The article discusses the issue of identifying data doppelgängers in biomedical data and how those functional data doppelgängers can confound machine learning models. According to the authors, data doppelgängers, also known as functional doppelgängers, are similar samples that occur in both training and validation sets. Data doppelgängers are most likely to appear in the same class but as different objects. Because of the presence of data doppelgängers, most machine learning models will have unreliable high-accuracy validation results, no matter what random features are selected to train the model. In other words, these trained machine learning models will only have high accuracy when data scientists use this specific dataset and will be expected to perform poorly when new data comes in. Those machine learning models cannot generalize the actual pattern between explanatory variables and response variables. Thus, authors try to detect data doppelgängers that interfere the machine learning model results.

In my opinion, doppelganger effects are not unique to biomedical data. Because many other independently derived data could be different objects but falsely classified into the same class and thus become functional data doppelgängers that interfere with the model learning. For example, when data scientists classify text into different industries with Natural Language Processing (NLP), words belonged to different categories could be falsely classified into the same class if we don't use the industrial specific NLP library to differentiate them. In specific, if we want to classify a Twitter comment into different categories, NLP will confuse when it comes to words such as "futures". The word "futures", in finance, means future contract, a type of financial commodity. However, the data collecting process using Twitter could easily label all tweets containing the word "futures" as lifestyle or education. This process will generate keyword data doppelgängers, which should have belonged to different classes but now classified as the same class.

Another example is the image recognition of handwriting digits, which can be used in postal mail sorting, bank check processing, and form data entry based on Shamim et al's research (Shamim et al., 2018). Handwriting number 6 or number 9 could easily be mistaken as number 0 in image recognition. Because the circle part of the number 6 or 9 can overlap with the number 0, especially when people write the circle part relatively large. As a result, some of the number 6 and number 9 handwriting images could be falsely classified as number 0 in the dataset. If we include those falsely classified samples in both the train and test datasets, we will have data doppelgängers that confound machine learning models.

We already know that data doppelgängers will affect the accuracy of machine learning models. In my perspective, there are several ways to avoid and detect data doppelgängers in machine learning models. Next, I want to discuss more about how we can avoid and check the data doppelganger effect in machine learning models.

In the beginning, we could consider more from the data collection perspective. Data availability could be a very critical issue. According to Benowitz, mice and humans share approximately 70 percent of the same protein-coding gene sequences, which is just 1.5 percent of these genomes (Benowitz, 2015). If most of the gene sequencing features we generate happened to locate within the identical protein-coding gene sequences, it would be hard for machine learning models to differentiate mice and humans using these gene sequence data. Take an extreme example, if all columns are data collected from the same protein-coding gene sequences of humans and mice, and some of our samples are data doppelgängers – mice falsely labeled as human. It would be very hard for machine learning models to determine which samples are data doppelgängers because we lack critical data. However, if we collect more features and have more data columns related to the remaining 30 percent different gene sequencing between humans and mice, machine learning models will have more confidence to differentiate those two different creatures. Just as Vabalas et al. propose in the research article, "with a larger training sample size, models have higher statistical power to learn a pattern discriminating between classes and achieve higher performance" (Vabalas et al., 2019). As a result, a larger sample size and more relevant data features can decrease the risks for data doppelgängers to confound the machine learning models.

