ORİJİNAL ARAŞTIRMA ORIGINAL RESEARCH

DOI: 10.5336/biostatic.2019-64948

Small Drug Molecule Classification Using Deep Neural Networks

Küçük İlaç Moleküllerinin Derin Sinir Ağları Kullanılarak Sınıflandırılması

Selçuk KORKMAZ[a]

[a]Department of Biostatistics and Medical Informatics, Trakya University Faculty of Medicine, Edirne, TURKEY

Received: 19.01.2019
Received in revised form: 19.03.2019
Accepted: 20.03.2019
Available Online: 27.03.2019

Correspondence:
Selçuk KORKMAZ
Trakya University
Faculty of Medicine,
Department of Biostatistics and Medical Informatics, Edirne, TURKEY/TÜRKİYE
selcukorkmaz@gmail.com

ABSTRACT Objective: Early phase of drug discovery studies include a virtual screening phase of detecting active molecules among a large number of small drug molecules. The number of publicly available datasets for drug molecules are growing exponentially every year thanks to the databases, such as PubChem and ChEMBL. Therefore, there is a strong need for analyzing and retrieving useful information from these datasets using automated processes. For this purpose, machine learning algorithms are often used for activity prediction of small drug compounds, since they are faster and comparatively cheaper. Deep neural networks has emerged as a powerful machine learning method with great advantages to deal with high-dimensional big datasets. **Material and Methods:** In this study, we applied different settings of deep neural networks models to reveal the effects of learning rate, batch size and minority class weight on performance of the network. **Results:** Small learning rate and large batch size are found to be the most important factors that improve performance of the deep neural network. The best performed model yielded 89% accuracy and 0.78 area under the curve value. **Conclusion:** Findings of this study is promising for use of deep neural networks in virtual screening of small drug compounds from publicly available databases.

Keywords: Drug molecule; deep learning; classification; databases; virtual screening

ÖZET Amaç: İlaç keşfi çalışmalarının ilk aşamasında çok sayıdaki ilaç molekülü arasından aktif moleküllerin tespit edilmesi için sanal tarama çalışmaları yürütülür. İlaç moleküllerini içeren veri setlerinin sayısı, PubChem ve ChEMBL gibi veritabanları sayesinde, her yıl katlanarak artmaktadır. Bu nedenle, otomatize edilmiş süreçlerle bu verilerin analiz edilerek yararlı bilgilerin elde edilmesine ihtiyaç duyulmaktadır. Bu amaçla, makine öğrenmesi algoritmaları hem daha hızlı hem de daha ucuz oldukları için ilaç bileşiklerinin aktivitilerinin kestiriminde sıklıkla kullanılırlar. Derin sinir ağları, yüksek boyutlu büyük verierle baş edebilen ve çeşitli avantajlara sahip güçlü bir makine öğrenmesi yöntemi olarak ortaya çıkmıştır. **Gereç ve Yöntemler:** Bu çalışmada, öğrenme hızı, küme büyüklüğü ve azınlık sınıf ağırlığının ağın performansı üzerindeki etkilerini ortaya koymak için farklı sinir ağ modelleri uygulandı. **Bulgular:** Küçük öğrenme hızı ve büyük küme büyüklüğü, derin sinir ağının performansını artıran en önemli faktörler olarak bulundu. En iyi performans gösteren model %89 doğruluk oranı ve 0,78 eğri altında kalan alan değeri vermiştir. **Sonuç:** Bu çalışmanın bulguları, ücretsiz veri tabanlarından elde edilen küçük ilaç bileşiklerinin sanal taramasında derin sinir ağlarının kullanımının umut verici olduğunu göstermektedir.

Anahtar Kelimeler: İlaç molekülü; derin öğrenme; sınıflandırma; veri tabanları; sanal tarama

Drug discovery is often considered as difficult and time consuming process, since it can take almost 15 years and cost more than one billion dollar. Discovery of a novel drug process can be simply characterized in four parts: target identification, lead optimization, pre-clinical studies and clinical studies.[1] In the first phase of drug discovery and development process, i.e. target identification, thousands of small drug molecules (e.g. compounds) are screened in order to detect active molecules, which can be drug candidates in the future.

