Supplementary Data

1. Summary of Models

Artificial Neural Network (ANN)

An ANN is a multilayer feedforward network that can model complex, nonlinear relationships. It consists of an input layer, one or more hidden layers, and an output layer. Each neuron in a layer takes in inputs, applies a weighted sum, adds a bias, and passes the result through an activation function (such as ReLU or tanh) to introduce nonlinearity.

Each layer ℓ performs the transformation:

$$z^{(\ell)} = W^{(\ell)}a^{(\ell-1)} + b^{(\ell)}, a^{(\ell)} = \emptyset\left(z^{(\ell)}\right)$$

For binary classification, the final output is:

$$\hat{y} = \sigma(w^{(L)^T}a^{(L-1)} + b^{(L)})$$

ANNs are trained using backpropagation, which computes how each weight contributes to prediction error, and an optimiser like stochastic gradient descent (SGD) updates weights to reduce that error. They are powerful but require hyperparameter tuning and lots of data to perform well without overfitting. ANN models may suffer from interpretability and hyperparameter issues (layers, neurons, activation functions, learning rate, batch size). Challenges like unit saturation (e.g., hidden activations near ±1) may hinder learning, and explainability tools (e.g., SHAP, weight visualisation) are often required to interpret predictions.

Convolutional Neural Network (CNN)

A CNN is a deep learning model good at recognising patterns in images, such as shapes or textures. It does so by scanning them with small filters. Although it automatically learns to detect features, the internal decision making process is often considered a "black box," meaning it can be difficult to interpret how exactly the model derives its conclusions (1).

A CNN model uses layers of filters (kernels) that perform convolutions, sliding window operations, to extract spatial features such as edges, textures, or structures. Each convolutional layer is followed by a nonlinear activation function (e.g., ReLU) and often a pooling layer that reduces the spatial resolution

while preserving key features. The deeper layers detect increasingly complex patterns, and the final fully connected layers make predictions, often using a softmax or sigmoid function.

The above can be expressed mathematically, using a series of convolutional operations, where each filter $K \in \mathbb{R}^{k_h \times k_w}$ is convolved with a local region of the input $X \in \mathbb{R}^{H \times W}$ to produce a feature map Y, according to:

$$Y_{i,j} = \sum_{u=0}^{k_h-1} \sum_{v=0}^{k_w-1} X_{i+u,j+v} K_{u,v} + b$$

This is followed by nonlinear activation (commonly ReLU: $ReLU(z) = \max(0, z)$) and often down sampling via max-pooling to build translation invariant features. In other words, ReLU, outputs the input if it's positive, or zero if it's negative. Deeper layers capture increasingly abstract representations (e.g., textures, shapes, pathological features).

CNN excels at medical image classification tasks. In the context of venous thrombus embolism (VTE), they have been explored for their utility in thrombi imaging for both ultrasound and computed tomography pulmonary angiography. However, they demand high computational resources and large annotated datasets to prevent overfitting of the derived model.

Gradient Boosting (XGBoost)

XGBoost is a machine learning technique that builds a model by combining many simple models one at a time. Each new tree learns from the mistakes made by the previous ones to improve accuracy at each step.

At each step, the models fits a tree to the current residuals (errors) and adds it to the ensemble with a small weight (learning rate υ)

$$F_m(x) = F_{m-1}(x) + \nu \times h_m(x)$$

Gradient boosting can model complex, non-linear relationships and highlight which features are most important. this is acheived by using second-order derivatives for better precision and regularisation to avoid overfitting. At each step, the model fits a tree to the current residuals (errors) and adds it to the ensemble with a small weight (learning rate ν).

Least Absolute Shrinkage and Selection Operator (LASSO) Regression

LASSO Regression is a form of regularised regression that not only fits a predictive model but also performs feature selection. It does this by adding a penalty to the size of the coefficients, where larger coefficients are penalised more heavily.

For linear regression, LASSO solves:

$$\min_{\beta}\left\{\frac{1}{2N}\sum_{i=1}^{N}(y_i-\beta_0-x_i^T\beta)^2+\lambda\sum_{j=1}^{p}\left|\beta_j\right|\right\}$$

In classification, it replaces the squared error with log-basis. The regularisation term $\lambda\sum_{j=1}^{p}\left|\beta_j\right|$ encourages sparsity, which means that many coefficients shrink to zero and the model only keep the most useful predictors. This is useful in high-dimensional data where p ≫ N.

LASSO is highly effective in high-dimensional settings (e.g., genomic or imaging data) and enhances interpretability by removing irrelevant variables. However, it assumes linear or log-linear relationships, struggles with collinear features (tending to keep only one), and may require feature engineering to capture interactions. The value of λ is typically selected via cross-validation to balance bias and variance.

