Supplementary Materials: Causal Machine Learning Framework for Early Detection of Long COVID

Sindy Licette Pinero
pinsy007@mymail.unisa.edu.au
University of South Australia
Adelaide, SA, Australia

Thi Nhu Ngoc Duong
duoty016@mymail.unisa.edu.au
University of South Australia
Adelaide, SA, Australia

Xiaomei Li
maisie.zhang@csiro.au
Commonwealth Scientific and Industrial Research Organisation
Marsfield, NSW, Australia

Lin Liu
lin.liu@unisa.edu.au
University of South Australia
Adelaide, SA, Australia

Jiuyong Li
jiuyong.li@unisa.edu.au
University of South Australia
Adelaide, SA, Australia

Sang Hong Lee
hong.lee@unisa.edu.au
University of South Australia
Adelaide, SA, Australia

Thuc Duy Le
thuc.le@unisa.edu.au
University of South Australia
Adelaide, SA, Australia

CCS Concepts

• **Applied computing → Health informatics**; *Bioinformatics*; • **Computing methodologies → Machine learning approaches**; *Causal reasoning and diagnostics.*

Keywords

Long COVID, PASC, early detection, causal inference, differential causal effects, TabPFN, gene expression, machine learning, foundation models, biomarkers

ACM Reference Format:
Sindy Licette Pinero, Thi Nhu Ngoc Duong, Xiaomei Li, Lin Liu, Jiuyong Li, Sang Hong Lee, and Thuc Duy Le. 2025. Supplementary Materials: Causal Machine Learning Framework for Early Detection of Long COVID. In *Proceedings of Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Classification Metrics Definitions

1.1 Classification Metrics Terminology

The following terminology is used throughout this study for binary classification evaluation:

- **True Positives (TP):** Correctly predicted Long COVID cases
- **True Negatives (TN):** Correctly predicted non-Long COVID cases
- **False Positives (FP):** Incorrectly predicted Long COVID cases (Type I error)
- **False Negatives (FN):** Missed Long COVID cases (Type II error)

1.2 Performance Metrics

1.2.1 Accuracy.

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

Represents the overall proportion of correct predictions across all classes.

1.2.2 Precision (Positive Predictive Value).

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

Measures the proportion of predicted Long COVID cases that are actually positive, indicating model reliability for positive predictions.

1.2.3 Recall (Sensitivity, True Positive Rate).

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

Quantifies the proportion of actual Long COVID cases correctly identified, crucial for clinical applications where missing cases has serious consequences.

1.2.4 F1-Score.

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

Provides the harmonic mean of precision and recall, balancing both metrics for overall model performance assessment [19].

CIKM '25, Seoul, Republic of Korea

ACM ISBN 978-1-4503-XXXX-X/2025/11
https://doi.org/XXXXXXX.XXXXXXX

1.2.5 ROC AUC (Area Under the Receiver Operating Characteristic Curve).

$$\text{ROC AUC} = \int_0^1 \text{TPR}(FPR^{-1}(t))dt \tag{5}$$

where TPR = $\frac{TP}{TP+FN}$ (True Positive Rate) and FPR = $\frac{FP}{FP+TN}$ (False Positive Rate).

The ROC AUC measures the model's ability to distinguish between classes across all classification thresholds, with values ranging from 0.5 (random) to 1.0 (perfect) [6].

2 Machine Learning Models: Equations, Purpose and Rationale

2.1 Linear Models

2.1.1 Logistic Regression.

Equation:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \sum_{i=1}^{p} \beta_i x_i)}} \tag{6}$$

where β_0 is the intercept, β_i are the coefficients for features x_i, and p is the number of features.

Purpose: Serves as a fundamental baseline for binary classification, providing interpretable coefficients that indicate the linear relationship between gene expression features and Long COVID risk.

Rationale: Linear models are essential benchmarks in genomics studies due to their interpretability and ability to handle high-dimensional data [11]. They provide insights into whether the relationship between gene expression and Long COVID can be captured through linear combinations.

2.1.2 Ridge Classifier.

Equation:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{7}$$

where λ is the regularization parameter controlling the strength of L2 penalty.

Purpose: Addresses multicollinearity in gene expression data through L2 regularization, preventing overfitting when dealing with correlated genetic features.

Rationale: Gene expression data often exhibits high correlation between related genes. Ridge regression maintains all features while shrinking coefficients, making it suitable for scenarios where multiple genes may collectively contribute to the outcome [12].

2.1.3 Linear Discriminant Analysis (LDA).

Equation:

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k \tag{8}$$

where μ_k is the class mean, Σ is the shared covariance matrix, and π_k is the prior probability.

Purpose: Assumes Gaussian distributions for each class and finds linear combinations of features that best separate Long COVID from non-Long COVID cases.

