Question 3 Report

Introduction

Machine learning (ML), a sub-branch of artificial intelligence (AI), is a currently widely used tool for completing tasks that are usually done by humans. ML technologies are helpful for reducing humans' workload on tedious tasks with better efficiency. The most commonly used types of ML technologies are supervised and unsupervised learning. In supervised learning, we use labelled nominal data to train the ML model, and the trained model can be used to classify unseen data during testing. If the data is numeric, we may have a regression task for supervised learning. The trained model can then be applied for prediction. For unsupervised learning or clustering, we do not have labelled data. Instead, the model learns to group the data into different clusters.

These days, with the development of technology, more and more data are available in the biomedical field, such as medical imaging and genomic data. Thus, ML is also used in biomedical areas, such as identifying functional networks in the human brain [1] and tumour purity prediction [2]. Also, both supervised and unsupervised learning are utilized [3]. These ML technologies are important for disease diagnosis[4], treatment [5], and even drug development [6]. To ensure the trained ML model generates accurate and reliable results for these applications in actual practice, we need to evaluate the model to understand its ability to perform specific tasks.

In a general way, we have different data for training and testing. For example, we may have the data of 50 subjects for training and the data for another 50 subjects that are not seen by the ML model during training for testing and evaluation. The most common validation method to avoid overlapping in test sets is called cross-validation. To use this validation method, we split the data into n subsets of equal size and each time we build the model, we use one subset as the test data, and the remaining (n-1) subsets are train data. Thus, the data we used for the test and training are independent in each iteration of the total n iterations. The overall accuracy can be calculated using the averaged accuracy of all iterations.

However, as Wang et al. [6] mentioned, functional doppelgangers can cause the overestimation of the model performance because of the high similarity between the train and test data. If we are unaware of the potential functional doppelgangers and their effects, the conclusion generated based on the results from models that show exemplary performance may even be inaccurate and unreliable. Additionally, there are many data doppelgangers in biological data, such as predicting the protein functions and drug discovery [6]. Thus, it is important to identify the doppelgangers and try to avoid such issues.

Not Unique to Biomedical Data

Although this research paper discussed the abundance of doppelgangers in biological data, I believe the doppelganger effects also exist in other data types. The reason for this is that I thought of examples in other fields. For instance, let's think of one application of machine learning, the age estimation from facial images, similar to the research done by Abbas et al. [7].

In this scenario, we want to train a machine learning classifier to accurately classify different input images of human faces into different age groups. It is possible that the classifier learns well and is able to do the classification with very high accuracy. As mentioned by Abbas et al.

[7], the aging process can be influenced by many factors, including environment, lifestyle, and expressions. Then, if we use similar facial images with similar facial expressions for both training and testing and get the model with high accuracy, what will happen if we use images of humans in the same age group but has relatively different expressions or with varying angles of photographing? This model will be likely to classify these dissimilar images to the wrong age group. Thus, in this case, the doppelgangers can also lead to an overestimate of the ML model performance.

The example I used is similar to the protein example mentioned in the given paper [6]. The similar images in the same group for train and testing can be considered as the proteins with similar sequencing in the same functional class. Test on less similar facial images of the same age group is also possible to lead to false results, just like testing on protein with the less similar sequence but have similar functions [6].

We can also generalize the issue to a broader perspective based on the age estimation and the protein example. If we train and evaluate the ML model using independent data but having some similar patterns or distribution, we may get an accurate classifier with good evaluation results. For instance, the model accurately classified some data into *class A*, and these train and test data all have similar *pattern B*. However, if we have a new sample with *pattern C* in place of *pattern B*, but it still belongs to *class A,* the classifier may falsely classify it into another group. Thus, the evaluation using the test data similar to the train data will inflate the performance. This issue is possible to occur in different data, not just biomedical data, so I think the doppelganger effects are not unique to biomedical data.

Ways to Avoid

As I think these doppelganger effects are not just for biomedical data, we really need to be aware of it when evaluating our ML models, and it is even more important to attempt to avoid them. Based on the recommendations proposed by Wang et al. [6], I came up with some ideas about how to prevent the doppelganger effects.