An experimental method, called high-throughput screening (HTS) is often used to test large number of small compounds for their activity as inhibitors or activators of a specific biological target (e.g, receptor, enzyme).[2] Alternatively, modern machine learning techniques, which can deal with high dimensional data, can be used as a virtual screening (VS) method to classify compounds as active and inactive. VS methods are comparatively cheaper and faster than the traditional HTS methods. Starting from the late 1990s, many machine learning methods are used to classify drug molecules in the early phase of drug discovery and development studies. One of the early studies conducted by Sadowski and Kubinyi (1998), in which they used neural networks (NN) to discriminate between drug-like and non-drug-like compounds.[3] Byvatov et al. (2003) and Zernov et al. (2003) used support vector machines (SVM) and NN for classification of drug molecules. Korkmaz et al. (2014) used SVM with different feature selection methods in order to improve classification performance.[4-6] Korkmaz et al. (2015) compared performances of twenty-three different machine learning methods and they have created a web-based tool to classify drug compounds as active or inactive using ten best performing algorithms.[1] Other machine learning methods, including naive Bayes (NB), k-nearest neighbor (kNN), Bayesian neural networks and Random Forest (RF) are also used to distinguish between active and inactive compounds.[7-12]

Nowadays, we are in the era of the big data. Digitally stored data size is growing exponentially.[13] One of the serious limitations of the classical machine learning methods is that they cannot handle such big data size. Fortunately, in conjunction with the advance in the computer technology, deep neural networks (DNN), which are now state-of-the-art methods in machine learning, can handle such big data size. The DNN achieve great performances in various fields, such as language translation, speech recognition, text classification, natural language processing and among many others.[14-17] Recently, new studies have been emerged showing that the DNN achieved remarkable performances in drug discovery studies. Ma et al. (2015) used a DNN model for prediction of quantitative structure-activity relationships and they found that the DNN outperformed RF model.[18] Mayr et al. (2016) used a multi-task deep learning architecture to predict toxicity of compounds and won the Tox21 challenge.[19] Ramsundar et al. (2015) also applied a multi-task deep neural networks on various publicly available molecular compound datasets (PCBA, MUV, DUD-E, Tox21).[20-24] Koutsoukas et al (2017) investigated optimization of hyper-parameters of a DNN model and compared the performance of the DNN model with SVM, RF, NB and kNN algorithms.[25] Lenselink et al (2017) compared performance of a DNN model with NB, RF, SVM and logistic regression using ChEMBL bioactivity dataset.[26,27]

There has been a huge increase in the amount of publicly available compound activity and biochemical data.[22,28] PubChem (https://pubchem.ncbi.nlm.nih.gov/), hosted by National Center for Biotechnology Information (NCBI), is a public repository for small drug compounds and their bioactivities.[29] As of September 2015, it contains 60 million unique compounds and 1 million biological assay descriptions.[30] The size of the repository is growing exponentially every year.

In this study, we aimed to investigate performance of DNN for classification of active and inactive drug compounds that are retrieved from the PubChem database. Moreover, we examined how batch size, learning rate and minority class weight affect the performance of the DNN model.

MATERIAL AND METHODS

DATASET

The dataset derived from a quantitative high-throughput screening (QHTS) assay for inhibitors of aldehyde dehydrogenase 1 (ALDH1A1) and it is recorded under AID1030 on PubChem database.[31] The dataset contains 216,890 small drug molecules. There are three labels as activity outcome: active, inactive and inconclusive. We considered inconclusive compounds as inactive, since they do not show any activity indication. Because of the number of active molecules (15,965) are considerably less than the number of inactive molecules (200,925), the dataset is highly imbalanced as expected in a drug compound dataset. After obtaining our dataset, we needed to calculate molecular descriptors. For this task, we used PADEL software to generate 1D-2D-3D descriptors and PubChem fingerprints. More detailed information regarding molecular descriptors can be found in Yap et al. (2010).[32] We generated 1,361 molecular descriptors for each compound using PADEL software.

