

Explainable Artificial Intelligence for Medical Science

Jamie Andrew Duell

Submitted to Swansea University in fulfilment
of the requirements for the Degree of Doctor of Philosophy




Swansea University
Prifysgol Abertawe

Department of Computer Science
School of Mathematics and Computer Science
Swansea University

December 2023

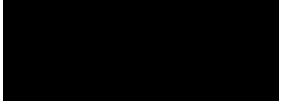
Declaration

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed  (candidate)


Date 18/12/2023

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed  (candidate)

Date 18/12/2023

I hereby give consent for my thesis, if accepted, to be available for electronic sharing

Signed  (candidate)

Date 18/12/2023

The University's ethical procedures have been followed and, where appropriate, that ethical approval has been granted.

Signed  (candidate)

Date 18/12/2023

Abstract

Explainable Artificial Intelligence (XAI) is at the forefront of Artificial Intelligence (AI) research. As the development of AI has become increasingly complex with modern day computational capabilities, the transparency of the AI models decreases. This promotes the necessity of XAI, as it is illicit as per the General Data Protection Regulations (GDPR) “right to an explanation” to not provide a person with an explanation given a decision reached after algorithmic judgement. The latter is crucial in critical fields such as Healthcare, Finance and Law. For this thesis, the Healthcare field and more specifically Electronic Health Records are the main focus for the development and application of XAI methods.

This thesis offers prospective approaches to enhance the explainability of Electronic Health Records (EHRs). It presents three different perspectives that encompass the *Model*, *Data*, and the *User*, aimed at elevating explainability. The model perspective draws upon improvements to the local explainability of black-box AI methods. The data perspective enables an improvement to the quality of the data provided for AI methods, such that the XAI methods applied to the AI models account for a key property of missingness. Finally, the user perspective provides an accessible form of explainability by allowing less experienced users to have an interface to use both AI and XAI methods.

Thereby, this thesis provides new innovative approaches to improve the explanations that are given for EHRs. This is verified through empirical and theoretical analysis of a collection of introduced and existing methods. We propose a selection of XAI methods that collectively build upon current leading literature in the field. Here we propose the methods Polynomial Adaptive Local Explanations (PALE) for patient specific explanations, both Counterfactual-Integrated Gradients (CF-IG) and Quantified Uncertainty Counterfactual Explanations (QUCE) that utilise counterfactual thinking, Batch-Integrated Gradients (Batch-IG) to address the temporal nature of EHR data and Surrogate Set Imputation (SSI) that addresses missing value imputation. Finally, we propose a tool called ExMed that utilises XAI methods and allows for the ease of access for AI and XAI methods.

Acknowledgements

First and foremost, I would like to thank Dr. Xiuyi Fan for constant guidance and support throughout the duration of my PhD. Throughout the years of my study, Xiuyi played an integral part in my growth as not only an academic but as a person, I could not be more thankful for the unprecedented guidance that I *promise* to pass forward. Secondly, to my supervisor Dr. Monika Seisenberger for continued support throughout the final year of my PhD and involving me in different events, providing me with valuable life experiences I would not have been able to get elsewhere, whilst simultaneously supporting my writing, identifying erratum, and thus helping me improve greatly during a crucial period of time.

I would like to thank my mother Donna Duell and grandfather David Duell. Merely expressing my gratitude through the preface of this thesis is not enough to emphasize my appreciation for everything. So, I hope my actions posterior to my PhD will be enough to suffice. I am deeply thankful for everything.

I would also like to thank my colleagues and friends, namely: Veera Raghava Reddy Kovvuri, Kira Pugh, Xavier Crean, Harry Bryant, Le Minh Thao Doan. I am also thankful for my friends, Kieran John, Thomas Macpherson, Zion Sky, Mark Beardsley, Keithleen Yaoto, Kacper Blaszczyk, Joel Okpara, Yun Tung Venessa Lau, Ryan Gardiner, Sarah McCall, Aqsa Abbas and Jamie Myland - for providing me with words of encouragement and constant support in a plethora of ways.

I would also like to briefly thank the anonymous reviewers for the manuscripts that are contained within the body of this thesis, both those who rejected and accepted the manuscripts played a crucial role in enhancing their quality. I am deeply grateful for their valuable feedback, which greatly contributed to the overall improvement of the manuscripts.

Finally, none of this would have been possible without the support of the UKRI Centre for Doctoral Training in Artificial Intelligence, Machine Learning & Advanced Computing (AIMLAC CDT). Therefore, I extend my gratitude to all members of the AIMLAC CDT, with special mention to Professor Gert Aarts and Roz Toft.

I attribute any accomplishments to those listed above.

Contents

Contents	ix
List of Figures	xii
List of Tables	xviii
I Introduction	1
1 Thesis Introduction	3
1.1 Motivation	3
1.2 Aims and Contributions	10
1.3 List of Publications	11
1.4 Conference/Workshop Talks	15
II Background	17
2 Explainable Artificial Intelligence	19
2.1 Introduction	19
2.2 Feature Attribution	21
2.3 Counterfactual Explanations	29
2.4 Conclusion	30
III Evaluating Existing XAI Methods	31
3 A Comparison of Model-Agnostic Explanations Given on Electronic Health Records	33
3.1 Introduction	33
3.2 Methods and Materials	35
3.3 Results	38
3.4 Explanations	41
3.5 Results	49

3.6	Conclusions	52
IV Enhancing Explainability - a Model Perspective		55
4	Polynomial Adaptive Local Explanations	57
4.1	Introduction	57
4.2	Related Work	58
4.3	Method	59
4.4	Comparative Methods	62
4.5	Results	63
4.6	Conclusion	66
5	Counterfactual Integrated Gradients	69
5.1	Introduction	69
5.2	Integrated Gradients	72
5.3	Counterfactual Explanations	72
5.4	Method	74
5.5	Results	78
5.6	Conclusion	83
6	Formalising Batch-Integrated Gradients for Temporal Explanations	87
6.1	Introduction	87
6.2	Method: Batch-Integrated Gradients	89
6.3	Properties for Explainability	91
6.4	Formal Evaluation	93
6.5	Controlled Experiments	96
6.6	Applications	98
6.7	Conclusion	101
7	The Minimisation and Quantification of Path-Based Uncertainty for Generative Counterfactual Explanations	103
7.1	Introduction	103
7.2	Axioms for Path-Based Explainers	105
7.3	Proposed Model: QUCE	107
7.4	Experimental Setup	115
7.5	Quantitative Evaluation	116
7.6	Conclusion	118
V Enhancing Explainability - a Data Perspective		119
8	Explaining Incomplete Data	121
8.1	Introduction	121
8.2	Background	124