The second approach we could use is feature selection methods. Chowdhury and Turin claim that since all features are included in the model from the beginning of the process, backward elimination offers the benefit to assess the joint predictive ability of variables (Chowdhury and Turin, 2020). The least significant variables are also eliminated early on through backward selection, leaving the most significant variables in the model. We can run backward selection using generalized linear models. Then we generate a line graph of the best Sequential Feature Selector (SFS) performance which shows the tendency of model performance changes each time the number of features increases. After that, we choose the model features with the lowest AIC score, X features p-values < 0.05, and highest performance accuracy. Feature selection helps us select the most critical features with high prediction power on y variables and eliminate the redundant and irrelevant ones. For example, in the Kaggle dataset contributed by Barupal on the case study of human serum metabolome variability, we could apply feature selection on explanatory features such as intra-class correlation coefficient (ICC) level, biomedical type and biomedical mass to predict if each person is a diabetes patient or not (Barupal, 2020). By using feature selection, we can decrease the probability of getting data doppelgängers because we only select the most relevant features in our data set. As a result, backward selection keeps the most significant explanatory variables that have the most explaining powers; By using the backward selection method, we could potentially remove columns with data doppelgängers in the early stage and avoid the negative effect of data doppelgängers.

The third approach is to change the number of data doppelgängers pairs in both the train dataset and test dataset. Before we start, we should stratify train-test split, keep the random state of train-test split constant, and use only one explanatory variable X and one response variable y each time. Because we use only one X column and y column each time, we avoid the effect of other X columns. Then we use stratify train-test split to make sure the ratio of each category in the selected column X or y is the same in both train and test dataset. For example, if the y column contains only 5% of patients getting strokes, using stratify train-test split will ensure 5% of patients in both train and test data. After that, we keep the random state constant each time so that the accuracy will not be affected by the random train test split. After our preparations, we use very small test size data such as 10% to test the model. Then we increase the test size to 40% and test our model again. Because the test size is very small initially, the number of potential data doppelgängers pairs included in both train and test data will be relatively small. After we increase the test size to 40%, our model will potentially have more data doppelgängers pairs in both train and test data. In this case, we can achieve different precision and recall each time we run the model. If the precision and recall changes are very volatile and significant, we know that this specific X column contains data doppelgängers that change the model accuracy in an unreasonable range, such as 60% accuracy when using 10% test data compared to 95% accuracy when we use 40% test data. As a result, by using only one X column and y column each time, we can control the size of test data to manipulate the data doppelgängers pairs in both the train and test dataset, which results in detecting abnormal model performance and identifying X feature with data doppelgängers.

The fourth approach is to use false positive rate (FPR) to evaluate machine learning model performance, which is a great indicator of the data doppelgängers effect. After we train the model with selected features, we check the confusion matrix performance and false positive rate to compare the performance of these two models with different test sizes. We know that data doppelgängers are usually the cases in the same class but as different patients. In other words, some of the patients originally not in this class but are falsely classified into this class and confound the machine learning models. For instance, patient A is not a diabetes patient (actually negative) but is classified as a diabetes patient (predicted positive). In this case, we should look at the false positive rate, which is described by Bruce as fraction of negative cases falsely identified as positive cases (Bruce, 2022). Because the machine learning models falsely learn the pattern of data doppelgängers, those models are more likely to classify patients without diabetes as diabetes patients. In other words, we expect the false positive rate of the model to increase as the data doppelgängers increase in both train and test data. As a result, false positive rate (FPR) would be a great way to check the data doppelgängers effect in the model.