DEEP NEURAL NETWORKS

Deep neural networks (DNN), also known as deep learning, is a subfield of machine learning which uses successive layers of increasingly meaningful representation of data.[33] A DNN includes multiple non-linear hidden layers between an input layer and an output layer.[34] Since the DNN includes large number of layers and parameters, this makes it so expressive that can learn very complex relationships between input and output.[35,36] The layers, which stacked one after the other, are what we called neural networks and input data are stored in these layers as weights. The DNN model tries to find a set of values for the weights of all layers in the network. Finally, the DNN model will correctly map input data to associated true classes using these weights.[33]

A loss function is used to find the optimal values for the weights. The loss function compares predictions of the DNN model and the true classes, and computes a distance score. This score shows how well the DNN model performed. In order to optimize the loss score, the score produced by the loss function is used as a feedback. This procedure is handled by the optimizer, which implements the backpropagation method.[37] At first, the value of the weights are assigned randomly, which makes the loss score very high. However, the weights are adjusted slightly in the right direction to decrease the loss score. This process is repeated iteratively for a number of times (i.e. number of epoch) to obtain the weights that minimize the loss function. A DNN with a minimum loss score yields a trained network which produces predictions that are as close as they can be to the true classes.[33] A general architecture and workflow of a DNN model illustrated in Figure 1.

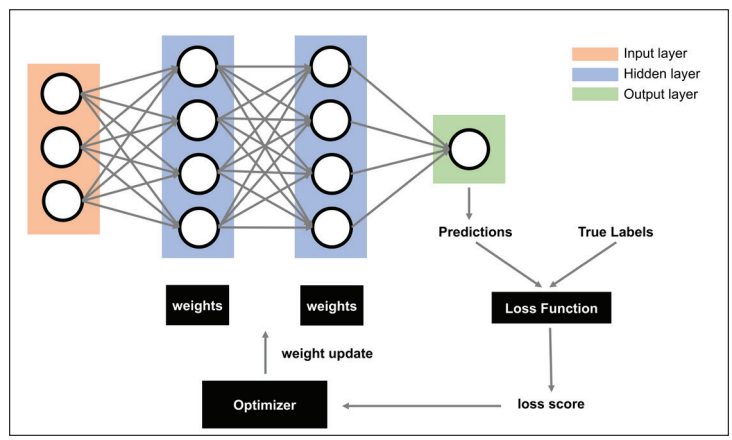

FIGURE 1: A general architecture and workflow of a deep neural networks model.

MODEL BUILDING

First, we split dataset into two parts as 90% training and 10% test set. Then, we centered and scaled training set using z-score transformation. Next, the test set is centered and scaled based on the parameters of the training set (i.e. mean and standard deviation). We used a fully connected four-hidden-layer DNN architecture. The ReLU (rectified linear unit) function is used as activation function for input and hidden layers.[38] On the other hand, a sigmoid activation function is used for the output layer. A batch normalization is applied in each layer to improve the performance and the stability of the DNN. We applied 30% dropout rate at each layer in order to avoid the overfitting as suggested by Srivastava et al. (2014).[36] We used binary loss function and Adam method for stochastic optimization. The number of epoch is defined as 100. A validation set defined as 20% to test the performance of network at each epoch. To investigate the effects of different batch sizes, learning rates and minority class weights, we defined a set of values for these parameters. We selected batch size as 16, 32, 64, 128, 256 and 512. For learning rate, we chose values as 0.001, 0.0001 and 0.00001. Since compound activity dataset is highly imbalanced (i.e. more inactives than actives), we defined a set of weights for the minority class (active compounds). We set the weights for the active compounds as 4, 6, 8, 10 and 12. These weights are used for weighting the loss function during the training. Finally, we trained 90 different DNN models and compared their performances using our test set.

PERFORMANCE ASSESSMENT

In order to measure the performance of our trained DNN models, we calculated various performance measures.

Accuracy is the proportion of the correctly predicted classes among the total number of instances and it ranges between 0 and 1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision, also known as positive predictive value, is the proportion of true positives among instances classified as positive and it ranges between 0 and 1.

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

Recall, also known as sensitivity, is the proportion of true positives among all positives and it ranges between 0 and 1.

$$Recall = \frac{TP}{TP + FN}$$

F1 score is the weighted average value of the precision and recall. It takes both false negatives and false positives into account and it ranges between 0 and 1. This measure is more useful than the accuracy when the dataset is imbalanced.

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

Matthews correlation coefficient (MCC) is another useful performance measure when the data set is imbalanced. The MCC is a correlation coefficient between observed and predicted classes. It ranges between -1

and +1. When the MCC equals -1, it shows a total disagreement between observed and predicted classes, a value of 0 shows no better than random prediction, and a coefficient of +1 shows a perfect prediction.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Area under curve (AUC) is one of the most widely used measures which gives the overall performance of a classification model. It ranges between 0 and 1. A model with AUC equals 0 means it has worst measure of separability, whereas AUC equals 1 means there is perfect separation, and when AUC equals 0.5 it means there is no class separation.

$$AUC = \frac{1}{n_0 n_1} \sum_{i=1}^{n_0} \sum_{j=1}^{n_1} \left(I[Y_{1i} > Y_{0j}] + \frac{1}{2} I[Y_{1i} = Y_{0j}] \right)$$

where, TP = true positives, TN = true negatives, FP = false positives and FN = false negatives, i runs over all n_0 data points with true label 1, j runs over all n_1 data points with true label 0, Y_{1i} is the i^{th} value for true label 1, Y_{0j} is the j^{th} value for true label 0 and I is the indicator function.