8.3	Nominative Properties of Imputation and Explanation Methods	127
8.4	Imputation Method	130
8.5	Evaluation	132
8.6	Experiment Results	133
8.7	Conclusion	140
VI Enhancing Explainability - a User Perspective		143
9	ExMed: An AI Tool for Experimenting Explainable AI Techniques on Medical Data Analytics	145
9.1	Introduction	145
9.2	Related Work	146
9.3	ExMed Workflow	147
9.4	Case Study I: COVID-19 Control Measures	148
9.5	Case Study II: Lung Cancer Life Expectancy	152
9.6	Conclusion	154
VII Conclusion		157
10	Summary	159
10.1	Conclusion	159
10.2	Future Work	161
A	Machine Learning	167
A.1	Linear Regression	167
A.2	Logistic Regression	168
A.3	k-Nearest Neighbour	169
A.4	eXtreme Gradient Boosting	170
A.5	Artificial Neural Networks	170
A.6	Supplementary Mathematics for ML	172
B	Parametric effects on the Gaussian Distribution	181
C	Simulacrum Data	183
D	Conceptualising Batch-Integrated Gradients for Temporal EHR Explanations	185
D.1	Introduction	185
D.2	Method	187
D.3	Results	188
D.4	Conclusion	193
E	QUCE Supplementary Material	195

E.1	Computing QUCE Explanations	195
E.2	Proof of Proposition 2	197
E.3	Proof of Proposition 3	198
E.4	Counterfactual Reconstruction Error	199
E.5	QUCE Evaluated against Further Properties of Explainability	199
E.6	Experimental Setup	200
E.7	Deletion Experiments	201
F	Multiple Value Imputation Experiments	203
G	Further Comparisons of Explanations on EHRs	207
G.1	Introduction	207
G.2	Background	209
G.3	Method	209
G.4	Dataset and Prediction Result	212
G.5	Explanation Results	213
G.6	Conclusion	216

List of Figures

1.1	Illustration of the years of publication and the XAI methods utilised for the papers in Table 1.1.	8
2.1	Prevalence of the search terms “Explainable Artificial Intelligence” and “Explainable AI”, given between the years 2013-2023. These results are extracted from Google Trends. Empirically, there is evidence of an increase in the search terms post 2016.	20
2.2	Simple illustration for the intuition behind the LIME method to explain the large red cross. Here, we let the circle define the neighbourhood. The background colour depicts the black-box classifier. The dashed line illustrates the local linear model. The colour saturation depicts the importance of each feature (bold has a stronger weight associated). The black points illustrate instances that are outside of the weighted neighbourhood.	22
2.3	Simple intuition of the Integrated Gradients method. Here a straight line is given from the baseline \mathbf{x}' to the instance to explain \mathbf{x} . Here, $\mathbf{x} = \langle 2, 2 \rangle \in \mathbb{R}^2$ and $\mathbf{x}' = \langle 0, 0 \rangle \in \mathbb{R}^2$	28
3.1	Comparison of algorithm performance across each available data set, with performance metrics for Logistic Regression, XGBoost and the EBM method	39

3.2	AUROC for LC-DA determining the best fit for the model.	40
3.3	The AUROC for LC-MD determining the best fit for the model.	40
3.4	AUROC for LC-ST determining the best fit for the model.	41
3.5	SHAP global explanation for the LC-DA problem, the x-axis provides a weighting with positive SHAP values shifting towards survival and negative SHAP values shifting towards deceased.	42
3.6	Direct comparison of feature attribution towards the output classes. From the bar plot we can determine that “Weight” is the most important feature that corresponds to longer survival, with “M Best” having the most impact for short term survival	42
3.7	SHAP global explanation for the LC-ST problem with feature attribution measured against class[0] <i>less than 6 months survival</i> . From this we can observe that the most influential features towards least survival are “M-Best”, “Age” and “Weight”.	43
3.8	SHAP global explanation for the LC-ST problem with feature attribution measured against class[1] <i>between 6 and 12 months survival</i> From this we can observe that the most influential features towards the longest survival time are “Weight”, “M-Best” and “Height”.	43
3.9	SHAP global explanation for the LC-ST problem with feature attribution measured against class[2] <i>greater than 12 months survival</i> . From this we can observe that the most influential features towards the survival bracket are “Age”, “Weight” and “CReg Code”.	44
3.10	SHAP global explanation for the LC-MD problem. We observe that the top 3 most influential features are “Weight”, “Time Delay” and “Height” towards the reduction of drug dose administration.	45
3.11	An explanation generated by LIME for Alive / Deceased classificaiton.	46
3.12	An explanation generated by SHAP for a patient: The width of each descriptive block and colour are indicative of the shift in probability to a given case. The colors red and blue denote the direction of prediction shift towards the target classes “Dead” and “Alive,” respectively.	46
3.13	Anchors give a conditional conjunction of cases, which identify that given certain elements are true; then the prediction will suffice. We see the coverage of these conditions also provided with a precision value $> \tau$	46
3.14	Demonstration of a local LIME explanation for the LC-ST problem, We observe that the three most important contributors to the prediction outcome for < 12 Months survival are “Height”, “M Best”, “CNS”.	47
3.15	Demonstration of a local SHAP explanation for the LC-ST problem. We observe that the three most important instances contributing to the patients predicted survival time are “Height”, “Performance” and “CNS”. From the output we observe a shift towards the defined target class “ > 6 Months and < 1 year”.	47

