Mitigating Data Leakage: Enhancing Machine Learning Education with Reproducible Results

Introduction

The problem we are trying to solve with this project is the lack of awareness and understanding of data leakage in the context of machine learning education. Data leakage, which refers to unwanted inclusion of information from the test set into the training process, leads to overly optimistic evaluation of model performance and undermines the reproducibility and validity of research findings. Despite its significance, data leakage is often overlooked in introductory machine learning courses, leaving students and researchers unaware of its implications. This project seeks to address this gap by developing learning materials that effectively demonstrate instances of data leakage and its impact on results, thereby enhancing student's understanding of this critical concept and promoting reproducible and valid research practices in machine learning.

The current state of the problem revolves around the widespread occurrence of data leakage in machine learning applications across various scientific domains. Data leakage involves complex scenarios like pre-processing errors, feature selection before data splitting, temporal and group leakage. Understanding these concepts requires a solid grasp of machine learning principles, which may be challenging for students with diverse backgrounds. While numerous courses on machine learning and its applications are available online and in universities, the literature specifically addressing data leakage is notably sparse. Existing resources offer only a high-level overview of the problem, leaving a gap in detailed understanding and practical guidance on mitigating data leakage in machine learning research.

Name of the paper	**Area of application**	**Type of data leakage**	**Type of model**
Characterization of Term and Preterm Deliveries using Electrohysterograms Signatures (Khan et al. (2019))	Application in Healthcare - This paper classifies term and preterm deliveries using EHG database.	Applying over-sampling before partitioning the data into mutually exclusive training and testing sets.	Support vector machine classifier
Personalized predictive models for symptomatic Covid-19 patients using basic preconditions: Hospitalizations, Mortality, and the Need for an ICU or Ventilator (Wollenstein-Betech et al. (2020))	Application in Healthcare - This study works on developing predictive models using real-world dataset.	Non-independence between train and test sets results in data leakage.	Support Vector Machines, Sparse Logistic Regression, Random Forests, Gradient boosted decision trees
Designing a Sum of Squared Correlations framework for enhancing SSVEP based BCIs (Kumar and Reddy (2019))	Application in Healthcare - The study claims state-of-the-art in the field of steady-state visual evoked potential (SSVEP) based brain-computer interfaces (BCIs).	Data leakage due to the absence of any train-test split	Sum of Squared Correlation (SSCOR)
Learning from Failure Across Multiple Clusters: A Trace-Driven Approach to Understanding, Predicting, and Mitigating Job Terminations (El-Sayed et al. (2017))	Application in AI for IT Operations- This paper uses machine learning to address the issue of interruptions and premature terminations of jobs in large-scale computing platforms.	Temporal splitting of data (Inadvertent introduction of information from the future into the training process)	Random Forest

Table 1

Project Goals

The initial three papers in Table 1 shed light on the integration of machine learning in healthcare, while the fourth paper explores its application in the IT sector. Together, these papers underscore the wide-ranging influence of machine learning in different domains.

The research conducted by Khan et al. (2019) delves into the application of machine learning in distinguishing between term and preterm deliveries, a crucial area of study given that preterm birth stands as a primary cause of infant mortality and morbidity worldwide. Many datasets within the healthcare sector suffer from imbalances, necessitating the use of oversampling techniques to effectively implement machine learning models. Addressing this challenge provides valuable learning opportunities for students, particularly in understanding the correct implementation of oversampling techniques on such datasets.

The study by Wollenstein-Betech et al. (2020) focuses on developing personalized predictive models for critical events like hospitalization, mortality, ICU admission, and ventilator requirement, crucial for resource allocation in healthcare. In predictive modeling, selecting the predictand variable—the outcome to be predicted—from a range of features is essential. It's critical to understand the relationship between predictor and predictand variables to avoid data leakage, where the predictand variable is influenced by predictors, potentially compromising model integrity. This research underscores the importance of verifying the independence of training and testing sets, highlighting the necessity to mitigate biases and ensure the reliability of machine learning predictions.

Kumar and Reddy (2019) worked on creating a new framework to enhance the performance of brain-computer interfaces (BCIs). The authors claimed achieving state-of-the-art for the performance in steady-state visual evoked potential (SSVEP) based BCIs but their method suffered from a data leakage issue during implementation because the authors didn’t use train-test split for model evaluation. Addressing

this problem helps beginners learn the importance of splitting data into two mutually exclusive sets- one for train and other for test.

The study by El-Sayed et al. (2017) delves into the issue of interruptions and premature termination of jobs within large-scale computing platforms. Reproducing this study will offer valuable insights for introductory machine learning students, particularly regarding the temporal aspects of data. Specifically, it will highlight challenges related to data splitting in AIOps modeling and demonstrate the importance of using Time based-splitting methods to prepare train and test sets based on the specific use case.

Expected Deliverables- A Trovi artifact including all the packages which will contain all the notebooks mentioned below.