RESULTS

We investigated the effects of batch size, minority class rate and learning rate on the performance of the DNN. Since larger sizes did not improve the performance results, we defied the epoch size as 100 for 90 models.

There was no significant difference in accuracy as the batch size increases (Figure 2A). Although there was a slight increase in precision and recall for smaller batch sizes (i.e. 16 and 32), this increase has stopped after the batch size 32, which means larger batch sizes did not improve these measures (Figures 2B-C). Moreover, there were slight but significant improvements in F1, AUC and MCC measures as the batch sizes increases (Figures 2D-F). Learning rate was the most influential factor on the performance of the DNN. Smaller learning rates yielded better performances and there were significant increases in all measures (accuracy, precision, recall, F1, AUC, MCC) as the learning rate decreases (Figures 2G-L). Although there were significant alterations in accuracy, precision and recall (Figures 2M-O), however these improvements were not valid since smaller or larger weights did not significantly increase the F1, AUC and MCC (Figures 2P-S). On the other hand, despite there was no significant changes on the mean performances of the F1 and MCC, larger minority class weights yielded narrower confidence intervals, which indicates more precise estimates.

Consistent with the above results, hierarchical clustering results showed that the most important factor that affects the performance of the DNN is learning rate. Moreover, the second most important factor is found to be as the batch size. Finally, as stated earlier, the minority class weight was not a significant factor that improve the performance of the DNN. The hierarchical clustering plot is given in the Figure 3. In the light of these findings, smaller learning rate and larger batch size combination yield the best performance for the DNN. On the other hand, the minority class weight improves the performance measures, including accuracy, precision and recall, however, these measures are not good indicator of the performance under imbalanced dataset. Moreover, larger minority class weight did not significantly improve the mean performance of the F1, AUC and MCC. Therefore, as can be seen from the hierarchical clustering plot in the Figure 3, minority class weight is not an influential factor on the performance of the DNN.

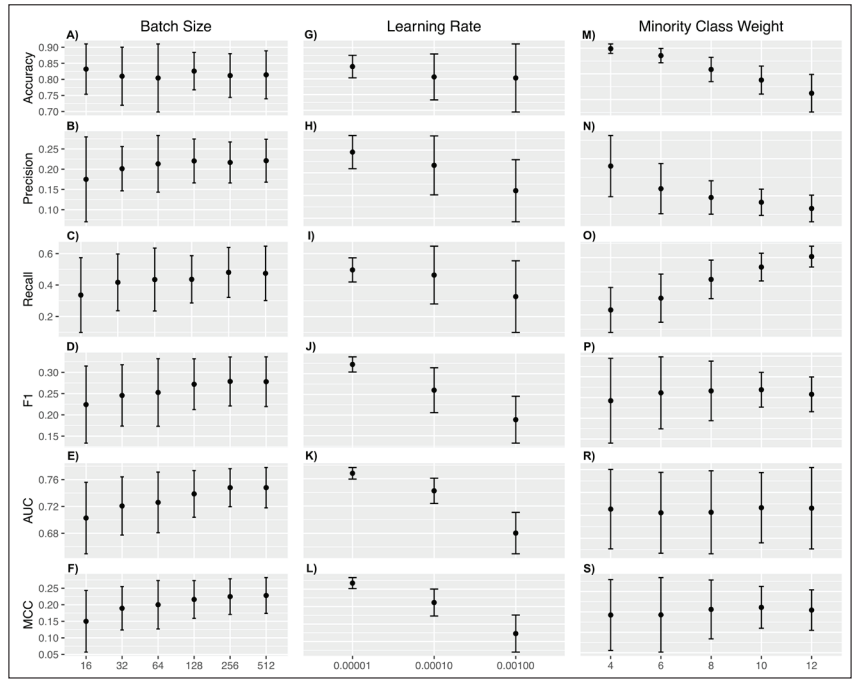

FIGURE 2: Performance comparison for batch size, learning rate and minority class weight (error bars represent mean and standard deviation).
AUC: Area under curve, MCC: Matthews correlation coefficient.