3.16	Demonstration of a local Anchors explanation for the LC-ST problem. Demonstrating all the Anchors that contribute towards the prediction with the coverage and precision. We can observe “Cycle Number”, “Height” and “M Best” as the first set of anchors provided.	47
3.17	Demonstration of a local LIME explanation for the LC-MD problem, from this we can observe that the largest impact toward drug dose reduction being predicted is the tumour “Behaviour” being malignant followed by “Time Delay” and “Age”.	48
3.18	Demonstration of a local SHAP explanation for the LC-MD problem. We observe that “Regimen Stopped Early”, “Height” and “Weight” hold the greatest influence towards drug dose reduction being necessary.	48
3.19	Demonstration of a local Anchors explanation for the LC-MD problem. From this, we can observe the anchors that the first returned anchors for drug dose reduction are “Height”, “Morph” and the “Cancer Plan”.	49
3.20	Comparing the shared features across each problem using both SHAP and LIME	50
3.21	Most important feature returned or the first anchor (scoped rules) for the first 1000 instances on the test data set for both LC-DA and LC-MD problems.	51
3.22	Most important feature returned or the first anchor (scoped rules) for the first 400 instances on the test data set for the LC-ST problem.	51
4.1	RMSE measurements for a subset of 100 Simulacrum patient instances across 4 datasets. We can observe how the increase in polynomial degree improves the local model accuracy for most instances, but in some cases a simpler model will suffice.	64
4.2	Derivation of the quadratic polynomial term - Simulacrum patient instance. The explanation determines how an instantaneous increase in each feature value x_i influences the local polynomial function $g_{m,i}$ at the location of the instance, where we have $g_{2,i}$. Higher (resp. lower) values on the y -axis represent a large (resp. small) feature importance value.	65
4.3	Derivation of the cubic polynomial term - Simulacrum patient instance. The explanation determines how an instantaneous increase in each feature value x_i influences the local polynomial function $g_{m,i}$ at the location of the instance, where we have $g_{3,i}$. Higher (resp. lower) values on the y -axis represent a large (resp. small) feature importance value.	66
4.4	A comparison of explanations given by the linear model, quadratic model, cubic model and the SHAP model for a patient instance.	67
5.1	Illustrative example of CF-IG. Given an input \mathbf{x} , its linear interpolation to its nearest counterfactual example $\hat{\mathbf{x}}'_c$ in the dataset is shown. The explanation $\hat{\mathbf{x}}_c$ produced by CF-IG is the point crossing the decision boundary (the dotted line) on this interpolation. CF-IG also produces feature-attribution values for its explanations.	71

5.2	Illustrative explanation example of the CF-IG method, highlighting the bar interaction features of the explainer. Here the counterfactual method is applied to a breast cancer patient example. In this explanation we inspect the features: <i>T Best</i> , <i>Dose Administration</i> and <i>Weight</i> . Here we also observe the magnitude of the <i>Clinical Trial</i> attribution by value.	84
5.3	Illustrative explanation example of the CF-IG + Wachter method, highlighting the bar interaction features of the explainer. Here the counterfactual method is applied to a breast cancer patient example. In this explanation we inspect the features: <i>T Best</i> , <i>Dose Administration</i> and <i>Weight</i> . Here we also observe the magnitude of the <i>Ethnicity</i> attribution by value.	84
5.4	Illustrative explanation example of the CF-IG + DiCE method, highlighting the important features identified by the explainer in changing the outcome. Here the counterfactual method is applied to a breast cancer patient example.	85
6.1	Illustrative example of Batch-IG. Given an input at time point \mathbf{x}_t , Interval-IG is demonstrated between time points, and the path for accumulated gradients over linear interpolations is shown to the destination time point \mathbf{x}_{t+2} . Here we show how Batch-IG would traverse through a cluster to minimise out-of-distribution interpolations. Batch-IG also produces feature-attribution values for its explanations.	89
6.2	Attribution over time transitions, between weeks 1, 2 and 4 for a student for predicting the how determined they are, where the accumulated average gradients per time-interval are associated with Likert value 5 for determined (extremely determined.).	99
6.3	Attribution over time transitions, between the days 12 and 13, whereby most features had transitioned into the next period of time that the control measures had been in place.	100
7.1	A simple illustration demonstrating QUCE generated examples and associated paths. It illustrates paths and counterfactual examples weighted towards being in distribution. The dotted line is the straight line IG path towards the QUCE generated examples. The bold line is the QUCE generated path. The grey point is an example of an instance generated on proximity and opposing class alone.	106
7.2	Here we illustrate two explanations produced on the Wisconsin Breast Cancer Dataset given by the proposed QUCE method. We observe how each feature influenced the change in the prediction in attempting to generate a counterfactual example. We see the left explanation has almost no uncertainty in generated explanation, whereas the right image demonstrates a large degree of uncertainty in the generated counterfactual explanation.	111

8.1	The framework of the Surrogate Set Imputer method. Green arrows represent returned outputs; Red arrows represent an iterative process; and the Blue represent transitions in the pipeline once the Green-Red iterative processes are completed.	125
8.2	The mean ΔI for the SSI, kNN, MICE, GAIN, SoftImpute and MissForest imputation methods on the W-BC dataset as the number of missing features is increased randomly for each instance. Here we observe a lower average error for the SSI method.	139
8.3	The mean EF for the SSI, kNN, MICE, GAIN, SoftImpute and MissForest imputation methods on the W-BC as the number of missing features is increased randomly for each instance. Here we observe more similar explanations produced using SSI when compared to the ground truth. . . .	139
8.4	An example explanation for imputation given for an instance taken from the the Wisconsin Breast Cancer dataset, providing the imputed feature values, attribution values and magnitude of importance.	141
9.1	ExMed Workflow. ExMed provides the user with a sequence of simple actions, including loading, merging and editing data, and creating prediction as well as explanation models. Various visualisation techniques are supported in several stages of this pipeline.	147
9.2	Example of an Explanation computed with SHAP and LIME. For this instance, both explainers consider top measures contributing to this prediction being <i>Domestic Travel</i> , <i>Cafes and Restaurants Closure</i> and <i>Pubs and Bars Closure</i>	150
9.3	Global explanations generated using SHAP on our COVID dataset for the prediction whether $R_t \geq 1$. We see that closing down cafes and restaurants as well as pubs and bars are the most effective control measures. When their feature values are high (red), they have a strong negative impact to the prediction; whereas when their feature values are low (blue), they have strong positive impact to the prediction.	151
9.4	Local explanation on the Lung Cancer life expectancy data set for a patient instance. We see that the most impactful features amongst SHAP and LIME are ubiquitous: “Grade” <i>How the cancer cells act; the higher the grade the less normality the cell resembles and it may act more aggressive</i> and “M Best” <i>Presence or Absence of Distant Metastatic Spread</i> , followed by a disagreement on age attribution.	153
9.5	The largest impact towards the survival boundaries <i>greater than 1 year and less than 6 months</i> is the cancer grade. It has a direct impact on the longest and least time survival. Height, weight and patient age are also significant factors.	154
9.6	Global explanation measuring feature attribution against the class <i>Survival time of less than 6 months</i> , where we see the cancer grade of higher value - indicative of cell abnormality and aggressiveness, followed by “M Best”, “weight” and “height” determinants of body mass index (BMI) and “age”.	155

