CPET Analysis

Aaron HA Fletcher

September 2024

1 Non-time-series data

1.1 Preprocessing and Dimensionality Reduction

Input data was matched to the output data by research ID. All inputs and outputs were checked to ensure that no duplicates were present. If duplicates were present, and they were identical duplications, only one of the duplicates was kept. There were 4 cases of duplicates in the output data on Research ID, where the row with the complete information was selected (index 2010, 2167, 2287, 2742).

Data was valid if it had a corresponding valid BxB file. A valid BxB file is one which has more than 100 lines of data. This resulted in 3892 valid input and output records, with 256 features. All categorical features were label encoded.

Engineered features were calculated from the DateOfCPETtest and OperationDate. Each test was validated as a date (format MM/DD/YYYY, a valid calendar date and checked to ensure year was not before 1900 and after 2024). Each date was then represented as cyclical encoding using sin/cos functions, extracted to year, month and day, represented as day of the week, day of the month, boolean weekend indicator and year quarter.

Missing values were imputed using the mean of that feature. Any features that needed 30% or more of their values imputed were removed. This resulted in the removal of 17 features in table 1. Counts of missing values by remaining features are in table 2.

Finally this 259 dimensional dataset was reduced to 90% explained variance using PCA. 97 features were dropped based on features with a PCA-importance score based on PCA loadings being below mean - standard deviation or correlation coefficient being above 0.9. The features dropped within this approach are listed in table 3. The final dataset had the shape (3892, 162).

The binary outcome labels were switched for:

- Days_at_home_90_days_Binary
- Days_at_home_180_days_Binary

This was done as the minority class was originally 0.

1.2 Dataset characteristics

The dataset was split using a stratified approach to ensure balanced representation of all six binary outcomes across the splits. First, the data was divided into 80% training+validation and 20% testing sets. The 80% portion was further split into 81.25% training and 18.75% validation, resulting in final proportions of 65% training, 15% validation, and 20% testing of the total dataset. This yielded 2,529 training examples, 584 validation examples, and 779 testing examples. To maintain the distribution of all binary outcomes (90-day and 180-day days at home, and 30-day, 90-day, 720-day, and 1,825-day mortality), a composite stratification variable was created by combining all six outcomes. For example, a patient with positive 90-day days at home, negative 180-day days at home, and negative mortality outcomes would be assigned to the stratum "1_0_0_0_0_0". This approach ensured that the relative proportions of all possible outcome combinations remained consistent across the splits. The class distributions for each outcome are shown in Table 1, where class 1 represents the minority class and class of interest (positive class) for each outcome. The stratified

splitting successfully maintained similar class distributions across all splits, with the most imbalanced being 30-day mortality (2.3% positive cases) and the most balanced being 1,825-day mortality (23.2% positive cases). Class distribution is in table 4, with class 1 being the minority class and the class of interest (positive class). Due to the extreme class imbalance, SMOTE was not used as it would have resulted in more synthetic samples than actual positive cases.

1.3 Evaluation metrics

Precision-Recall AUC was used to evaluate the performance of models during hyperparameter tuning. Precision is the ratio of tp / (tp + fp) where tp is the number of true positives and fp is the number of false positives. Recall is the ratio of tp / (tp + fn) where tp is the number of true positives and fn is the number of false negatives. Precision-recall is calculated at different probability thresholds, based on every unique probability score in validation set. Area under the curve is then used to evaluate the overall performance of the model. PR AUC is used because the dataset is highly imbalanced, and precision and recall are more important than accuracy. PR AUC is also used over ROC AUC as it is more sensitive to the performance of the minority class.

1.4 Model hyperparameter tuning

A variety of models were explored for the non-time series data, such as linear models (Logistic Regression, Support Vector Machines), Tree-based models (Random Forest), Neural Netowrks (Deep Neural Networks), Probabalistic Models (Maximum Entropy) and instance based models (K-Nearest Neighbours). For each model, a hyperparameter grid search was performed using the training and validation sets to find the best performing hyperparameters. The hyperparameter search space is outlined for each model in table 5. Successful models were those which had the greatest Precision-Recall AUC score on the validation set and predicted the positive class. Final model hyperparameters are shown in table 10 along with their PR AUC score on the test set.

1.4.1 Model specific adjustments

Owing to the extreme class imbalance, adjustments were made to some models to improve their PR AUC score.