Rationale: LDA provides dimension reduction capabilities while maintaining discriminative power, potentially useful when gene expression patterns follow multivariate normal distributions within each class [7].

2.2 Instance-Based Models

2.2.1 k-Nearest Neighbors (KNN).

Equation:

$$\hat{y} = \text{mode}\{y_i : x_i \in N_k(x)\} \tag{9}$$

where $N_k(x)$ represents the k nearest neighbors of point x based on distance metric $d(x_i, x_j)$.

Purpose: Makes predictions based on similarity to k nearest samples in the feature space, capturing local patterns in gene expression profiles.

Rationale: Genetic similarity often translates to phenotypic similarity. KNN can identify patients with similar gene expression patterns who may share Long COVID outcomes, providing a non-parametric approach to classification [5].

2.3 Neural Network Models

2.3.1 Multi-Layer Perceptron (MLP).

Equation:

$$h^{(l)} = \sigma(W^{(l)} h^{(l-1)} + b^{(l)}) \tag{10}$$

$$\hat{y} = \text{softmax}(W^{(L)} h^{(L-1)} + b^{(L)}) \tag{11}$$

where $h^{(l)}$ is the l-th hidden layer, $W^{(l)}$ and $b^{(l)}$ are weights and biases, and σ is the activation function.

Purpose: Captures non-linear relationships between gene expression features through hidden layers, potentially identifying complex gene interaction patterns.

Rationale: Biological systems exhibit complex non-linear interactions. MLP can model these relationships without requiring explicit specification of interaction terms, potentially uncovering hidden patterns in gene expression data [18].

2.3.2 Support Vector Classifier (SVC).

Equation:

$$\min_{w,b,\xi} \frac{1}{2} ||w||^2 + C \sum_{i=1}^{n} \xi_i \tag{12}$$

subject to: $y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$

where w is the weight vector, b is the bias, ξ_i are slack variables, C is the regularization parameter, and $\phi(x_i)$ maps input to higher-dimensional space.

Purpose: Finds optimal hyperplanes that separate Long COVID cases with maximum margin, handling non-linear relationships through kernel transformations.

Rationale: SVC is particularly effective for high-dimensional data like gene expression, where the number of features exceeds the

number of samples. The kernel trick allows modeling of complex decision boundaries [4].

2.4 Tree-Based Models

2.4.1 Random Forest.

Equation:

$$\hat{y} = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \tag{13}$$

where each tree T_b is trained on a bootstrap sample $\mathcal{D}_b$ with random feature selection at each split.

Purpose: Combines multiple decision trees with bootstrap sampling and random feature selection to reduce overfitting while maintaining predictive power.

Rationale: Tree-based methods naturally handle feature interactions and non-linearities common in biological data. Random Forest provides built-in feature importance measures and handles mixed data types effectively [2].

2.4.2 Extra Trees (Extremely Randomized Trees).

Equation:

$$\hat{y} = \frac{1}{M}\sum_{m=1}^{M} T_m(x, \Theta_m) \tag{14}$$

where Θ_m represents random parameters for both feature and threshold selection in tree T_m.

Purpose: Further randomizes the tree construction process by selecting split points randomly, reducing variance compared to Random Forest.

Rationale: Extra randomization can improve generalization, particularly important when dealing with noisy gene expression data where overfitting to specific expression patterns is a concern [10].

2.4.3 Bagging Classifier.

Equation:

$$\hat{y} = \text{mode}\{\hat{f}_1(x), \hat{f}_2(x), ..., \hat{f}_B(x)\} \tag{15}$$

where each $\hat{f}_b$ is trained on bootstrap sample $\mathcal{D}_b^* \sim \mathcal{D}$.

Purpose: Implements bootstrap aggregating by training multiple decision trees on different bootstrap samples of the training data and averaging their predictions.

Rationale: Bagging reduces model variance by combining predictions from multiple trees trained on different data subsets. This approach is particularly valuable for gene expression data where individual trees might overfit to specific patterns [1].

2.4.4 Gradient Boosting.

Equation:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{16}$$

$$h_m = \arg\min_h \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + h(x_i)) \tag{17}$$

where h_m is the m-th weak learner and γ_m is the step size.

Purpose: Sequentially builds weak learners that correct errors from previous iterations, focusing on difficult-to-classify cases.

Rationale: Boosting methods excel at reducing bias and can identify subtle patterns in data. This is valuable for Long COVID prediction where the biological mechanisms may involve complex, interconnected pathways [9].

2.4.5 AdaBoost.

Equation:

$$\alpha_m = \frac{1}{2}\ln\left(\frac{1-\epsilon_m}{\epsilon_m}\right) \tag{18}$$

$$w_i^{(m+1)} = w_i^{(m)} \exp(-\alpha_m y_i h_m(x_i)) \tag{19}$$

where ϵ_m is the weighted error and w_i are sample weights.