Paper 1 (Khan et al. (2019))- The Term-Preterm EHG Database contains 300 uterine EMG records for 300 pregnancies since these records cannot be directly used for classification tasks. (github)

A. Notebook 1- To perform the required preprocessing - Empirical Mode Decomposition (EMD) will be performed to extract Intrinsic Mode Functions (IMFs) and first IMF will be selected for feature extraction.
B. Notebook 2- To access extracted features and here we'll use ADASYN as our over-sampling technique to balance the number of classes in the dataset and then classification will be performed using SVM.
C. Notebook 3- To perform over-sampling in the correct manner and to show the results without the data leakage. Using the same code for SVM from the last notebook.
D. Notebook 4- To implement a toy example. The code to generate random artificial data is already provided on the github repository. Code for SVM will be reused from before and code for comparing results for both the cases (with leakage and without leakage) will be written from scratch.
E. Notebook 5- Questions to assess the ability of students about this particular data leakage.

Paper 2 (Wollenstein-Betech et al. (2020)) - (github)

A. Notebook 1- To download the data and perform necessary preprocessing as mentioned in the original paper- outlier removal, one-hot encoding, and removing correlated variables.
B. Notebook 2- To reproduce the results with the data leakage. Code for spare LR and Decision trees will be written from scratch.
C. Notebook 3- To re-implement the method as mentioned in the paper but without data leakage. Model codes will be reused here but preprocessing code will be written again to avoid data leakage
D. Notebook 4- To generate simulated data and explain the leakage with simple data. Model code reused again, preprocessing code will be used from notebook 1 and 3.
E. Notebook 5- Contains questions related to the data leakage explained in this example.

Paper 3 (Kumar and Reddy (2019)) - (github)

A. Notebook 1- To import data and perform necessary preprocessing as mentioned in the paper- extraction of EEG data in a given interval, filtering etc. MATLAB code will be converted to python code for preprocessing.
B. Notebook 2- To reproduce the results with the data leakage. Model code will be converted and used to reproduce the original results.
C. Notebook 3- To re-implement the paper without the data leakage. Model code from notebook 2 and preprocessing code from notebook 1 will be reused but with a proper train-test split.
D. Notebook 4- Toy example will be implemented from scratch (explained in the next section).
E. Notebook 5- To provide questions related to this particular data leakage.

Paper 4 (El-Sayed et al. (2017))- This paper uses three publicly available datasets. All the code will be written from scratch- Preprocessing, visualizations, random forest model etc.

A. Notebook 1- To show examples of exploratory data analysis to understand the dataset used in the paper and to perform necessary preprocessing.
B. Notebook 2- To reproduce the original results with the data leakage.
C. Notebook 3- To produce the actual results without the data leakage.
D. Notebook 4- To implement a toy example to explain the data leakage issue more clearly using simple time series datasets like- The Covid-19 Dataset from Kaggle.
E. Notebook 5- Questions to test the students about this particular data leakage.

Implementation Plan

Engagement with mentors and peers within the open-source community will be dynamic and collaborative. I will attend weekly meetings where I will succinctly present my project advancements, creating a platform for insightful feedback and discussion among other participants. Furthermore, our interactions will extend beyond meetings through active participation in GitHub discussions, leveraging issues and comments to address queries and facilitate ongoing collaboration.

Paper 1 (Khan et al. (2019))- (github)

A. Dataset- https://physionet.org/content/tpehgdb/1.0.1/ (245 MB)
B. Toy example- Code provided in the github repository to generate a simulated dataset.
C. Data preprocessing and Model- Code is provided in the github repository to reproduce the results.

Paper 2 (Wollenstein-Betech et al. (2020)) - (github)

A. Dataset- Data is already present in the github repository.
B. Toy example- Use a simulated dataset to show the effect of non-independence of train-test sets on the prediction result. Generate data using numpy in which the output variable depends on the input variable and we can train a model on this dataset to show both skewed and non-skewed results. (Results will be skewed since both sets are not independent)
C. Data preprocessing and Model- Code is provided in the github repository to reproduce the results.

Paper 3 (Kumar and Reddy (2019)) - (github)

A. Dataset- http://bci.med.tsinghua.edu.cn/download.html (approximately 3.5 GB)
B. Toy example- A simple dataset (sample.mat in the github repository) is provided with reduced features. This dataset can be used to create a toy example to provide an easier way of understanding

the problem or we can train any binary classification model on a simple datasets like Titanic dataset from Kaggle to show the importance of train-test split.

C. Data preprocessing and Model- Code is provided in the github repository. All the algorithms are written in the MATLAB but it can be easily converted into python.

Paper 4 (El-Sayed et al. (2017))-

A. Dataset- https://github.com/google/cluster-data (41 GB), gsutil command-line tool to download the data.

B. Data preprocessing and Model- The data preprocessing step is explained in a replication paper by Lyu et al. (2021b). Hyperparameter settings for the Random Forest model are provided in both the original paper and the replication paper.

