Supplementary Material 1: Technical Algorithms and Mathematical Formulations

Random Forest

Algorithm Steps:

1. Draw T bootstrap samples from the training data.
2. For each tree, at each split, select the feature j that minimizes the Gini impurity, given by:

$$Gini(D) = 1 - \sum_{i=1}^{C} p_i^2, \tag{1}$$

where p_i is the proportion of $class_i$ in node D that belong to class i, for i=1,2,..,C

3. Grow each tree to maximum depth or until a stopping criterion is met.
4. Each tree predicts a class; the Random Forest prediction is by majority vote.
5. Feature importance is calculated as the average decrease in Gini impurity:

$$Importace(j) = \frac{1}{T}\sum_{t=1}^{T} I_{j,t} \tag{2}$$

where $I_{j,t}$ is the importance of feature j in tree t.

Multi-Layer Perceptron (MLP) Neural Network

Algorithm Steps:

1. Each layer computes a weighted sum followed by an activation function:

$$h^l = \sigma(W^{(l)}h^{(l-1)} + b^{(l)}) \tag{3}$$

where σ is the activation function

2. The output layer uses the softmax function:

$$P(x) = \frac{exp(zk)}{\sum_{j=1}^{k} exp(zj)} \tag{4}$$

where z_k is the logit for class k.

3. The loss function is categorical cross-entropy:

$$L = \sum_{i=1}^{N}\sum_{k=1}^{K} y_{i,k} log\, P(y = k|x_i) \tag{5}$$

4. Network weights are updated by backpropagation using the Adam optimizer.

Hybrid Stacking Ensemble
Algorithm Steps:

1. Train multiple base learners (Random Forest, SVM, XGBoost, MLP) on the training data.
2. Each base learner outputs predictions (usually class probabilities).
3. Collect base learner predictions as input features for the meta-learner.
4. The meta-learner (XGBoost) is trained to map base predictions to the final class label:

$$\hat{y} = f_{meta}(f_1(x), f_2(x), f_3(x), f_4(x)) \tag{6}$$

where $f_k(x)$ are the predictions of the k-th base model.

5. The final prediction is made by the meta-learner combining all base outputs.

SHAP Values (Feature Interpretability)
The SHAP value for feature j is:

$$\phi_j = \sum_{S \subseteq F\{j\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} [f_{S\cup\{j\}}(S\cup\{j\}) - f_s(X_s)] \tag{7}$$

where ϕ_j is the contribution of feature j, S is a subset of features, and f_s is the model trained on features S.