- Grid search was performed using class weights. Class weights are used to balance the loss / cost function for the minority class during training.
- DNN were trained using focal loss. Focal loss is used to prevent the model from being overconfident on the majority class.
- DNN used dropout to prevent overfitting on the majority class.
- KNN used sample replication to balance the class distribution based on class_weight function.
- Class weighting was used for all models.
- Models predictions were performed using an optimum threshold as derived from validation data split, to reduce overly aggressive or conservative predictions.

1.5 Grid search results

Best models were selected based on dual criteria: Predicted at least 1% of the positive class and Highest PR AUC score on the validation set. Outcome model specific hyperparameter tuning results on the validation set are in table 10. Generally, DNN models performed best in all cases with PR AUC scores, except for 30 day mortality, where KNN performed best. I expect that this is due to the extreme class imbalance within the Mortality at 30 days outcome (2.3% positive cases).

1.6 Test set performance

Feature	Missing (%)
Haematocrit	96.4
MeanCellHaemoglobinConc	59.6
Neutrophilsx109L	99.8
Lymphocytesx109L	99.8
Monocytesx109L	99.8
Eosinophilsx109L	99.8
Basophilsx109L	99.9
TotalProteingL	30.7
Globulin	100.0
TotalbilirubinumolL	31.6
AlkalinephosphataseIUL	31.0
ALTUL	31.2
ASTUL	99.8
CalciummmolL	35.1
AdjustedcalciummmolL	36.3
CRPmgL	65.1
FerritinugL	62.6

Table 1: Percentage of Missing Values by Features

Feature	Missing Values	Mean Value Used
VO2_mLminmLmin@Rest	145	393.41
VO2_mLminmLmin@WorkMax	25	1219.43
VO2_mLminmLmin@MaxValue	25	1520.11
VO2_mLminmLmin@VO2MaxPred	25	84.86
VO2_mLminmLmin@WorkMaxPred	25	71.92
VO2_mLminmLmin@ATVO2Max	70	70.45
VO2_mLminmLmin@ATPred	70	57.17
VO2_mLminmLmin@ATWorkMax	70	84.05
Work_WattsWatts@PredMax	58	115.38
Work_WattsWatts@AT	339	55.51
Work_WattsWatts@VO2Max	252	80.18
Work_WattsWatts@WorkMax	64	86.16
Work_WattsWatts@MaxValue	64	86.16
Work_WattsWatts@VO2MaxPred	280	74.13
Work_WattsWatts@ATVO2Max	468	92.09
Work_WattsWatts@ATPred	370	51.25
Work_WattsWatts@ATWorkMax	339	62.33
VO2WorkSlopemLminwatt@AT	621	8.99
VO2WorkSlopemLminwatt@VO2Max	471	15.01
VO2WorkSlopemLminwatt@WorkMax	347	8.10
MeanCellVolumefL	560	87.66
HaemoglobingL	555	129.63
Whitecellcountx109L	555	7.94
Plateletsx109L	561	267.02
RBCcountx1012L	560	4.44
SodiummmolL	530	138.89
PotassiummmolL	547	4.37
UreammolL	515	6.16
CreatinineumolL	513	87.74
eGFR_Calculated	516	77.54
AlbumingL	733	41.45
Age	9	70.66
Sex	3	1.37
Height	4	168.72
Weight	3	80.66
NewBMI	4	28.24
IMD_SCORE	10	7.30
LeesCRIfactors	336	0.34
TotalPolyPharm	336	2.06
PolyPharm_Cat	336	1.11

Table 2: Summary of Feature Imputation

Outcome	All Data		Train Set		Validation Set		Test Set	
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1
Days at home (90 days)	91.3%	8.7%	2310	219	533	51	712	67
Days at home (180 days)	94.1%	5.9%	2382	147	549	35	733	46
Mortality (30 days)	97.7%	2.3%	2471	58	570	14	761	18
Mortality (90 days)	97.1%	2.9%	2456	73	567	17	757	22
Mortality (720 days)	88.4%	11.6%	2235	294	516	68	689	90
Mortality (1825 days)	76.8%	23.2%	1943	586	449	135	599	180

Table 4: Class distribution across all data splits

2 Time-series data

Next the time-series data was preprocess and analysed using the same dataset splits as the non-time-series data.

For each patient, their corresponding time-series data was extracted from the BxB file. This data underwent several transformations.

- Binning and normalisation of the data.
- Imputation of revolutions per minute (RPM) data.