A.4	Solving for β through the normal equation, with $\beta = (X^T X)^{-1} X^T y$ and $\beta \pm 1$. Here, it is shown that the best fit is given by the normal equation β .	168
A.5	Extrapolated from the linear regression example in Figure A.1, by applying the Sigmoid function to the solution of βX . The previous values of y in linear regression have been transformed such that, a value 1 is assigned if $y \geq \frac{1}{N} \sum_{y \in \mathbf{y}} y$ and 0 otherwise. Here, N is the number of samples in both X and y respectively.	169
A.9	Examples of the kNN algorithm applied to $k = \{1, 3, 5\}$. Here, consider two classes, one containing black points, the other containing grey. A new data point in red is added, and then assigned to the appropriate class. Here, when $k = 1$ the red point is assigned to the black class. When $k = 3$, the red point is assigned to the grey class. When $k = 5$, the red point is assigned to the black class.	170
A.10	Demonstrative diagram for the example Neural Network given in section A.5.	171
A.11	Simple example of gradient descent on a one dimensional value. Here, our value is initialised at 5. x is then updated through the gradient descent process, with a learning rate of $\alpha = 0.1$ and 10 step iteration process. Here, each red point illustrates a step location of the function with respect to each updated x . Here, we see convergence from example A.1 to the value $f(x) \approx 0.0022$.	179
A.12	Simple example of gradient descent on a two dimensional input. Here, our inputs are initialised at $x = 5$ and $y = 3$, where $z = f(x, y)$. x and y are then updated through the gradient descent process, with a learning rate of $\alpha = 0.1$ and 20 step iteration process. Here, each red point illustrates a step location of the function with respect to each update.	180
B.3	Collection of plots illustrating the effects of μ and σ^2 on the distribution.	181
D.1	Feature attribution for the features from time interval $t_0 \rightarrow t_1$. We observe the dose administration had positive attribution towards the class ≥ 6 Months and the cycle number transition from $1 \rightarrow 3$ had negative attribution towards the ≥ 6 Months class.	189
D.2	Feature attribution for the features from time interval $t_1 \rightarrow t_2$. We observe the dose administration had positive attribution towards the class ≥ 6 Months. This time interval was observed during the drug cycle 3, where there exists only change to the drug administration.	190
D.3	Feature attribution for the features from time interval $t_2 \rightarrow t_3$. We observe the dose administration had negative attribution towards the class ≥ 6 Months and the cycle number transition from $3 \rightarrow 5$ also had negative attribution towards the ≥ 6 Months class.	190
D.4	Evaluation of the partial derivative of the prediction w.r.t the change in drug dose administration between time intervals $t_0 \rightarrow t_1$.	191
D.5	Evaluation of the partial derivative of the prediction w.r.t the change in drug dose administration between time intervals $t_1 \rightarrow t_2$.	192

D.6	Evaluation of the partial derivative of the prediction w.r.t the change in drug dose administration between time intervals $t_2 \rightarrow t_3$	193
F.1	Here we have the random imputation experiments on the ΔI metrics for (from left to right): SBC, SLC, SLyC, SRC, SSC, Diabetes and SEER datasets. Here we observe a competitive performance displayed with the proposed SSI method.	204
F.2	Here we have the random imputation experiments for EF on the datasets (from left to right): SBC, SLC, SLyC, SRC, SSC, Diabetes and SEER. Here we observe aberration in performance with all methods, with a competitive performance displayed by the proposed SSI method.	205
G.1	Global explanation for LIME across the factual (NonCF) and counterfactual (CF) data set. From this, we can observe that across both factual and counterfactual datasets, “M Best” is the most import feature. We observe a strong similarity in feature attribution towards predictions.	213
G.2	Global explanation for SHAP across the factual (NonCF) and counterfactual (CF) data set. From this, we can observe that across both factual and counterfactual datasets, there’s a similarity in feature attribution towards predictions, with the most importance on the feature “Weight”.	214
G.3	Demonstrating the pearson correlation between the global explanations from SHAP and LIME.	214

List of Tables

1.1	The set of papers exploring how XAI has been used in medicine. This is accompanied by the XAI methods used, the medical research question proposed, the year of publication and the type of explanation method. Here we observe a wide range of applications often utilise the same set of XAI methods with LIME and SHAP being common in application.	6
3.1	Overview of available Simulacrum data set tables with the corresponding number of columns.	36
3.2	Status of Survival Time Feature Missing Values	37
3.3	Baseline performances for logistic regression, XGBoost an EBM tested on each medical problem.	38
3.4	A sample patient record in the Simulacrum data set for the LC-DA problem	45
3.5	A sample patient record in the Simulacrum data set for the LC-ST problem	47

3.6	A sample patient record in the Simulacrum data set for the LC-MD problem	48
5.1	Comparison of the property satisfiability of counterfactual methods. Here ‘✓’ indicates the property is satisfied, ‘✗’ indicates the property is not satisfied and ‘-’ indicates that the property is not applicable or cannot directly be evaluated by the method.	80
5.2	Comparison of the consistency given across a collection of counterfactual methods that produce attribution/importance values. We observe that the CF-IG method produces consistent explanations given $N = 100$ and $R = 10$.	81
5.3	Comparison of the proximity between original and counterfactual instances using the cosine distance between vectors. This is experimented over 100 instances on each dataset.	82
5.4	Comparison of the proximity between original and counterfactual instances using l_2 distance. This is experimented over 100 instances on each dataset.	82
5.5	Encoded breast cancer patients for counterfactual explanations given on for a patient instance. Here for clarity ‘ \approx ’ implies an infinitesimally small change to a feature value. Here we observe pairs of each explainer (separated with a double line) without (resp. with) the addendum of CF-IG, the closest distance to the origin instance for the pairs is in bold . For each pair we observe that CF-IG produces values that are closer to the original instance. Here we evaluate the normalised instances and corresponding Euclidean distance.	83
6.1	Qualitative evaluation of properties that are satisfied by Batch-IG, DeepSHAP, SHAP, LIME and Gradient \times Input when considering the temporal nature of data. Namely, for calculation of attribution for all methods besides Batch-IG, we take the difference in attribution between $t + 1$ and t and determine whether the properties still hold.	91
6.2	We demonstrate attribution recovery for an instance, such that we know the ground truth. Therefore, the difference in predictions should be fully recovered by the attribution given to x^2 . We take the difference in attribution between $t + 1$ and t for example $\Phi(x^2) = \Phi(x_{t+1}^2) - \Phi(x_t^2)$	97
6.3	Table containing the BIC and AIC scores for different path based methods. From this we observe that the lowest AIC and BIC in the given synthetic datasets is given by Batch-IG, indicating superior performance in the given cases (<i>note: the step sizes for each model equate to the same with respect to the Riemann approximations across the entire path.</i>)	97
6.4	The student example corresponding to the generated explanations given in Figure 6.2.	98
6.5	The public health COVID data corresponding to the explanations given Figure 6.3.	101
6.6	RMSE comparing Batch-IG against the IG path over 5 functionally equivalent Neural Networks for predicting the path interpolation values on the Education dataset.	101