Purpose: Adaptively adjusts the importance of training samples, emphasizing misclassified cases in subsequent iterations.

Rationale: AdaBoost can effectively handle class imbalance issues that may exist in Long COVID datasets, ensuring that minority class patterns are adequately learned [8].

2.5 Advanced Ensemble Methods

2.5.1 XGBoost.

Equation:

$$\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{20}$$

where $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda||\omega||^2$ is the regularization term.

Purpose: Implements optimized gradient boosting with advanced regularization techniques and efficient computation.

Rationale: XGBoost has demonstrated superior performance in many genomics applications due to its ability to handle missing values, regularization capabilities, and optimization for predictive accuracy [3].

2.5.2 LightGBM.

Equation:

$$\text{Gain} = \frac{1}{2}\left[\frac{(\sum g_L)^2}{H_L+\lambda} + \frac{(\sum g_R)^2}{H_R+\lambda} - \frac{(\sum g)^2}{H+\lambda}\right] - \gamma \tag{21}$$

where g and H are gradient and hessian statistics, using leaf-wise tree growth.

Purpose: Provides fast gradient boosting with leaf-wise tree growth and categorical feature optimization.

Rationale: LightGBM's efficiency makes it suitable for large-scale genomics data while maintaining competitive predictive performance through its optimized algorithms [14].

2.5.3 CatBoost.

Equation:

$$\hat{y}_i^{(k)} = \frac{\sum_{j=1}^{i-1} \mathbb{I}[x_j^{(k)} = x_i^{(k)}] \cdot y_j + \alpha}{\sum_{j=1}^{i-1} \mathbb{I}[x_j^{(k)} = x_i^{(k)}] + 1} \tag{22}$$

implementing ordered target statistics for categorical features.

Purpose: Handles categorical features natively and provides robust performance with minimal hyperparameter tuning.

Rationale: CatBoost's automatic handling of categorical features and built-in regularization makes it valuable for mixed data types common in clinical genomics studies [17].

2.6 Meta-Ensemble Approaches

2.6.1 *Voting Classifier.*

Equation:

$$\hat{P}(y = c|x) = \frac{1}{M} \sum_{m=1}^{M} P_m(y = c|x) \tag{23}$$

where $P_m(y = c|x)$ is the predicted probability from the m-th base classifier.

Purpose: Combines predictions from multiple diverse algorithms through soft voting, leveraging the strengths of different modeling approaches.

Rationale: Different algorithms capture different aspects of the data. Voting classifiers can provide more robust predictions by aggregating diverse perspectives on the gene expression patterns [16].

2.6.2 *Stacking Classifier.*

Equation:

$$\hat{y}_i^{(1)} = f_m(x_i^{train \setminus \mathcal{K}_k}) \text{ for } i \in \mathcal{K}_k \tag{24}$$

$$\hat{y} = g(\hat{y}_1^{(1)}, \hat{y}_2^{(1)}, ..., \hat{y}_M^{(1)}) \tag{25}$$

where g is the meta-learner trained on base model predictions.

Purpose: Uses a meta-learner to optimally combine base model predictions, learning the best way to weight different algorithms.

Rationale: Stacking can achieve superior performance by learning how to combine models rather than using simple averaging, potentially identifying which models perform best for different types of gene expression patterns [20].

2.7 Foundation Model

2.7.1 *TabPFN (Tabular Prior-Fitted Networks).*

Equation:

$$P(y_{test}|x_{test}, D_{train}) = \text{Transformer}(x_{test}, D_{train}) \tag{26}$$

where $D_{train} = \{(x_i, y_i)\}_{i=1}^{n}$ is the training set and the model learns from context without parameter updates.

Purpose: Employs transformer architecture with in-context learning, leveraging causal reasoning principles embedded in its pre-training process.

Rationale: TabPFN represents a paradigm shift from traditional ML by learning from context rather than parameter optimization. Its causal reasoning capabilities align with our DCE-based feature selection, creating a synergistic framework for causal inference in Long COVID prediction [13].

3 Detailed Model Hyperparameters

3.1 Default Parameter Strategy

Purpose: Using default hyperparameters ensures fair comparison across all models without introducing optimization bias toward any particular algorithm.

Rationale: Our study focuses on evaluating the effectiveness of DCE-based feature selection rather than individual model optimization. Default parameters provide a standardized baseline that reflects how these algorithms perform in typical applications.