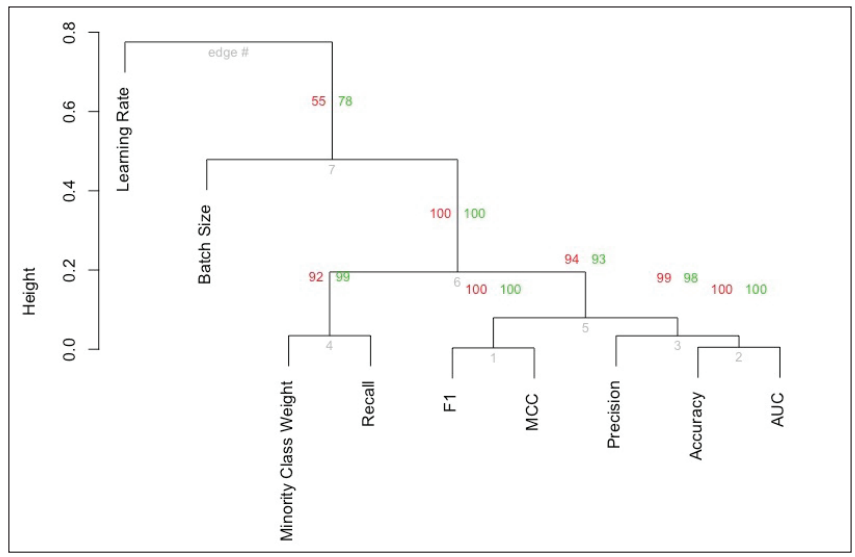

FIGURE 3: Hierarchical clustering results for performance measures and batch size, learning rate and minority class weight.
AUC: Area under curve, MCC: Matthews correlation coefficient.

The best 5 performed DNN model settings and corresponding performance measures are given in Table 1. For these best performed models, the batch sizes were 256 and 512 and the minority class weights were 4, 6 and 8. Finally, the learning rate was 0.00001 in all best 5 performed models.

The accuracy results were between 0.86 and 0.89, the precision results were between 0.26 and 0.31, the recall results were between 0.40 and 0.49, the F1 results were between 0.34 and 0.35, the AUC results were between 0.77 and 0.78 and the MCC was 0.29 for all 5 best performed models.

TABLE 1: The five best performed DNN model results.

Batch Size	Minority Class	Learning Rate	Accuracy	Precision	Recall	F1	MCC	AUC
256	4	0.00001	0.88	0.29	0.43	0.35	0.29	0.77
256	6	0.00001	0.86	0.27	0.47	0.34	0.29	0.77
512	4	0.00001	0.89	0.31	0.40	0.35	0.29	0.78
512	6	0.00001	0.87	0.28	0.47	0.35	0.29	0.78
512	8	0.00001	0.86	0.26	0.49	0.34	0.29	0.78

DNN: Deep neural networks, AUC: Area under curve, MCC: Matthews correlation coefficient.

DISCUSSION

In the era of big data, there is a great need for databases which contain labelled datasets for deep learning applications. PubChem is the largest chemical biology database and includes huge amount of publicly available bioassay datasets for compound activity prediction and classification. Ramsundar et al (2015) used this database to create a dataset, which includes 128 bioassay experiments, to perform multi-task deep learning.[20] ChEMBL is another open-access database, which contains binding, functional and ADMET (absorption, distribution, metabolism, excretion and toxicity) information for huge number of bioactive compounds.[27] Moreover, MoleculeNet, which is another publicly available database, contains more than 700,000 curated compounds from publicly available databases, including PubChem and ChEMBL, and it intends to create standard benchmark datasets in order to compare performances of different machine learning methods.[39]

Our performance measures in the Table 1 showed that the top 5 DNN models overall performed well with respect to accuracy and AUC. However, performances of the models were not adequate in terms of precision, recall, F1 and MCC. This indicates that different DNN architectures and different parameterization should be tried in the future studies to increase the classification performance of the DNN.