To begin with, my first idea is very similar to the third recommendation in this paper [6]. My initial thought was to involve more variations in the data for testing, so I totally agree that we need to perform the validation on as many independent datasets as possible [6]. However, this may be somewhat difficult to achieve if the amount of data available is limited. This is a realistic problem for biomedical data related to some rare diseases.

Because of the difficulty of involving a large amount of data in some cases, we need to have some more readily available methods to increase the variations in the data to reduce the relative amount of functional doppelgangers in the data. Based on my knowledge in database management, I recalled the salting in password encryption. In order to protect our password, the human-readable passwords are usually transferred into a non-human readable version using the hashing function. However, the hashed values can still be attacked through brute-force, dictionary, or hybrid attack [8]. As a result, the idea of salting [9] was introduced by researchers to further improve the storage security of passwords. The main idea of salting is to add a random string to the password before hashing to strengthen the password [9]. The idea of salting was not only used for password encryption but also used to protect biometric template that represents the features extracted from images [10]. Thus, I also thought it could be modified to deal with the doppelganger effects.

My idea is similar to salting but different on the scale. In password encryption, we add random strings to the password, while in our case, to fight the doppelganger effects, we add random noises to both the train and test datasets. Considering the protein function prediction task, we add a new component to the model, which is a generator (G) that can create fake "proteins" for training and testing. G is trained to create better fakes that simulate the real protein data. The aim of the classifiers in this model is still to predict the functions accurately, but it should also be able to discriminate the fake ones. This idea is then similar to the generative adversarial networks (GAN) proposed by Goodfellow et al. [11] in 2014.

Then, why does adding these "noises" help to reduce the doppelganger effects? As mentioned in the given research paper, more doppelganger pairs can lead to more overestimation of the model performance [6]. By introducing the fake protein data generated by G, I would expect the proportion of doppelganger pairs in the total train and test datasets are reduced. For example, if we have 2*m real data, 2*n fake data (2*n < m), and x doppelganger pairs. The original proportion of doppelganger pairs is $\frac{x}{2*m}$. However, after adding the 2*n fake data, the proportion became $\frac{x}{2*n+2*m}$, which must be smaller than the original one because (2*n+2*m) > 2*m.

Besides, we can also use G to create fake test data. Thus, when we do validation on the test data, which is a mixture of real and fake data, we also evaluate the performance of the classifier to distinguish between them. During the training, we aim to train G to create close-to-real data, but these data can be similar to the train data in different classes. As a result, the fake data can be similar to real data but are unlikely to be similar to only a specific pattern, so it is unlikely to introduce new doppelganger pairs in the fake class, giving a more reliable evaluation of the ML models. The idea of the model is shown in *Figure 1* below.

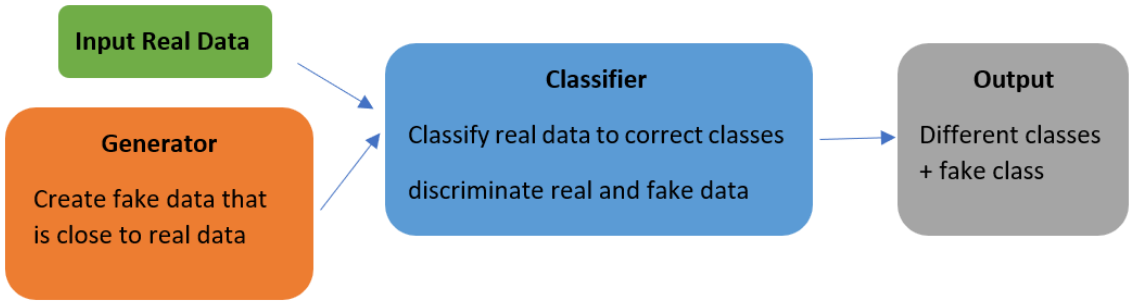

Figure 1 Idea of the Way to Avoid Doppelganger Effects.