2.1 Binning and normalisation of the data

BxB data has variable samples, meaning that the length of this time-series data varies between patients. To address this, the data was binned into a fixed number of bins, $nbins$, to ensure uniformity across records. This adaptive binning process starts by determining the appropriate number of bins based on the input data's length. Specifically, it calculates the number of data points per bin, then divides the data index range into approximately equal intervals. Each bin then contains the mean of all numeric values within its range.

If the number of actual bins created, $actual_{n_bins}$, is less than the target n_{bins}, the data is stretched by interpolating between values to meet the target bin count. After binning, each numeric feature column undergoes normalization. This scales each column to a 0-1 range based on the minimum and maximum values within the binned data. If a column's minimum and maximum are identical, values are set to 1, preserving consistency across features.

2.2 Imputation of RPM data

RPM data was missing for 720 patients, and a column-wise mean imputation occured. Specifically, the mean of non-zero RPM values across all available files was calculated for each time step, resulting in an average RPM profile. This average was then used to replace the missing RPM values in any file where all values were missing.

2.3 Feature selection

Similar to the non-time-series data, features were selected based on explained variance and intercorrelations. All BxB data was fit to a PCA model, and Features that explain 90% of the variance based on PCA were retained. Features with a correlation coefficient above 0.9 with others were marked for removal to mitigate multicollinearity- see Table 7 for removed features. This was performed on the binned data.

2.4 Time-series models

2.4.1 BiLSTM with skip connection

2.4.2 Hierachial attention based BiLSTM

https://pmc.ncbi.nlm.nih.gov/articles/PMC9204070/ https://link.springer.com/article/10.1007/s11227-020-03560-z

2.4.3 Temporal Fusion Network

2.5 Dataset representation selection

There are 4 ways in which the time series data can be presented:

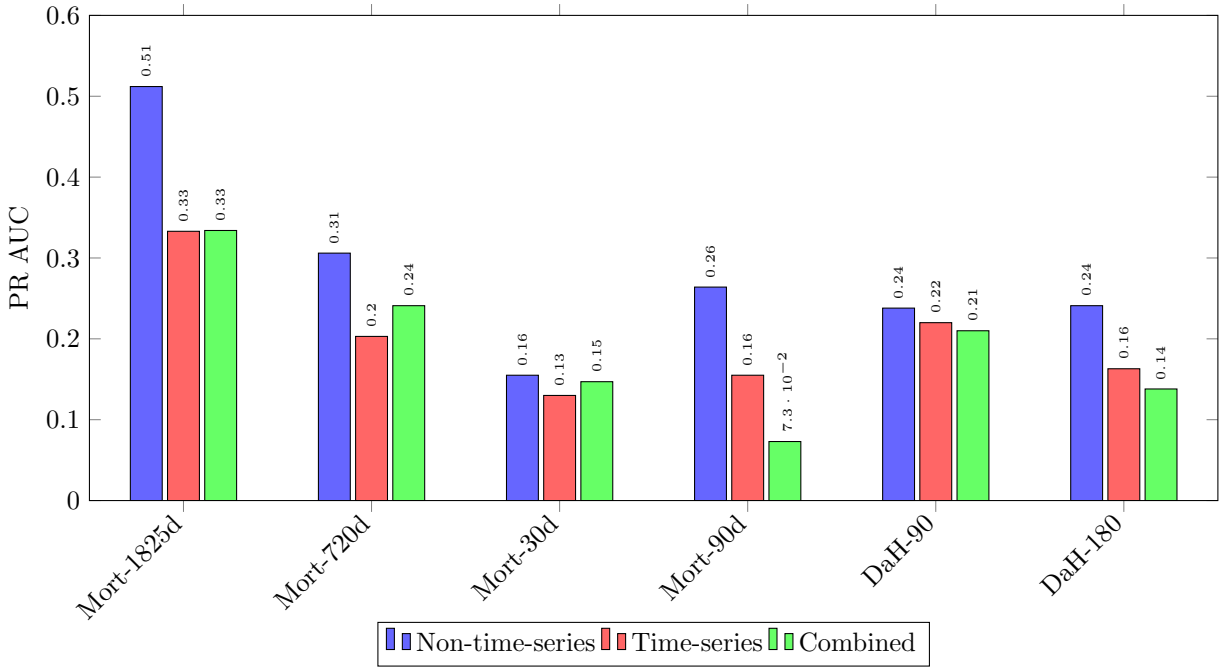
- Raw time-series data
- Binned time-series data
- PCA transformed time-series data

- PCA binned time-series data

A gridsearch was conducted on the Mortality@1825 days dataset to see which dataset format, on average, had the highest performing PR AUC. Four data representations were compared: raw (no binning or PCA), binned (binned into 100 bins), raw PCA (PCA approach on raw data) and binned PCA (PCA on binned data). This dataset was chosen as it had the most balanced data. BiLSTM with skip connections, Hierachial attention based BiLSTM and Temporal Fusion Network were used as the models for this investigation. Note that baseline PR AUC is 0.232. Of note is the poor performance of all time series models on temporal data, potentially indicating that the time-series data alone is not useful for predicting outcomes. As there was not statistical significance between the different data representations, the smaller dataset size was chosen for hyperparameter tuning.