The performance of a DNN can be affected by many different parameters, including architecture of the network, type of loss function and optimizer, batch size, learning rate, etc. Among those parameters, we investigated effects of batch size, learning rate and minority class weight on the performance of the DNN. Learning rate is found to be the most influential factor on performance of the DNN. Learning rate is a hyper-parameter, which controls the weight adjustment of the DNN with respect the loss gradient. The optimal value of learning rate is dependent on the topology of the loss function. It is clear that there is a trade-off between learning rate and time to train the network. The small learning rate provides more reliable training but optimization will be time consuming because of the small steps towards the minimum of the loss function. On the other hand, the large learning rate will take less time to train the network, however the weight changes will become so big and the optimizer will overshoot the minimum of the loss function where training is not converged. In our study, smaller learning rates yielded better performances compared larger ones. However, further investigations are needed to reveal the optimal learning rate for the DNN. Smith (2015) offers a method, named cyclical learning rates, to detect optimal learning rate.[40] This method allows the learning rate cyclically vary between pre-specified boundaries instead of monotonically decreasing learning rates.

Batch size is found to be another important factor that affect the performance of the DNN models. Because of their lack of generalization ability, it is suggested that the larger batch sizes yielded worse performance than the small batch sizes for the DNN model.[41] Interestingly, in our study we found that larger batch sizes yielded better performances. This may presumably occur because of the imbalanced nature of our dataset. As reported by Keskar et al. (2017), the performance of the DNN increases until a certain number of batch size and it dramatically drops after this threshold. Therefore, this phonemone may exist in our case and bigger batch sizes should be tested in future studies in order to clarify such threshold for the batch size.

Although we stated that the minority class weight did not improve the performance measures, when we set the weights less than 4, the performance dramatically decreased. This means that minority class weight has a significant impact until a specific point. Moreover, as can be seen from the Figure 2P and Figure 2S, the standard deviation decreases as the minority class weights increase for F1 and MCC. Subsequently, we can conclude that minority class weight can be considered as a significant factor that can alter the performance of the DNN model, especially in the case of imbalanced dataset. Therefore, deeper investigations are needed to clearly understand the effects of the minority class weights on the performance of the DNN in the case of imbalanced dataset in the future studies.

CONCLUSION

PubChem is a great resource for analyzing and implementing deep learning to retrieve useful information regarding activity of chemical compounds. In this study, we compared performances of different settings of the DNN models using publicly available bioassay dataset from the PubChem database. We found that smaller learning rates and larger batch sizes are the most important factors that affect the performance of the DNN model. Our best performed model yielded 89% accuracy, 0.35 F1 score, 0.29 MCC and 0.78 AUC. These results are promising for use of deep learning in VS of small drug molecules from publicly available databases, such as PubChem.

Deep learning is a great tool to analyze such large databanks for chemical libraries. The automated processes, including retrieving data, pre-processing, labelling, training models and developing tools are strongly needed in the early phase of drug discovery studies. Future studies may include developing cloud-based systems that can automatically detect active compounds directly from publicly available databases (i.e. PubChem, ChEMBL) using deep learning methods.

Acknowledgment

I would like to thank my colleagues Gökmen Zararsız and Dinçer Göksülük for their useful suggestions and comments.

Source of Finance

During this study, no financial or spiritual support was received neither from any pharmaceutical company that has a direct connection with the research subject, nor from a company that provides or produces medical instruments and materials which may negatively affect the evaluation process of this study.

Conflict of Interest

No conflicts of interest between the authors and / or family members of the scientific and medical committee members or members of the potential conflicts of interest, counseling, expertise, working conditions, share holding and similar situations in any firm.

Authorship Contributions

Idea/Concept: *Selçuk Korkmaz;* ***Design:*** *Selçuk Korkmaz;* ***Control/Supervision:*** *Selçuk Korkmaz;* ***Data Collection and/or Processing:*** *Selçuk Korkmaz;* ***Analysis and/or Interpretation:*** *Selçuk Korkmaz;* ***Literature Review:*** *Selçuk Korkmaz;* ***Writing The Article:*** *Selçuk Korkmaz;* ***Critical Review:*** *Selçuk Korkmaz;* ***References and Fundings:*** *Selçuk Korkmaz*