6.7	RMSE comparing Batch-IG against the IG path over 5 functionally equivalent Neural Networks for predicting the path interpolation values on the COVID dataset.	101
7.1	An overview of generative counterfactual methods and their consideration of key metrics. Here we observe of the three metrics QUCE is the method that accounts for all three.	106
7.2	Comparison of the average path uncertainty on the generated counterfactual instances. This is experimented over 100 instances from the training and testing sets of each dataset. Here we have 1000 steps (path interpolation instances) for the Riemann approximation of every path-based approach, thus effectively 100×1000 instances. Here the lower value the better. The proposed QUCE method shows superior performance on average when comparing counterfactual path-based approaches.	116
7.3	Comparison of the average reconstruction error between original instances and their generated counterfactual examples. This is experimented over 100 instances on each dataset. Here we observe that the proposed QUCE method performs best across all datasets.	117
8.1	A patient instance taken from the SEER breast cancer dataset. For illustration, we will consider Feature 7, Tumor Size to be missing in our comparison of imputation algorithms. Such missingness yields 0 feature attribution explanation.	123
8.2	Imputation results from different methods. The ground truth is Feature 7, Tumor Size with a feature value of 12. The SSI method produces the closest imputed value of 10.78. Similarly, the SSI method yields an explanation that is closest to the one computed with the feature value ground truth.	123
8.3	An overview of each dataset.	134
8.4	ICD-10 codes that form the associated datasets of the Simulacrum.	134
8.5	Example of the imputed values when compared to the ground truth for each imputation method applied to a single instance from each of the datasets. Values closer to the ground truth are better.	135
8.6	Example of SHAP values returned by the SHAP method comparing each imputation method against the ground truth on a patient instance from each of the datasets. Values closer to the ground truth are better.	135
8.7	Performance of imputation methods returned for the defected instances \mathbf{x}^d , for each instance averaged over in their respective datasets. These compared using oracle instances \mathbf{x} and the recovered instance \mathbf{x}^r . Here we observe the top 3 methods in the datasets ordered from best performing (1) to third best (3). The lower the value the better.	137

8.8	Explanation faithfulness of the imputation methods across the complete instances and recovered instances, for every instance averaged over in their respective datasets. Here we observe the top 3 methods in the datasets ordered from best performing (1) to third best (3). The higher the value the better.	138
8.9	We evaluate the runtime of the imputation algorithms evaluated in this work. Here we observe that SSI, MiCE and kNN imputation methods have the quickest run times when compared to GAIN, SoftImpute and MissForest. .	140
9.1	Non-pharmaceutical COVID Control Measures.	149
9.2	Prediction performance on the COVID dataset with four different classifiers.	150
9.3	Each patient is described with 20 features.	152
9.4	Predictions for the Lung Cancer dataset.	153
C.1	The simulacrum feature names with given description	184
D.1	Comparison of explanations returned by SHAP and Batch-IG ^R	192
D.2	We demonstrate attribution recovery for an instance, such that we know the ground truth. Therefore, the difference in predictions should be fully recovered by the attribution given to x^2 . As in the previous example, for calculation of attribution for all methods besides Batch-IG, we take the difference in attribution between $t + 1$ and t for example $\Phi(x^2) = \Phi(x_{t+1}^2) - \Phi(x_t^2)$	194
E.1	Comparison of the average sum of feature-wise reconstruction error between original instances and their generated counterfactual examples. This is experimented on 100 instances for each dataset. Here we observe that the QUCE method performs best in generating counterfactuals with minimal uncertainty across all datasets.	199
E.2	Comparison of the deletion scores for counterfactual generative methods that provide feature attribution values. This is experimented over 100 instances on each dataset. Here the lower the value the better. We observe that the proposed QUCE method performs best across a larger fraction of datasets.	201
G.1	XGBoost performance metrics where we use NonCF for training and CF for testing	213
G.2	Jaccard Index $v = 5$	215
G.3	Shared Attribution	215

List of Symbols and Acronyms

Acronyms

AGI	Adversarial Gradient Integration
AI	Artificial Intelligence
DiCE	Diverse Counterfactual Explanations
DNN	Deep Neural Network
EHR	Electronic Health Record
GAIN	Generative Adversarial Imputation Nets
GAN	Generative Adversarial Networks
GDPR	General Data Protection Regulation
IG	Integrated Gradients
LIME	Local Interpretable Model-Agnostic Explanations
MICE	Multivariate Imputation by Chained Equations
ML	Machine Learning
OoD	Out-of-Distribution
RMSE	Root Mean Squared Error
SHAP	SHapley Additive exPlanations
XAI	eXplainable Artificial Intelligence
XGBoost	eXtreme Gradient Boosting

Symbols

β	Model coefficients
χ_b	Set of instances in a time batch

δ	Distance function
γ	Represents a path between two instances
\mathbf{x}	Instance of a dataset
\mathbf{x}'	Baseline instance
\mathbf{x}^d	Defected instance
\mathbf{x}^r	Recovered instance
\mathbf{x}_c	Counterfactual instance
\mathbf{y}	Vector of labels
\mathbf{z}	Instance in a neighbourhood \mathcal{Z}
\mathcal{L}	Loss function
\mathcal{T}	Target class
\mathcal{Z}	Neighbourhood generated around an instance \mathbf{x}
∇F	Gradient of a Neural Network
\odot	Hadamard product
Φ	Feature attribution method
$\psi(\alpha)$	Mapping function to a point α on a path
τ	Hyper-parameter with a value in $[0,1]$
\mathbf{x}_t	Instance at time point t
b	Number of counterfactual instances in C
C	Set of counterfactual instances
F	Neural Network that produces a predicted probability or regression value
f	Arbitrary black-box model
g	Glass-box model for a neighbourhood \mathcal{Z}
J	Number of features
K	Number of steps in a Riemann approximation
N	Number of instances
P	Number of instances in a surrogate dataset \mathcal{Z}

R Number of iterations or runs for a process
 T Number of instances in a time batch
 X Dataset

Part I

Introduction

Chapter 1

Thesis Introduction

Contents

1.1	Motivation	3
1.2	Aims and Contributions	10
1.3	List of Publications	11
1.4	Conference/Workshop Talks	15

1.1 Motivation

The idea of replicating biological processes within machines can be traced back to the question posed by Alan Turing in 1950: “Can machines think?” [Tur50]. This query delves into the idea of recreating processes such as “thinking” and other biological interactions, implying a form of bio-mimicry such that a machine mimics or draws inspiration from biological models.

