Supplementary Table 1. Summary of hyperparameter optimization strategies applied to machine learning models.

Model	Search strategy	Key parameters tuned (examples)	Cross-validation	Primary metric	Notes
RF	GridSearchCV, Optuna	n_estimators, max_depth, max_features	3-fold	F1 score	Parallel jobs enabled
XGBoost	RandomizedSearchCV, Optuna	max_depth, learning_rate, reg_alpha, reg_lambda	3-fold	F1 score	Early stopping used
LightGBM	GridSearchCV, Optuna	num_leaves, max_depth, learning_rate	3-fold	F1 score	Gradient-based sampling applied
CatBoost	RandomizedSearchCV, Optuna	iterations, depth, learning_rate	3-fold	F1 score	GPU acceleration where possible
Logistic regression	GridSearchCV	penalty, C (regularization strength)	3-fold	F1 score	Used as baseline comparator
KNN	RandomizedSearchCV	n_neighbors, weights, distance metric	3-fold	F1 score	Manhattan vs. Euclidean distance tested
ANN (MLP)	Optuna	hidden layers, neurons, activation, optimizer	3-fold	F1 score	Adam optimizer, ReLU activations

ANN, Artificial Neural Network; KNN, K-Nearest Neighbors; RF, Random Forest.

Supplementary Table 2. Final hyperparameter configurations of machine learning algorithms.

Model	Final configuration
RF	n_estimators = 300, max_features = 'sqrt', criterion = 'gini'
XGBoost	max_depth = 10, learning_rate = 0.20, reg_alpha, reg_lambda (tuned)
LightGBM	max_depth = 24, learning_rate = 0.27, gradient-based one-side sampling, exclusive feature bundling
CatBoost	iterations = 467, depth = 10, learning_rate ≈ 0.23
Logistic regression	penalty = 'l2', C = 1
ANN (MLP)	Hidden layers: 64 & 32 neurons, activation = ReLU, optimizer = Adam, iterations = 300
KNN	n_neighbors = 18, weights = 'distance', metric = Manhattan (p = 1)

ANN, Artificial Neural Network; KNN, K-Nearest Neighbors; RF, Random Forest.