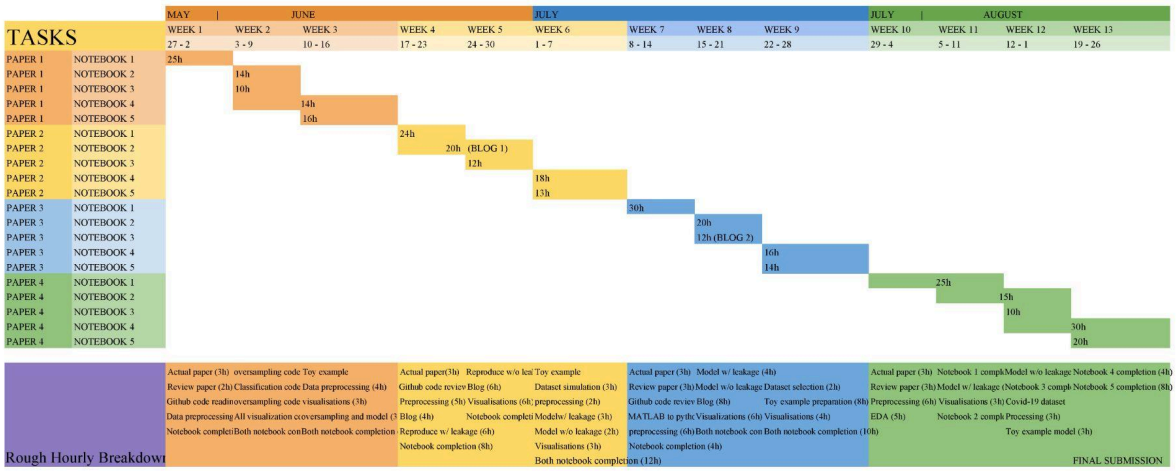
C. Toy examples- I'll be using a simple time-series dataset like Covid-19 dataset available on Kaggle to create a simpler replication of the problem about temporal splitting of data.

D. A Chameleon test bed is essential here due to the dataset's size and the computational demands of data preprocessing and model training, which could overwhelm local machines.

All the necessary libraries are mentioned here as well- numpy, wfdb, pandas, scipy, PyWavelets, neurokit, PyEMD, entropy, tsfresh, smote_variants, matplotlib, seaborn, tensorflow, xgboost, Pytorch.

Project Timeline

To access the weekly and hourly breakdown of the project implementation - [LINK]



Professional biography

Name - Satvik

Email - f20213047@goa.bits-pilani.ac.in

Affiliation - Birla Institute of Technology and Science, Pilani – Goa

Github - https://github.com/Satvik713

CV - Link

In my junior year pursuing a Bachelor's degree in Electronics Engineering at BITS Pilani, Goa campus, I've devoted the past two years to studying machine learning intensively. Through coursework such as "Machine Learning (BITS F464)" and online courses like Andrew Ng's "Supervised Machine Learning: Regression and Classification" and "Neural Networks and Deep Learning", I've built a solid theoretical foundation and practical skills. Completion of the machine learning course at my college has honed my proficiency in implementing basic algorithms like Random Forests, Support Vector Machines, and Logistic Regression. My CV, enclosed herewith, provides a comprehensive overview of my skills, coursework, projects, and past experiences.

References

1. Khan, M. U., Aziz, S., Ibraheem, S., Butt, A., & Shahid, H. (2019). Characterization of Term and Preterm Deliveries using Electrohysterograms Signatures. *IEEE*. https://doi.org/10.1109/iemcon.2019.8936292
2. Wollenstein-Betech, S., Cassandras, C. G., & Paschalidis, I. C. (2020). Personalized predictive models for symptomatic COVID-19 patients using basic preconditions: Hospitalizations, mortality, and the need for an ICU or ventilator. *International Journal of Medical Informatics*, *142*, 104258. https://doi.org/10.1016/j.ijmedinf.2020.104258
3. El-Sayed, N., Zhu, H., & Schroeder, B. (2017). Learning from Failure Across Multiple Clusters: A Trace-Driven Approach to Understanding, Predicting, and Mitigating Job Terminations. *IEEE*. https://doi.org/10.1109/icdcs.2017.317
4. Kumar, G. R. K., & Reddy, M. R. (2019). Designing a sum of squared correlations framework for enhancing SSVEP-Based BCIs. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *27*(10), 2044–2050. https://doi.org/10.1109/tnsre.2019.2941349
5. Lyu, Y., Li, H., Sayagh, M., Jiang, Z. M., & Hassan, A. E. (2021b). An empirical study of the impact of data splitting decisions on the performance of AIOPs solutions. *ACM Transactions on Software Engineering and Methodology*, *30*(4), 1–38. https://doi.org/10.1145/3447876