The concept of the Artificial Neural Network (ANN) is rooted in the structure of the brain. The biological brain is composed of neurons and synapses. In the realm of Machine Learning (ML), an early attempt at replication was seen in Rosenblatt’s perceptron, also known as the McCulloch-Pitts neuron. The perceptron was tailored for supervised, linearly separable binary classification tasks. However, the computational limitations required to utilise this approach hindered further progress in ANN research. This scenario changed with the advent of Feed-Forward Neural Networks (FFNN), which paved the way for the Multilayer Perceptron, this became a conceivable approach given the improvement of computation. This advancement in the architecture incorporates non-linear activations and multiple layers to the neural network. The concept of back-propagation, introduced in [RHW86], along with learning techniques like gradient descent (for further details see Appendix A), resonates with the earlier question, “Can machines think?” One could argue that the notion of “thinking” is intrinsic to the process of “learning” [Kad15].

While limitations were present with the inception of such methods, it was evident that Machine Learning (ML) was a conceivable idea, although this was not necessarily in the

form of ANNs initially, as a consequence of computational complexities and architectural constraints at the time. Progressing from this point, in 1967, the *computable* k-Nearest Neighbour (kNN) algorithm was introduced [CH67]. The term *computable* is emphasized here, as the conceptualization of the nearest neighbor classification idea loosely dates back to the “Book of Optics” (Ibn al-Haytham) [Pel14], which, as stated by [Pel14], is believed to have been written in the 1030s.

Moving ahead, the 1990s witnessed the emergence of boosting models, where the utilization of weak learners to create a more proficient learner was introduced. This concept was outlined in the paper “The Strength of Weak Learnability” [Sch90], thus laying the groundwork for contemporary ensemble and boosting methods like eXtreme Gradient Boosting (XGBoost) [CG16].

As the computational capacity evolved into the 2000s, it became evident that employing ANNs would be a viable approach for learning complex relationships and thus increased model performance for higher dimensional and more complex tasks, as the increase in computational capabilities facilitated the training of more intricate networks, leading to an increase in model complexity. From 2006 overarching advancements in AI lead to the popularisation of *deep learning*, this can be seen in [HOT06, Hin07] and the introduction of Generative Adversarial Neural Networks in 2014 [GPAM⁺14]. For further information on Machine/Deep Learning please see Appendix A.

1.1.1 An Increase in Complexity, a Decrease in Transparency

As the complexity of computers increased and the prominence of big data emerged, the adoption of more intricate systems (such as Deep Neural Networks and ensemble models) gained traction, arguable outperforming humans in numerous tasks, for example outperforming radiologists in diagnostics [DK19]. Consequently, the concept of Machine Learning (ML) stepping in to aid humans in domain-specific tasks gained substantial appeal.

In line with this, the introduction of the General Data Protection Regulations (GDPRs) “*right to an explanation*”, outlined the necessity for explainability [SP17]. As ML models evolve and grow in complexity, they often exhibit enhanced accuracy. Consequently, this escalated complexity leading to an improved accuracy often hampers the innate interpretability of the ML models. This has paved the way for the emergence of a subset within Artificial Intelligence (AI) known as eXplainable Artificial Intelligence (XAI). In layman’s terms, XAI can be regarded as a means to comprehend the underlying workings of a machine learning model rather than delving into the intricacies of the data. For the purpose of this thesis, it’s important to establish this clear demarcation. Essentially, XAI serves as a method to unravel the “thought process” of the machine. Thus, it endeavors to tackle the question:

“How can we make black-box models explainable?”

In this thesis, the above question is answered from a *modelling, data* and *user* perspective. Explainability is commonly given in the form of feature-attribution, which aims to answer the question:

“How does each feature contribute towards a prediction?”

Feature-attribution is often considered throughout the body of this thesis, although this is not the only form of explainability. Feature-attribution provides an intuitive way to observe how a given prediction is obtained. This is achieved by analysing how each feature of a predicted instance influences the given prediction. Therefore, enabling transparency of an predicted outcome.

Producing transparent models is crucial in the field of medicine. It is clear, such a critical field must deploy transparency *if* AI is to be used in any process (e.g. decision-making, analytics). Therefore, this thesis first provides the reader with a comparison of state-of-the-art XAI techniques for medical data. The comparison, provided in the body of this thesis, was the first study (to the knowledge of the author) of this kind, exploring EHR tabular data with state-of-the-art XAI methods. With the identified dissonance from our comparison of XAI models, it elucidates open research opportunity for the development of new XAI methods.

In Table 1.1, we provide an overview of papers with their associated research question and XAI method(s). The set of publications provided in this table are accumulated between January 2020 and June 2023. Here I provide a brief overview of XAI methods used in recent healthcare research.

Table 1.1: The set of papers exploring how XAI has been used in medicine. This is accompanied by the XAI methods used, the medical research question proposed, the year of publication and the type of explanation method. Here we observe a wide range of applications often utilise the same set of XAI methods with LIME and SHAP being common in application.