3 Time-series and non-time-series data

Investigations into combining time-series and non-time-series data were conducted. This was done through concatenation of data from a deep neural network model (the, on average, most performing model from the non-time-series section) and a bidirectional lstm model (which was the best performing model from the time-series section). The concatenated data was then passed through a series of fully connected layers before a sigmoid activation function was applied to the output layer to output a probability. This results in poorer performance than models which just consider non-time-series data alone.



Feature Name
ChronotropicIndex
DateofCPETtest_day_cos
DateofCPETtest_day_sin
DateofCPETtest_is_weekend
DateofCPETtest_month_cos
DateofCPETtest_month_sin
DateofCPETtest_quarter_cos
DateofCPETtest_quarter_sin
HRR@AT
HRR@ATVO2Max
HRR@MaxValue
HRR@Rest
Operationdate_day_cos
Operationdate_day_sin
Operationdate_is_weekend
Operationdate_month_cos
Operationdate_month_sin
Operationdate_quarter_cos
Operationdate_quarter_sin
Systemicsteroid
VCO2_mLminmLmin@ATPred
VCO2_mLminmLmin@VO2MaxPred
VCO2_mLminmLmin@WorkMaxPred
VEVCO2@ATPred
VEVCO2@VO2MaxPred
VEVCO2@WorkMaxPred
VEVO2@ATPred
VEVO2@VO2MaxPred
VEVO2@WorkMaxPred
VdVt_est@ATWorkMax
VO2HRmLbeat@ATPred
VO2HRmLbeat@VO2MaxPred
VO2HRmLbeat@WorkMaxPred
VO2Pred@AT
VO2Pred@MaxValue
VO2Pred@Rest
VO2Pred@VO2Max
VO2Pred@WorkMax
VO2WorkSlopemLminwatt@AT
VO2WorkSlopemLminwatt@ATVO2Max
VO2WorkSlopemLminwatt@ATWorkMax
VO2_kgmLkgmin@ATPred
VO2_kgmLkgmin@VO2MaxPred
VO2_kgmLkgmin@WorkMaxPred
VO2_mLminmLmin@ATPred
VO2_mLminmLmin@VO2MaxPred
VO2_mLminmLmin@WorkMaxPred

Table 3: List of Features Selected for Removal

Table 5: Hyperparameter Search Space for All Models

Model	Hyperparameter	Values	Description
DNN	Layer Sizes	{(32), (64), (128), (64, 32), (128, 64), (256, 128)}	Number and size of hidden layers
	Activation	{relu, tanh, elu}	Activation function
	Dropout Rate	{0.0, 0.2}	Regularization strength
	Optimizer	{adam, sgd}	Optimization algorithm
	Learning Rate	{0.001, 0.01}	Step size
	Batch Size	{32, 64}	Mini-batch size
	Epochs	{10, 20, 30, 50}	Training iterations
KNN	n_neighbors	{3, 5, 7, 9, 11}	Number of neighbors
	weights	{uniform, distance}	Weight function
	metric	{euclidean, manhattan}	Distance metric
	class_weight	{None, balanced, {0:1, 1:10/25/50}}	Class weighting scheme
	leaf_size	{30, 60, 90}	Leaf size for tree
Logistic Regression	C	{0.001, 0.01, 0.1, 1.0, 10.0, 100.0}	Inverse of regularization strength
	penalty	{l1, l2, elasticnet}	Regularization type
	solver	{liblinear, saga, newton-cg, lbfgs}	Optimization algorithm
	class_weight	{None, balanced, {0:1, 1:10/25/50}}	Class weighting scheme for imbalanced data
Random Forest	n_estimators	{100, 200, 500}	Number of trees
	max_depth	{None, 10, 20, 30}	Maximum tree depth
	min_samples_split	{2, 5, 10}	Minimum samples to split
	min_samples_leaf	{1, 2, 4}	Minimum samples in leaf
	max_features	{sqrt, log2}	Features to consider
	class_weight	{None, balanced, {0:1, 1:10/25/50}}	Class weighting scheme for imbalanced data
SVM	C	{0.1, 1.0, 10.0, 100.0}	Regularization parameter
	kernel	{rbf, linear}	Kernel function
	gamma	{scale, auto, 0.1, 0.01, 0.001}	Kernel coefficient
	class_weight	{None, balanced, {0:1, 1:10/25/50}}	Class weighting scheme for imbalanced data
MaxEnt	C	{0.001, 0.01, 0.1, 1.0, 10.0, 100.0, 1000.0}	Inverse of regularization strength
	penalty	{l1, l2}	Regularization type
	solver	{liblinear}	Optimization algorithm
	max_iter	{1000, 2000, 5000}	Maximum iterations
	class_weight	{None, balanced, {0:1, 1:10/25/50}}	Class weighting scheme for imbalanced data