REFERENCES

1. Korkmaz S, Zararsiz G, Goksuluk D. MLViS: a web tool for machine learning-based virtual screening in early-phase of drug discovery and development. PLoS One. 2015;10(4):e0124600. PMID: 2592888 [Crossref] [PubMed] [PMC]
2. Broach JR, Thorner J. High-throughput screening for drug discovery. Nature. 1996;384(6604 Suppl):14-6. PMID: 8895594
3. Sadowski J, Kubinyi H. A scoring scheme for discriminating between drugs and nondrugs. J Med Chem. 1998;41(18):3325-9. PMID: 9719584 [Crossref] [PubMed]
4. Byvatov E, Fechner U, Sadowski J, Schneider G. Comparison of support vector machine and artificial neural network systems for drug/nondrug classification. J Chem Inf Comp Sci. 2003;43(6):1882-9. PMID: 14632437 [Crossref] [PubMed]
5. Zernov VV, Balakin KV, Ivaschenko AA, Savchuk NP, Pletnev IV. Drug discovery using support vector machines. The case studies of drug-likeness, agrochemical-likeness, and

enzyme inhibition predictions. J Chem Inf Comp Sci. 2003;43(6):2048-56. PMID: 14632457 [Crossref] [PubMed]
6. Korkmaz S, Zararsiz G, Goksuluk D. Drug/nondrug classification using Support Vector Machines with various feature selection strategies. Comput Methods Programs Biomed. 2014;117(2):51-60. PMID: 25224081 [Crossref] [PubMed]
7. Fang JS, Yang RY, Gao L, Zhou D, Yang SQ, Liu AL, et al. Predictions of BuChE ınhibitors using support vector machine and naive Bayesian classification techniques in drug discovery. J Chem Inf Model. 2013;53(11):3009-20. PMID: 24144102 [Crossref] [PubMed]
8. Sun H. A naive bayes classifier for prediction of multidrug resistance reversal activity on the basis of atom typing. J Med Chem. 2005;48(12):4031-9. PMID: 15943476 [Crossref] [PubMed]
9. Miller DW. Results of a new classification algorithm combining K nearest neighbors and recursive partitioning. J Chem Inf Comp Sci. 2001;41(1):168-75. PMID: 11206369 [Crossref] [PubMed]
10. Abdo A, Chen B, Mueller C, Salim N, Willett P. Ligand-based virtual screening using Bayesian Networks. J Chem Inf Model. 2010;50(6):1012-20. PMID: 20504032 [Crossref] [PubMed]
11. Ehrman TM, Barlow DJ, Hylands PJ. Virtual screening of Chinese herbs with random forest. J Chem Inf Model. 2007;47(2):264-78. PMID: 17381165 [Crossref] [PubMed]
12. Plewczynski D, von Grotthuss M, Rychlewski L, Ginalski K. Virtual high throughput screening using combined random forest and flexible docking. Comb Chem High Throughput Screen. 2009;12(5):484-9. PMID: 19519327 [Crossref]
13. Chen H, Engkvist O, Wang Y, Olivecrona M, Blaschke T. The rise of deep learning in drug discovery. Drug Discov Today. 2018;23(6):1241-50. PMID: 29366762 [Crossref] [PubMed]
14. Wu Y, Schuster M, Chen Z, Le QV, Norouzi M, Macherey W, et al. Google's Neural Machine Translation System: bridging the gap between human and machine translation. arXiv e-prints 2016. https://arxiv.org/abs/1609.08144
15. Hannun A, Case C, Casper J, Catanzaro B, Diamos G, Elsen E, et al. Deep speech: scaling up end-to-end speech recognition. arXiv e-prints 2014. https://arxiv.org/abs/1412.5567
16. Hughes M, Li I, Kotoulas S, Suzumura T. Medical text classification using convolutional neural networks. Stud Health Technol Inform. 2017;235:246-50. PMID: 28423791
17. Young T, Hazarika D, Poria S, Cambria E. Recent trends in deep learning based natural language processing. arXiv e-prints 2017. https://arxiv.org/abs/1708.02709
18. Ma J, Sheridan RP, Liaw A, Dahl GE, Svetnik V. Deep neural nets as a method for quantitative structure-activity relationships. J Chem Inf Model. 2015;55(2):263-74. PMID: 25635324 [Crossref] [PubMed]
19. Mayr A, Klambauer G, Unterthiner T, Hochreiter S. DeepTox: toxicity prediction using deep learning. Front Environ Sci. 2016;3. https://www.frontiersin.org/articles/10.3389/fenvs.2015.00080/full [Crossref]