XAI in EHR Paper	XAI Method(s)	Research Question	Year	XAI Type
[DFB ⁺ 21]	SHAP, LIME, Anchors	Mortality prediction for Lung Cancer patients	2021	Model-Agnostic
[DFS22]	PALE, SHAP, LIME, Log Reg	Survival time prediction for different cohorts of cancer patients	2022	Model-Agnostic, Model-Intrinsic
[MTvMH ⁺ 21]	SHAP	Breast cancer survival prediction	2021	Model-Agnostic
[KED ⁺ 21]	SHAP, LIME	Lung cancer life expectancy and COVID restriction measures	2021	Model-Agnostic
[TVA ⁺ 22]	Log Reg, EBM, SHAP	COVID-19 diagnosis through blood test variables	2022	Model-Agnostic, Model-Intrinsic
[LKO ⁺ 20]	xAI-EWS	Predicting acute illness for sepsis, acute kidney injury and acute lung injury	2020	Model-Specific
[LGZ ⁺ 21]	Attention	Identify chronic cough patients	2021	Model-Specific
[TBL ⁺ 22]	Attention	ICD Classification	2021	Model-Specific
[CFB ⁺ 21]	MCE	Unsupervised Heart patient clustering	2021	Model-Specific
[CLE ⁺ 21]	SHAP	Forecasting adverse surgical events	2021	Model-Agnostic
[NCC ⁺ 21]	SHAP, Log Reg	Predicting adverse outcomes of COVID-19 patients	2021	Model-Intrinsic, Model-Agnostic
[JGVM ⁺ 20]	SHAP, LIME	Breast cancer survival prediction	2020	Model-Agnostic
[TRO22]	SHAP	Predicting falls in the older population	2022	Model-Agnostic
[EVS ⁺ 22]	LIME	Predicting of major adverse cardiovascular events	2022	Model-Agnostic
[RTVA22]	TREPAN	Survived and Failed Kidney Transplant prediction	2022	Model-Agnostic
[WKSP22]	SHAP	Explaining early detection of health change predictions	2022	Model-Agnostic
[DDDV22]	LIME, SHAP	ICD Classification for Gastrointestinal Discharge	2022	Model-Agnostic
[WGGP23]	DeepSHAP	Medical Event for Heart Failure Patients	2023	Model-Specific

Regarding the agglomeration of research papers are displayed in Table 1.1, we follow by observing the number of publications per year, illustrated in Figure 1.1. Similarly, the XAI methods used in application to EHRs are shown in Figure 1.1. It is apparent that feature-attribution methods are most prominent in XAI for medicine in recent years. Although, inherently interpretable methods such as Logistic Regression are still seen, diversification in research papers is clear, as XAI methods are used in parallel to inherently interpretable models. Therefore, this motivates the comparisons of explanations presented in the body of this thesis.

1.1.2 Increasing Explainability for Electronic Health Records

Feature-attribution methods are a common form of explanation representation. Whilst it is clear that explanations are valuable, there is a clear disagreement problem that we highlight in Chapter 3. Therefore, we see aberrations in application, and due to the field of XAI being recent, there is a lack of ground truth, and no clear formal properties to adhere to, having only recently been explored and proposed. Thus it is clear, there exists no definition for a correct explanation.

XAI gained large traction posterior to the proposal of Local Interpretable Model-Agnostic Explanations (LIME) [RSG16]. LIME innovated the area of XAI with the idea of black-box approximation, informally aiming to approximate the black-box decision boundary with a simple local model. The authors of LIME use a linear regression model as to approximate the decision boundary and extract coefficients to explain a local instance with in a neighbourhood (see Chapter 2.2.1 for details). LIME poses limitations of linearity with the local model [BHLL20] as the assumption of local linearity reduces model accuracy. This, a similar issue as to what occurs globally with standard linear regression, but on a smaller scale, which leads to the use of more complex methods that are difficult to interpret. Therefore, this thesis introduces the idea of Polynomial Adaptive Local Explanations (PALE), building upon the LIME framework to construct adaptive polynomial models, in order to minimise error to best fit each instance. Inherently, in the scope of medicine, one will achieve instance specific explanations, that are more faithful to the black-box model (see Chapter 4).

A consequence of model-agnostic approaches is the reliance on decision boundary approximation, this could exhibit biases in the XAI method itself, therefore looking at model-specific methods may provide more reliable explanation, accentuating that bias is more likely to originate from the model or data itself, as after all XAI is about explaining the model, from hereinafter I build upon the Integrated Gradients (IG) framework to produce explanations for specific data representations and clinical questions.

In application to healthcare, a commonly posed question is presented in the form of counterfactual thinking. Often methods for causal inference such as the ATE is used to determine interventions. Naturally, assessing causal effects is inherently limited as one cannot evaluate a patient in two states simultaneously, thus A/B testing with control groups is used to determine the ATEs of interventions. Unfortunately, the control groups tend to not be large in participant size, and thus counterfactual explanations can remove the need of large control groups at one time, and instead use existing

