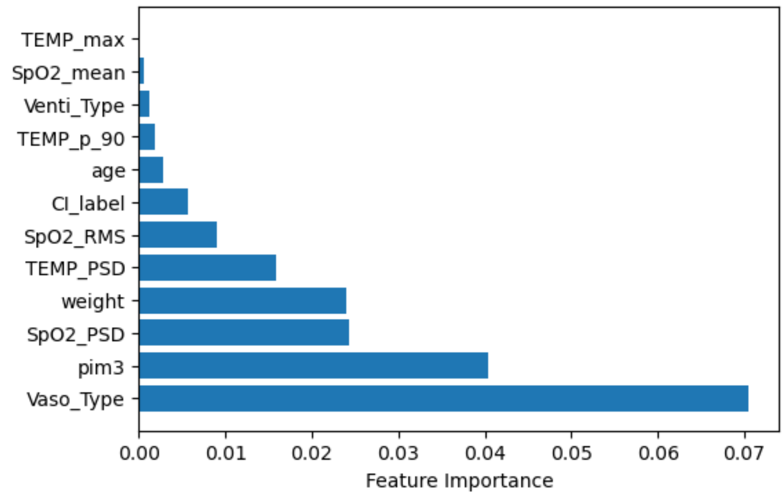
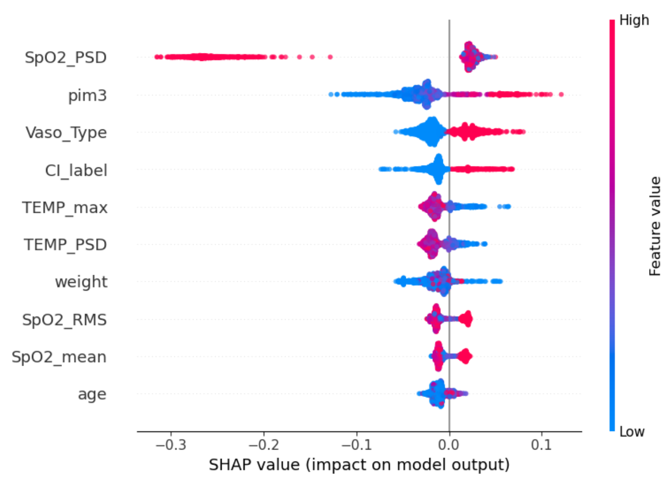
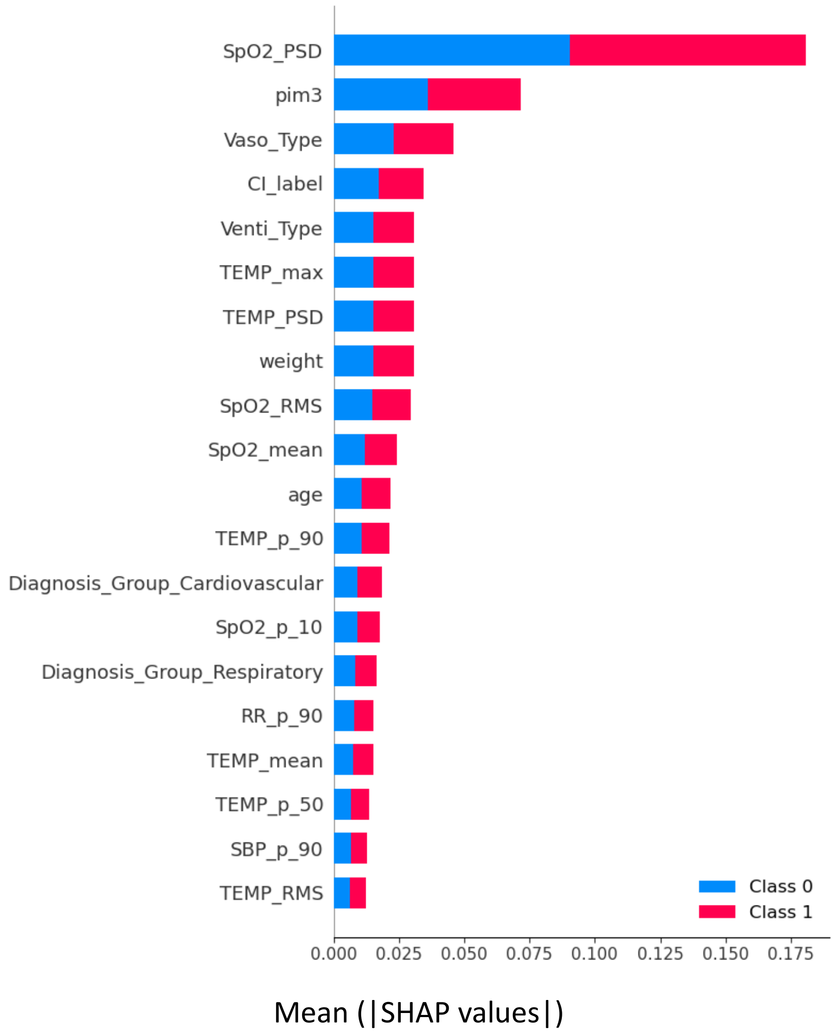
Supplementary Information



Supplementary Figure S1: In Random Forest, feature importance is usually calculated based on the mean decrease in impurity (MDI) from each feature. Essentially, it measures how much each feature decreases the uncertainty of the model (e.g., Gini impurity for classification tasks). It is computed during the construction of the Random Forest by adding up the weighted impurity decrease for all nodes a feature is used to split.