Risks and Problems

Although this method may have the potential to avoid the doppelganger effects, there are still some risks or problems I can think of for this method I proposed. First of all, the method I explained in the previous paragraphs is still on the idea level. The implementation achievability is questionable. However, because GAN has been successfully implemented [11] and can be modified to perform domain adaptation, it is possible to implement such a deep learning model in *Figure 1*.

Even this model can be implemented, its effect of avoiding doppelganger effects should also be tested and verified in research. It is possible that this method is not effective enough. If this is the case, we need to first tune some model parameters, such as the fraction of fake involved in the training and testing. Too much fake data may be overwhelming, while too little may be insufficient to avoid the doppelganger effects. Many experiments need to be done to find the optimal fraction to make sure we use the generator to avoid the doppelganger effects to the maximum extent. If we cannot find the optimal results, we may need to design more suitable and effective algorithms for the modelling.

Conclusion

From the given research paper [6], I learned about doppelganger effects and their effects on the evaluation of ML models. I got to know that the doppelganger effects are critical to be aware of and be avoided in bioinformatics or the use of other types of data in ML. The method of using pairwise Pearson's correlation coefficient (PPCC) to identify doppelganger seems pretty good, and the recommendations of avoiding the doppelganger effects by using meta-data, stratification, and divergent validation make a lot of sense to me [6]. These information also hinted me to come up with an idea combining salting and GAN to avoid the doppelganger effects. Because I currently have limited knowledge and experience in ML, I am passionate to do more research in this area and also biomedical data to learn more and develop models that are realistic and useful in real practice.

References

[1] V. M. Vergara, A. R. Mayer, K. A. Kiehl, and V. D. Calhoun, "Dynamic functional network connectivity discriminates mild traumatic brain injury through machine learning," *NeuroImage: Clinical,* vol. 19, pp. 30-37, 2018.

[2] M. U. Oner *et al.*, "Obtaining spatially resolved tumor purity maps using deep multiple instance learning in a pan-cancer study," *Patterns,* p. 100399, 2021.

[3] S. Hussein, P. Kandel, C. W. Bolan, M. B. Wallace, and U. Bagci, "Lung and pancreatic tumor characterization in the deep learning era: novel supervised and unsupervised learning approaches," *IEEE transactions on medical imaging,* vol. 38, no. 8, pp. 1777-1787, 2019.

[4] M. Z. Islam, M. M. Islam, and A. Asraf, "A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images," *Informatics in medicine unlocked,* vol. 20, p. 100412, 2020.

[5] S. He, L. G. Leanse, and Y. Feng, "Artificial intelligence and machine learning assisted drug delivery for effective treatment of infectious diseases," *Advanced Drug Delivery Reviews,* vol. 178, p. 113922, 2021.

[6] L. R. Wang, L. Wong, and W. W. B. Goh, "How doppelgänger effects in biomedical data confound machine learning," *Drug discovery today,* 2021.

[7] A. R. Abbas and A. R. Kareem, "Intelligent age estimation from facial images using machine learning techniques," *Iraqi Journal of Science,* pp. 724-732, 2018.

[8] L. Bošnjak, J. Sreš, and B. Brumen, "Brute-force and dictionary attack on hashed real-world passwords," in *2018 41st international convention on information and communication technology, electronics and microelectronics (mipro)*, 2018: IEEE, pp. 1161-1166.

[9] W. Khowfa and O. Silasai, "The Efficiency of using Salt Against Password Attacking," *JOURNAL OF SOUTHERN TECHNOLOGY,* vol. 12, no. 1, pp. 217-227, 2019.

[10] V. K. Gunjan, P. S. Prasad, and S. Mukherjee, "Biometric template protection scheme-cancelable biometrics," in *ICCCE 2019*: Springer, 2020, pp. 405-411.

[11] I. Goodfellow *et al.*, "Generative adversarial nets," *Advances in neural information processing systems,* vol. 27, 2014.