20. Ramsundar B, Kearnes S, Riley P, Webster D, Konerding D, Pande V. Massively multitask networks for drug discovery. arXiv e-prints 2015. https://arxiv.org/pdf/1502.02072.pdf
21. Wang Y, Xiao J, Suzek TO, Zhang J, Wang J, Zhou Z, et al. PubChem's BioAssay Database. Nucleic Acids Res. 2012;40(Database issue):D400-12. PMID: 22140110 [Crossref] [PubMed] [PMC]
22. Rohrer SG, Baumann K. Maximum unbiased validation (MUV) data sets for virtual screening based on PubChem bioactivity data. J Chem Inf Model. 2009;49(2):169-84. PMID: 19434821 [Crossref] [PubMed]
23. Mysinger MM, Carchia M, Irwin JJ, Shoichet BK. Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking. J Med Chem. 2012;55(14):6582-94. [Crossref] [PubMed] [PMC]
24. Tice RR, Austin CP, Kavlock RJ, Bucher JR. Improving the human hazard characterization of chemicals: a Tox21 update. Environ Health Perspect. 2013;121(7):756-65. PMID: 23603828 [Crossref] [PubMed] [PMC]
25. Koutsoukas A, Monaghan KJ, Li X, Huan J. Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. J Cheminform. 2017;9:42. [Crossref] [PubMed] [PMC]
26. Lenselink EB, Ten Dijke N, Bongers B, Papadatos G, van Vlijmen HWT, Kowalczyk W, et al. Beyond the hype: deep neural networks outperform established methods using a ChEMBL bioactivity benchmark set. J Cheminform. 2017;9(1):45. PMID: 29086168 [Crossref] [PubMed] [PMC]
27. Gaulton A, Bellis LJ, Bento AP, Chambers J, Davies M, Hersey A, et al. ChEMBL: a large-scale bioactivity database for drug discovery. Nucleic Acids Res. 2012;40(Database issue):D1100-7. PMID: 21948594 [Crossref] [PubMed] [PMC]
28. Dander A, Mueller LA, Gallasch R, Pabinger S, Emmert-Streib F, Graber A, et al. [COMMODE] a large-scale database of molecular descriptors using compounds from PubChem. Source Code Biol Med. 2013;8(1):22. PMID: 24225386 [Crossref] [PubMed] [PMC]
29. Cheng T, Pan Y, Hao M, Wang Y, Bryant SH. PubChem applications in drug discovery: a bibliometric analysis. Drug Discov Today. 2014;19(11):1751-6. PMID: 25168772 [Crossref] [PubMed] [PMC]
30. Kim S. Getting the most out of PubChem for virtual screening. Expert Opin Drug Discov. 2016;11(9):843-55. PMID: 27454129 [Crossref] [PubMed] [PMC]
31. QHTS Assay for Inhibitors of Aldehyde Dehydrogenase 1 (ALDH1A1). 2008. at https://pubchem.ncbi.nlm.nih.gov/bioassay/1030#section=Identity.
32. Yap CW. PaDEL-descriptor: an open source software to calculate molecular descriptors and fingerprints. J Comput Chem. 2011;32(7):1466-74. PMID: 21425294 [Crossref] [PubMed]
33. Chollet F, Allaire JJ. Deep Learning with R. Manning Publications Co.; 2018. p.335.
34. Larochelle H, Bengio Y, Louradour J, Lamblin P. Exploring strategies for training deep neural networks. J Mach Learn Res. 2009;10:1-40.
35. Patterson J, Gibson A. Deep Learning: A Practitioner's Approach. 1st ed. Beijing: O'Reilly; 2017. p.532. Kırmızı renkle yazıldığı şekilde bulundu.
36. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res. 2014;15:1929-58.
37. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature. 1986;323:533-6. [Crossref]
38. Dahl GE, Sainath TN, Hinton GE. Improving deep neural networks for LVCSR using rectified linear units and dropout. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing; 2013. p. 8609-13. [Crossref]
39. Wu Z, Ramsundar B, Feinberg EN, Gomes J, Geniesse C, Pappu AS, et al. MoleculeNet: a benchmark for molecular machine learning. Chem Sci. 2018;9(2):513-30. [Crossref] [PubMed] [PMC]
40. Smith LN. Cyclical learning rates for training neural networks. arXiv e-prints 2015. https://arxiv.org/abs/1506.01186
41. Shirish Keskar N, Mudigere D, Nocedal J, Smelyanskiy M, Tang PTP. On large-batch training for deep learning: generalization gap and sharp minima. arXiv e-prints 2016. https://arxiv.org/abs/1609.04836