Linear Discriminant Analysis (LDA)

LDA is a classification algorithm that assumes each class has data points that are normally distributed, with their own mean but a shared variance. It calculates the best linear boundary (a discriminant function) that separates classes by maximising the difference between group means while minimising variation within each group.

It calculates a score, called the discriminant function, for each data point using the formula:

$$\delta(x)=x^T\sum^{-1}(\mu_1-\mu_0)\ -\frac{1}{2}\left(\mu_1^T\sum^{-1}\mu_1-\mu_0^T\sum^{-1}\mu_o\right)+\log\left(\frac{\pi_1}{\pi_0}\right)$$

This function combines how close the input *x* is to each class mean (μ_k) and how likely each class is overall (π_k). A new observation is classifies as class 1 if $\delta(x) > 0$; otherwise, it is assigned class 0. This means (μ_0, μ_1) and the shared covariance $\sum$ are estimated from the training data.

LDA is computationally efficient, yields closed-form solutions, and performs well when its assumptions hold. However, if classes have unequal covariances or the feature distributions are not approximately normal, performance may degrade. Moreover, LDA's linear boundary limits its flexibility in capturing nonlinear class separations, unless extended via kernel methods or nonlinear feature engineering.

Logistic Regression

Logistic regression is a statistical model used for binary classification. It estimates the probability of an event (e.g. disease = Yes or No) by applying a sigmoid function to a linear combination of input variables. Mathematically, this is presented as:

$$P(Y = 1 \mid x) = \frac{1}{1 + e^{-(\beta_0 + \sum_{j=1}^{p} \beta_j x_j)}}$$

The model learns the best parameters $\beta_j$by maximising the likelihood that the observed outcomes match the predicted ones. It is widely used in medical risk prediction due to its simplicity, interpretability, and computationally efficiency.

However, it assumes linear effects on the log-odds scale and does not capture nonlinear interactions unless explicitly added (e.g. polynomials, interaction terms). Thus, while interpretable, it may underperform compared to nonlinear models in complex clinical settings.

Polygenic Risk Score (PRS) Model

A PRS is a genetic risk model that combines the effects of many single nucleotide polymorphisms (SNPs), each contributing a small amount to disease risk. For each SNP *j*, the individual's genotype $G_j \epsilon \{0,1,2\}$ (representing the number of risk alleles) is multiplied by a known log-odds ratio B_j, and summed over all M SNPs:

$$PRS = \sum_{j=1}^{M} \beta_j G_j$$

PRSs are particularly effective when risk is polygenic, enabling stratification of individuals at the tails of genetic risk distribution. However, limitations include reduced portability across ancestries, exclusion of environmental or clinical covariates, and susceptibility to bias in effect size estimation. Their optimal use lies in integrated models combining genetic and non-genetic predictors to improve clinical utility.

Random Forest (RF)

A RF is an ensemble classifier made up of many decision trees, each trained on a random sample (with replacement) from the dataset (called a bootstrap sample). At each decision point (node) in a tree, only a random subset of features is considered for splitting, which helps reduce overfitting and increase diversity among trees.

Each tree makes its own prediction, and the forest combines these (e.g., by majority vote for classification). To decide on a split at each node, the algorithm chooses the feature and threshold that best reduces Gini impurity:

$$Gini(m) = 1 - \sum_{k=0}^{1} p_{m,k}^2$$

Where $p_{m,k}^2$ is the proportion of observations of class *k* in node *m*. This measures how "pure" a node is (the lower, the better). Random forests are robust, can work with missing data, and provide measures of feature importance.

Supervised Predictive Modelling (Bleeding Risk Analysis)

Supervised modelling involves training algorithms (like logistic regression, random forests, or support vector machines) on labelled datasets where the input features x and outcome y (e.g. bleeding event) are known. The model learns to minimise a loss function (e.g., cross-entropy, mean squared error) that measures how wrong its predictions are.

Unsupervised Clustering (Bleeding Risk Analysis)

Unsupervised clustering groups patients based on similarities in their features, without using any outcome labels (like whether they had a bleeding event). Techniques such as K-means clustering or Gaussian Mixture Models (GMMs) are commonly used. Clustering is exploratory; it does not predict risk directly, but it can reveal hidden subgroups that may later inform clinical decisions.

2. Summary of Integration of Explainability Methods

SHAP (Shapley Additive Explanations) and LIME (Local Interpretable Model-agnostic Explanations) are post hoc interpretability tools used to explain predictions from complex models such as random forests, XGBoost, or neural networks.