The fifth approach, another great way to check the data doppelgängers effect, is to train probabilistic classification models and select a reasonable probability cut-off threshold. Instead of using simple classification models that only output sample A will be predicted as category B, we can train probabilistic classification models that output sample A has a 0.632 probability of being classified as category B. In the former case, we can only see the category result. But in the latter case, probabilistic classification models give us more perspective on the probability of sample A to be classified as category B. According to Perelas, high serum beta-hydroxybutyrate levels in many conditions are associated with insulin deficiency, which leads to diabetic ketoacidosis (Perelas, 2021). Based on the author's research, the standard range of beta-hydroxybutyrate level should be less than 0.4 to 0.5 mmol/L, and levels above 1 mmol/L call for additional action, whereas levels above 3 mmol/L demand emergency medical attention. Then, for example, if patient A has the beta-hydroxybutyrate level of 1.3, patient A may have a 0.632 probability of getting diabetes. However, if patient B has a beta-hydroxybutyrate level of 2.5, patient B may have a 0.915 probability of getting diabetes. If our probability cut-off threshold for the y prediction (getting diabetes) is 0.5, we cannot differentiate patient A and patient B really well, because patient A has a relatively low diabetes probability of 0.632, which is very close to the cut-off of non-diabetes samples. But patient B has a very high probability of 0.915 of getting diabetes. To solve this problem, after we train the model and get the y prediction for our sample, we can loop over different cut-off values from 0 to 1 with 0.05 as each step (0.05, 0.1, 0.15, 0.2…0.95, 1) to determine what probability should be counted as 1 (diabetes patients). Then we can generate classification reports for each cut-off case and select the one with the lowest recall and false positive rate. If a smaller proportion of y=1 exists in the binary classification problem (y is either 1 or 0), a higher cut-off will usually work better. For data doppelgängers, in this case, we are likely to have different patient samples but classified as the same class – healthy people classified as diabetes patients. But because data doppelgängers are falsely classified, even though they are classified as diabetes patients, they are more likely to have a relatively low probability of classified as diabetes patients. By using probabilistic classification models and changing the probability cut-off threshold, we could potentially get rid of the data doppelgängers in our data set. Specifically, if we change the probability cut-off threshold to 0.75, patient A, a potential data doppelgänger, will be classified as a healthy individual rather than a diabetes patient. This method could be extremely useful, especially when we only have a binary categorical y column, which contains only two categories represented by 1 and 0 in the dataset, because changing the cut-off could easily switch a sample from one category to the other. As a result, using probabilistic classification models and changing the reasonable probability cut-off threshold can eliminate partial effects of data doppelgänger because a more accurate probability cut-off threshold will generate a more definite y prediction category result and eliminate data doppelganger with relatively low y prediction probability.

References

Barupal, Dinesh. “Human Serum Metabolome Variability.” *Kaggle*, 5 Oct. 2020, https://www.kaggle.com/datasets/desertman/human-serum-metabolome-variability.

Benowitz, Steven. “New Comprehensive View of the Mouse Genome Finds Many Similarities and Striking Differences with Human Genome.” *National Institutes of Health*, U.S. Department of Health and Human Services, 20 Aug. 2015, https://www.nih.gov/news-events/news-releases/new-comprehensive-view-mouse-genome-finds-many-similarities-striking-differences-human-genome#:~:text=Mice%20and%20humans%20share%20approximately,1.5%20percent%20of%20these%20genomes.

Bruce, Peter. “False Positive Rate - It's Not What You Might Think.” *Statistics.com: Data Science, Analytics & Statistics Courses*, Statistics.com, LLC, 6 Apr. 2022, https://www.statistics.com/famous-errors-in-statistics-its-not-what-you-might-think/.

Chowdhury, Mohammad Ziaul Islam, and Tanvir C Turin. “Variable Selection Strategies and Its Importance in Clinical Prediction Modelling.” *Family Medicine and Community Health*, BMJ Specialist Journals, 1 Feb. 2020, https://fmch.bmj.com/content/8/1/e000262.

Perelas, Apostolos. “Beta-Hydroxybutyrate.” *Reference Range, Interpretation, Collection and Panels*, Medscape, 24 May 2021, https://emedicine.medscape.com/article/2087381-overview?reg=1#a2.

Shamim, S M, et al. “Handwritten Digit Recognition Using Machine Learning Algorithms.” *Indonesian Journal of Science and Technology*, Indonesian Journal of Science and Technology, 2018, https://ejournal.upi.edu/index.php/ijost/article/view/10795.

Vabalas, Andrius, et al. “Machine Learning Algorithm Validation with a Limited Sample Size.” *PLOS ONE*, Public Library of Science, 7 Nov. 2019, https://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0224365#:~:text=This%20shows%20a%20well-known,classes%20and%20achieve%20higher%20performance.