Supplementary Figure S2: Contribution of features to model predictions using the SHapley Additive exPlanations (SHAP) analysis. The colours represent feature values: red for greater values and blue for smaller ones. A positive SHAP value for a feature indicates that higher values of that feature contribute to driving the model prediction towards the outcome of death, while a negative SHAP value for a feature indicates that lower values of that feature contribute to predicting the survival class.



Supplementary Figure S3: In SHAP values, based on game theory, aim to explain the prediction of an instance by computing the contribution of each feature to the prediction. SHAP values provide both local explanations (for individual predictions) and global explanations (across the entire dataset). It does this by comparing what a model predicts with and without each feature, but unlike MDI, it does so by taking all possible combinations of features into account.

Supplementary Table 1. Description of model input features.

Category	Feature	Feature description
Categorical data	age	Age at admission
	weight	Weight
	pim3	Paediatric Index of Mortality 3
	gender	Gender
	NO_Type	Inhale nitric oxide support (Yes/No)
	ECMO	ECMO support (Yes/No)
	Venti_Type	Invasive mechanical ventilation or Non-invasive ventilation support
	Vaso_Type	Vasoactive support (Yes/No)
	Diagnosis_Group_Cardiovascular	Yes/No
	Diagnosis_Group_Gastrointestinal	Yes/No
	Diagnosis_Group_Infection	Yes/No
	Diagnosis_Group_Neurological	Yes/No
	Diagnosis_Group_Respiratory	Yes/No
	Diagnosis_Group_Others	Yes/No
	CI_label	Critical Incident Event (Yes/No)
Numeric data (vital signs: systolic blood pressure (SBP), diastolic blood pressure (DBP), mean blood pressure (MBP), heart rate (HR), respiratory rate (RR), peripheral capillary oxygen saturation (SpO_2), body temperature)	mean	Mean value
	std	Standard deviation
	max	Maximum of signal x
	min	Minimum of signal x
	p_10	10th percentile value
	p_90	90th percentile value
	p_50	50th percentile value (= median)
	kurtosis	Kurtosis
	skewness	Skewness
	RMS	Root mean square (RMS)
	Peak	Peak value: $0.5 * (\max(x_i) - \min(x_i))$
	PSD	Power Spectral Density (PSD): Summation of power of signal $x(_i)$
	Fuzzy entropy	Uncertainty or randomness in time series
Note: For categorical data, they are directly used as features in the feature vector. For numeric data - Z-scores of high-frequency vital signs, features are calculated for each kind of vital signs (i.e., SBP, DBP, MBP, HR, RR, SpO_2, and body temperature) respectively.		

Supplementary Table 2. Data cleaning and preparation.

Data Type	Exclusion Criteria	Correction Method
Heart Rate	Less than 30 or greater than 300 beats per minute	Interpolated using linear regression
Body Temperature	Below 25°C or above 41°C	Abnormal readings adjusted to 36.5°C; missing values imputed with 36.5 °C
Oxygen Saturation (SpO_2)	Below 30%	Interpolated using linear regression
Blood Pressure	No specific range criteria	Backward interpolation to impute missing values

Supplementary Table 3. Free and open source software used in analysis and modelling.

Software Package	Version	License Type	Description	Citation
Python	3.10.9	PSF License	Programming language used for data analysis	Python Software Foundation. Python Language Reference, version 3.10.9. Available at http://www.python.org.
NumPy	1.23.5	BSD License	Numerical computing library	Harris, C.R., Millman, K.J., van der Walt, S.J., et al. (2020). Array programming with NumPy. Nature, 585, 357–362.
pandas	1.5.3	BSD License	Data manipulation and analysis library	McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 51-56.
scikit-learn	1.2.1	BSD License	Machine learning library	Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.
TensorFlow	1.7.5	Apache-2.0	Deep learning framework	Abadi, M., Barham, P., Chen, J., et al. (2016). TensorFlow: A System for Large-Scale Machine Learning. OSDI'16: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, 265-283.
Keras	2.12.0	MIT License	Deep learning library	Chollet, F., et al. (2015). Keras: The Python Deep Learning Library. https://keras.io.
Matplotlib	3.7.0	PSF License	Plotting library for Python	Hunter, J.D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90-95.
Seaborn	0.12.2	BSD License	Statistical data visualization library	Waskom, M., et al. (2020). Seaborn: Statistical Data Visualization. Journal of Open Source Software, 6(60), 3021.
SHAP	0.41.0	MIT License	Explainable AI visualization library	Lundberg, S.M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems, 30, 4765-4774.