SHAP is a model-agnostic method for interpreting machine learning predictions by quantifying the contribution of each feature to a specific prediction. Based on cooperative game theory, SHAP assigns each feature a "Shapley value" that reflects its average impact on the model's output across all possible feature combinations. This approach (1) provides both global and local interpretability, (2) is applicable to complex models like random forests and deep neural networks, and (3) supports visualisation of how individual features drive predictions. This makes it especially useful for uncovering insights in "black box" models (2).

LIME creates a simple model (like linear regression) around a single prediction by perturbing the inputs slightly. This local model approximates how the complex model behaves in that region of the input space.

For example, in a bleeding risk model, SHAP or LIME can tell a clinician that age and D-dimer were the main reasons a patient was flagged as high-risk.

3. Summary of Performance Metrics

Brier Score

The Brier score evaluates the accuracy of a model's probabilistic predictions by comparing each predicted probability with the actual outcome. This is mathematically expressed as:

$$(\text{predicted probability of VTE} - \text{actual outcome})^2$$

where the actual outcome is 1 if the patient had VTE and 0 otherwise, then average these squared errors across all patients. The resulting value lies between 0 and 1: a score of 0 indicates perfect prediction (all probabilities exactly match the outcomes), while a score of 1 would mean maximal error (for example, predicting 100% risk for every patient who never experienced the event). Because it incorporates both calibration (whether predicted probabilities match observed frequencies) and discrimination (separating high- versus low-risk patients), the Brier score provides a single summary of overall model performance. Lower Brier scores therefore reflect better-calibrated and more accurate probability estimates.

C-statistic

The C-statistic quantifies a model's ability to discriminate between patients who experience an event and those who do not. It is equivalent to the area under the Receiver Operating Characteristic (ROC) curve (AUC). It can interpreted as the probability that, if one randomly select one patient who developed VTE and one who did not, the model will assign a higher predicted risk to the patient who actually had VTE. Values range from 0.5 (no discriminative ability) up to 1.0 (perfect discrimination). In practice, a C-statistic above 0.7 is considered acceptable. Above 0.8 is considered good. And above 0.9 excellent. Because it focuses solely on ranking rather than the exact predicted probability values, the C-statistic complements the Brier score by isolating how well a model orders patients by risk, regardless of calibration.

Hosmer-Lemeshow

The Hosmer-Lemeshow test is a used to assess how well a model's predicted probabilities agree with observed outcomes across the spectrum of risk. To perform the test, patients are typically sorted by their predicted risk and divided into multiple equally sized groups (deciles). For each group, one would compare the number of events observed (in this context, VTE cases) to the number one would expect based on the model's average predicted probability. Summing the squared differences between observed and expected counts (weighted by the expected variance) yields a chi-square statistic. A corresponding p-value is then calculated: a large p-value (> 0.05) suggests no significant discrepancy between predicted and actual event rates, indicating good overall calibration.

While the Hosmer-Lemeshow test offers a simple, interpretable check on model calibration, it has important caveats. Its result can vary depending on how you choose the number and size of risk groups, and with very large datasets even small, clinically irrelevant miscalibrations can produce a statistically significant (low) p-value. Conversely, in small samples it may lack power to detect poor fit. Therefore, it is best used alongside calibration plots (visual graphs of observed versus predicted event rates) and other metrics (such as the Brier score) to form a complete picture of a model's reliability.

Receiving Operator Characteristics-Area Under the Curve

ROC analysis evaluates how well an index test performs. The AUC represents a single summary measure that quantifies the test's ability to differentiate between individuals with and without the condition. AUC values are from 0.5 (indicating no diagnostic ability beyond random chance) to 1.0 (indicating perfect discrimination). An AUC above 0.80 is considered clinically meaningful, whereas values below this threshold suggest limited diagnostic value (3).

ROC analysis may also determine the most appropriate cutoff point for a test. This is done balancing between sensitivity and specificity. The Youden Index is a common method for identifying this optimal threshold.

4. References

1. Dimas G, Cholopoulou E, Iakovidis DK. E pluribus unum interpretable convolutional neural networks. Sci Rep. 2023 Jul 14;13(1):11421.

2. Rodríguez-Pérez R, Bajorath J. Interpretation of machine learning models using shapley values: application to compound potency and multi-target activity predictions. J Comput Aided Mol Des. 2020 Oct;34(10):1013–26.

3. Çorbacıoğlu ŞK, Aksel G. Receiver operating characteristic curve analysis in diagnostic accuracy studies: A guide to interpreting the area under the curve value. Turk J Emerg Med. 2023 Oct;23(4):195–8.