Table 6: Best Performing Model Perform on validation dataset following hyperparameter tuning

Outcome	Model	Accuracy	ROC AUC	PR AUC
Days at Home 90	MaxEnt	0.877	0.659	0.209
	DNN	0.808	0.583	**0.238**
	LR	0.880	0.655	0.227
	SVM	0.863	**0.671**	0.186
	KNN	**0.911**	0.542	0.161
	RF	0.870	0.654	0.2334
Days at Home 180	MaxEnt	**0.913**	0.678	0.189
	DNN	0.909	0.649	0.237
	LR	**0.913**	0.686	0.207
	SVM	0.906	0.670	0.171
	KNN	0.877	0.561	0.205
	RF	**0.913**	**0.758**	**0.241**
Mortality at 30 days	MaxEnt	0.957	**0.780**	0.073
	DNN	0.709	0.206	0.022
	LR	0.957	0.642	0.112
	SVM	0.957	0.675	0.115
	KNN	0.943	0.553	0.125
	RF	**0.961**	0.779	**0.155**
Mortality at 90 days	MaxEnt	0.952	0.797	0.143
	DNN	**0.955**	0.742	**0.264**
	LR	0.952	**0.802**	0.168
	SVM	0.949	0.714	0.129
	KNN	0.940	0.570	0.163
	RF	0.949	0.754	0.216
Mortality at 720 days	MaxEnt	0.846	0.698	0.287
	DNN	0.836	**0.708**	**0.306**
	LR	0.846	0.698	0.287
	SVM	0.638	0.256	0.116
	KNN	**0.873**	0.560	0.204
	RF	0.832	0.664	0.299
Mortality at 1825 days	MaxEnt	**0.733**	0.678	0.387
	DNN	0.606	0.630	**0.512**
	LR	0.729	**0.681**	0.390
	KNN	0.717	0.547	0.291
	RF	0.740	0.685	0.393

Table 7: Features removed due to high correlation and low PCA explained variance

VO2_Lmin L/min
VO2WorkSlope mL/min/watt
PETCO2 mmHg
VEVO2
HRR %
Ti sec
PETO2 mmHg
FETO2_Fr Fraction
VE_BTPS L/min
VO2_kg mL/kg/min
VCO2_Lmin L/min
VO2Pred %
Breath

Table 8: Dataset representation selection

Model	Raw	Binned	Raw PCA	Binned PCA
BiLSTM w/ Skip Connections	0.292 ± 0.02	0.294 ± 0.02	0.286 ± 0.02	0.286 ± 0.02
Hierachial Attention BiLSTM	0.311 ± 0.02	0.290 ± 0.01	0.304 ± 0.01	0.282 ± 0.01
Temporal Fusion Network	0.279 ± 0.02	0.286 ± 0.02	0.275 ± 0.02	0.281 ± 0.03
Average PR AUC	0.294 ± 0.02	0.290 ± 0.012	0.288 ± 0.012	0.283 ± 0.015

Performance comparison across different data representation methods for three deep learning models on time-series data. The metrics shown are PR AUC scores (mean ± standard deviation). A one-way ANOVA analysis of all four representations yielded $F = 0.5024$, $p = 0.6912$, indicating no statistically significant differences between these representation approaches. While the Raw method showed slightly higher average performance (0.294), the differences between methods were smaller than the within-method variations.

