NY, USA, 2016.

[21] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *In Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, Sardinia, Italy, 2010.

[22] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *In Proceeding. 3rd International Conference on Learning Representations*, pp. 1–15, San Diego, 2015.