3.2 Model-Specific Hyperparameters

3.2.1 *TabPFN.*

- Device: CPU
- N_ensemble_configurations: 32 (default)
- No_preprocess_mode: False (default)

3.2.2 *Linear Models.*

- **Logistic Regression:** max_iter=1000, C=1.0, penalty='l2', solver='lbfgs'
- **Ridge Classifier:** alpha=1.0, solver='auto'
- **Linear Discriminant Analysis:** solver='svd', shrinkage=None

3.2.3 *Tree-Based Models.*

- **Random Forest:** n_estimators=100, max_depth=None, min_samples_split=
- **Extra Trees:** n_estimators=100, max_depth=None, min_samples_split=2
- **Bagging Classifier:** n_estimators=10, max_samples=1.0, max_features=1.0
- **Gradient Boosting:** n_estimators=100, learning_rate=0.1, max_depth=3
- **AdaBoost:** n_estimators=50, learning_rate=1.0

3.2.4 *Advanced Ensemble Methods.*

- **XGBoost:** use_label_encoder=False, eval_metric='logloss', n_estimators=100
- **LightGBM:** verbosity=-1, n_estimators=100, learning_rate=0.1
- **CatBoost:** verbose=0, iterations=1000, learning_rate=0.03

3.2.5 *Neural Networks.*

- **Multi-Layer Perceptron:** max_iter=1000, hidden_layer_sizes=(100,), alpha=0.0001
- **Support Vector Classifier:** C=1.0, kernel='rbf', gamma='scale', probability=True

3.2.6 *Instance-Based Models.*

- **k-Nearest Neighbors:** n_neighbors=5, weights='uniform', algorithm='auto'

3.2.7 *Meta-Ensemble Models.*

- **Voting Classifier:** voting='soft', estimators=[LogisticRegression, RandomForest, XGBoost]
- **Stacking Classifier:** final_estimator=LogisticRegression(), cv=5

4 Cross-Validation and Statistical Analysis

4.1 Stratified K-Fold Cross-Validation

Purpose: Ensures balanced representation of Long COVID cases across all folds while providing robust performance estimates.

Rationale: Stratification is crucial for medical datasets where class imbalance may exist. This approach ensures that each fold contains representative samples from both Long COVID and control groups, providing more reliable performance estimates [15].

4.2 Performance Aggregation and Reporting

Purpose: Provides robust estimates of model performance by aggregating results across multiple cross-validation folds, quantifying both central tendency and variability.

Rationale: Standard deviation indicates model stability and reliability. Models with high mean performance but high variability may be less trustworthy for clinical applications than models with moderate performance but consistent behavior.

5 Code Availability

The complete implementation of this framework is available at: https://github.com/SindyPin/Early_Detection_LongCOVID

The repository includes comprehensive documentation, implementation scripts, and reproducibility guidelines for the DCE-TabPFN framework.

References

[1] Leo Breiman. 1996. Bagging predictors. *Machine Learning*, 24, 2, 123–140. doi:10.1007/BF00058655.

[2] Leo Breiman. 2001. Random forests. *Machine Learning*, 45, 1, 5–32. doi:10.1023/A:1010933404324.

[3] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: a scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. doi:10.1145/2939672.2939785.

[4] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20, 3, 273–297. doi:10.1007/BF00994018.

[5] Thomas Cover and Peter Hart. 1967. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13, 1, 21–27. doi:10.1109/TIT.1967.1053964.

[6] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern Recognition Letters*, 27, 8, 861–874. doi:10.1016/j.patrec.2005.10.010.

[7] Ronald A. Fisher. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7, 2, 179–188. doi:10.1111/j.1469-1809.1936.tb02137.x.

[8] Yoav Freund and Robert E. Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55, 1, 119–139. doi:10.1006/jcss.1997.1504.

[9] Jerome H. Friedman. 2001. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, 29, 5, 1189–1232. doi:10.1214/aos/1013203451.

[10] Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine Learning*, 63, 1, 3–42. doi:10.1007/s10994-006-6226-1.

[11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. (2nd ed.). Springer, New York, NY. doi:10.1007/978-0-387-84858-7.

[12] Arthur E. Hoerl and Robert W. Kennard. 1970. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12, 1, 55–67. doi:10.1080/00401706.1970.10488634.

[13] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. 2023. TabPFN: a transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=cp5PvcI6w8_.

[14] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: a highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*. Vol. 30, 3146–3154. https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[15] Ron Kohavi. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*. Vol. 2. Morgan Kaufmann Publishers Inc., San Francisco, CA, 1137–1143.

[16] Ludmila I. Kuncheva. 2004. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons, Hoboken, NJ. doi:10.1002/0471660264.

[17] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. CatBoost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*. Vol. 31, 6638–6648. https://proceedings.neurips.cc/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf.

[18] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. *Learning representations by back-propagating errors*. Vol. 323, 533–536. doi:10.1038/323533a0.

[19] Marina Sokolova and Guy Lapalme. 2009. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45, 4, 427–437. doi:10.1016/j.ipm.2009.03.002.

[20] David H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5, 2, 241–259. doi:10.1016/S0893-6080(05)80023-1.