Table 9: Hyperparameter Search Space for all temporal models

Model	**Hyperparameter**	**Values**	**Description**
BiLSTM w/ Skip Connections	Hidden Layer Sizes	{(32), (64), (128)},	Dimension of hidden layers
	Number Layers	{1, 2, 3}	Number of stacked hidden layers
	Sequence Length	{50,100}	Sequence length
	Dropout	{0.2, 0.3, 0.5}	Dropout rate
	Learning Rate	{1e-4, 3e-4, 1e-3}	Learning Rates
	Batch Size	{32, 64, 128}	Mini-batch size
	Epochs	{30, 50}	Training iterations
Temporal Fusion Network	Hidden Layer Sizes	{32, 64, 128}	Dimension of hidden layers
	Number Layers	2, 3, 4	Number of temporal conv layers
	Dropout	0.3, 0.5	Dropout rate
	Learning Rate	1e-4, 3e-4	Learning Rates
	Batch Size	32, 64	Mini-batch size
	Epochs	30, 50	Training iterations
	Sequence Length	{50,100}	Sequence length
Hierarchial Attention Network	Hidden Layer Sizes	{32, 64, 128}	Dimension of hidden layers
	Number Layers	2, 3, 4	Number of temporal conv layers
	Dropout	0.3, 0.5	Dropout rate
	Learning Rate	1e-4, 3e-4	Learning Rates
	Batch Size	32, 64	Mini-batch size
	Epochs	30, 50	Training iterations
	Sequence Length	{50,100}	Sequence length

Table 10: Best Performing Model Perform on validation dataset following hyperparameter tuning

Outcome	Model	Accuracy	ROC AUC	PR AUC
Days at Home 90	BiLSTM w/ Skip Connections	0.817	**0.692**	**0.220**
	Hierachial Attention	0.644	0.581	0.157
	Temporal Fusion Network	**0.832**	0.585	0.179
Days at Home 180	BiLSTM w/ Skip Connections	**0.777**	**0.626**	**0.163**
	Hierachial Attention	0.555	0.570	0.118
	Temporal Fusion Network	0.741	0.595	0.143
Mortality at 30 days	BiLSTM w/ Skip Connections	0.952	**0.682**	**0.130**
	Hierachial Attention	0.534	0.669	0.041
	Temporal Fusion Network	**0.973**	0.567	0.051
Mortality at 90 days	BiLSTM w/ Skip Connections	0.959	**0.738**	**0.155**
	Hierachial Attention	0.625	0.672	0.072
	Temporal Fusion Network	**0.973**	0.672	0.112
Mortality at 720 days	BiLSTM w/ Skip Connections	**0.853**	0.602	0.203
	Hierachial Attention	0.598	**0.625**	0.172
	Temporal Fusion Network	0.668	0.610	**0.226**
Mortality at 1825 days	BiLSTM w/ Skip Connections	0.375	**0.615**	0.333
	Hierachial Attention	0.231	0.595	0.297
	Temporal Fusion Network	**0.390**	0.610	**0.339**

Gridsearch results for the best performing model on the validation dataset following hyperparameter tuning. Note that the PR AUC scores performance performed superior to other approaches three out of five times, with temporal fusion network performing better once better class balance was achieved.

Table 11: Comparative results of non-time-series, time-series and combined models

Outcome	Model	Accuracy	ROC AUC	PR AUC
Mortality at 1825 days	Non-time-series (DNN)	0.606	0.630	**0.512**
	Time-series (BiLSTM)	0.375	0.615	0.333
	Combined (DNN + BiLSTM)	**0.607**	**0.671**	0.334
Mortality at 720 days	Non-time-series (DNN)	0.836	**0.708**	**0.306**
	Time-series (BiLSTM)	**0.853**	0.602	0.203
	Combined (DNN + BiLSTM)	0.729	0.683	0.241
Mortality at 30 days	Non-time-series (RF)	0.961	**0.779**	**0.155**
	Time-series (BiLSTM)	0.952	0.682	0.130
	Combined (DNN + BiLSTM)	**0.966**	0.802	0.147
Mortality at 90 days	Non-time-series (DNN)	0.955	0.742	**0.264**
	Time-series (BiLSTM)	**0.959**	0.738	0.155
	Combined (DNN + BiLSTM)	0.892	**0.755**	0.073
Days at Home 90	Non-time-series (DNN)	0.808	0.583	**0.238**
	Time-series (BiLSTM)	**0.817**	**0.692**	0.220
	Combined (DNN + BiLSTM)	0.747	0.616	0.210
Days at Home 180	Non-time-series (RF)	**0.913**	**0.758**	**0.241**
	Time-series (BiLSTM)	0.777	0.626	0.163
	Combined (DNN + BiLSTM)	0.872	0.675	0.138