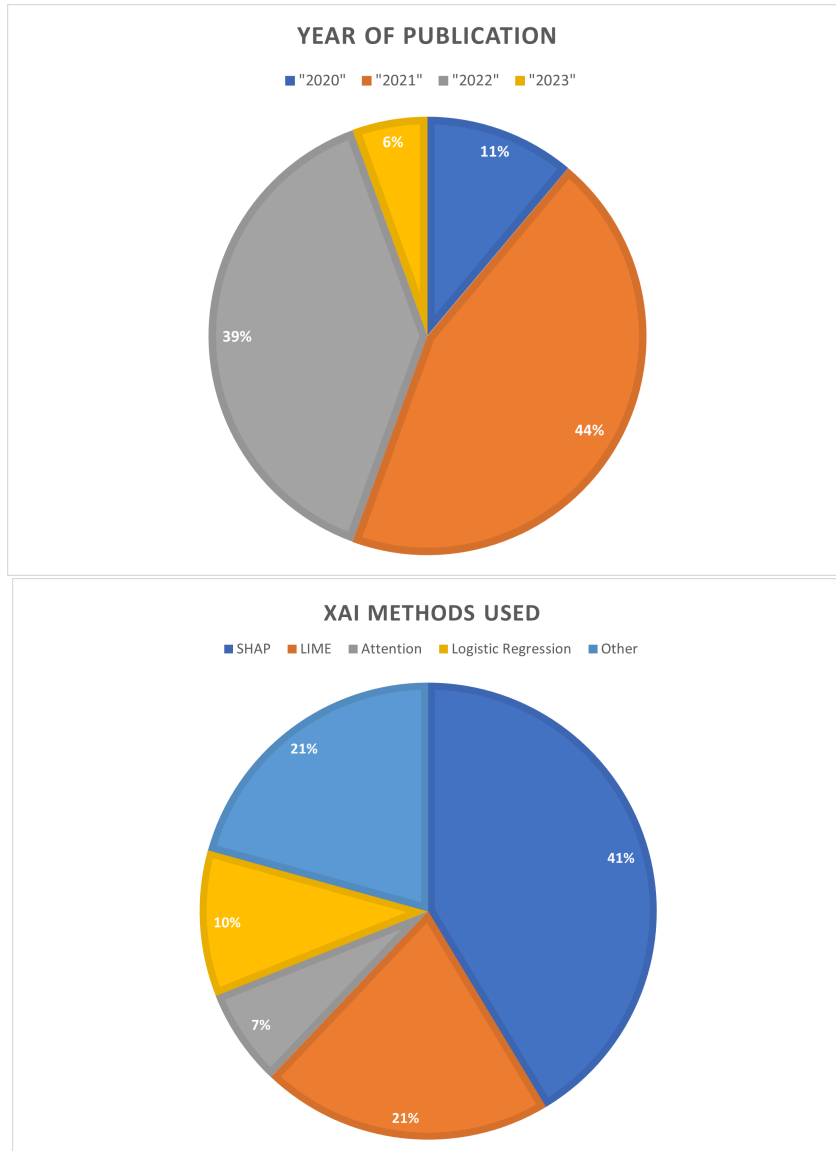


Figure 1.1: Illustration of the years of publication and the XAI methods utilised for the papers in Table 1.1.

knowledge of treatment outcomes from patient data eliminating the need for further testing and evaluate the patient in two states given a different intervention. Similarly, counterfactual explanations are not limited by discrete data and can instead modify continuous values willingly. Feature-attribution is a common form of XAI but is not often utilised with counterfactual XAI methods, instead counterfactuals are normally presented in the form of counterfactual examples, but evaluating effects of different elements through feature-attribution seems a promising approach and thus I propose the Counterfactual-Integrated Gradients (CF-IG) method.

Transcending this, one can consider how both data can be represented in EHRs, and what questions can be asked with respect to the data. Temporal data representation is common in EHRs, but is often not addressed by existing methods [SSV21]. Often a patient will have more than one visit to a healthcare provider during specific treatments (e.g. cancer treatment), in doing so one may be interested to see how the treatment has effected the outcome of the patient. Therefore, I build upon the IG framework and introduce a new XAI method named Batch-IG to address such question. The idea of Batch-Integrated Gradients is proposed and analysed empirically, where the notion of time is introduced into the proposed explainable method (see Chapter 6 and Appendix D). Naturally, this method utilises the concept of the line integral, but linear interpolations can lead to out-of-distribution paths. Informally, this can make gradients unreliable and the linear uniform interpolation path unlikely. Thus, the proposed Batch-Integrated Gradients traverses more *probable* space, supported by a thorough theoretical analysis and comparison against state-of-the-art explainers when measured against properties (see Chapter 6).

To further extrapolate on the path-based explanation formulation, the paths utilised in generating explanations can suffer from irregular gradients due to out-of-distribution interpolation, to formally address this concern, I propose the Quantifiable (Path-Based) Uncertainty Counterfactual Explanations (QUCE) method. The details of this approach are described in Chapter 7.

Extending from the development of XAI from a modelling perspective is the consideration of the data itself, a consequence of theoretical properties observed for feature-attribution methods is the property of “*missingness*”. This is observed, for the state-of-the-art methods LIME [RSG16], SHAP [LL17] and IG [STY17]. This is further described in Chapter 8, where I propose a data oriented approach to respond to the missingness property and the associated consequences on explainability.

Finally, to make XAI more accessible we introduce the tool ExMed [KED⁺21], this in turn improves explainability from the point of view that an increased user base can then access explanations for themselves with limited expertise in XAI.

Hereinafter, I propose that the aforementioned can be discretized into the following taxonomy:

- *Model Perspective*: The focus of producing explanations from a given machine learning model.

- *Data Perspective*: The focus of increasing the explainability of machine learning on models by improving the quality of data.
- *User Perspective*: The user-centric approach towards making explanations more accessible.

This taxonomy is outlined in the body of this thesis as follows:

1. Enhancing Explainability a Model Perspective: Chapter 4, Chapter 5, Chapter 6 and Chapter 7.
2. Enhancing Explainability a Data Perspective: Chapter 8.
3. Enhancing Explainability a User Perspective: Chapter 9.

The aims and contributions of this thesis are summarised in Section 1.2.

1.2 Aims and Contributions

The thesis proposes a collection of new methods that build upon and utilise state-of-the-art XAI approaches. Briefly summarising Section 1.1, this thesis aims to improve the quality and diversity of explanations produced. It first provides a comparison of state-of-the-art XAI approaches, where there is identification of a disagreement problem with returned explanations, and thus through evaluating the architecture of the XAI approaches there are identifiable limitations.

The contributions of this thesis are listed below. Contributions 1-4 and 7 are already published, see Section 1.3.

1. A novel contribution, providing a variety of comparisons for explanations that are given on Electronic Health Records. My contribution here illustrates the disagreement of XAI methods in application to medical records. [See publications 1, 9, and poster presentation 10][Chapter 3 and Appendix G of this thesis]
2. A novel model-agnostic eXplainable Artificial Intelligence method. The Polynomial Adaptive Local Explanations method introduced enhances the linear limitations of a state-of-the-art method by proposing a polynomial expansion to the original linear model to better approximate the black-box. The method allows for production of explanations whilst increasing local accuracy. The method utilises instance specificity by optimising for each instance independently as a novel method to produce patient specific explanations. [See publication 2][Chapter 4]
3. A novel approach Counterfactual-Integrated Gradients is introduced for generating counterfactual explanations, showcasing an improved performance for the proposed method. Here we utilise the line integral formulation for both closer and more consistent explanations. The explanation consistency and proximity are measured against state-of-the-art counterfactual explainers. [See publication 3][Chapter 5]

Here I also released a public code repository available at:

https://github.com/jamie-duell/Counterfactual-Integrated_Gradients

4. A novel approach to explaining temporal data. The model is first conceptualised, then further optimised and both theoretically and empirically analysed. This approach holds more theoretic guarantees in temporal application than existing state-of-the-art XAI methods. [See publications 4 and 5][Chapter 6 and Appendix D]
5. A novel approach for generating counterfactual explanations and examples. The method QUCE is work in progress for developing minimally uncertain path-based explanations and counterfactual examples. [Chapter 7 and Appendix E] This approach will be available posterior to the thesis submission at - <https://github.com/jamie-duell/QUCE>.
6. A novel approach for single and multiple value data imputation inspired by eXplainable Artificial Intelligence methods, aiming to increase the fidelity of imputations when compared against current state-of-the-art imputation methods. This approach unifies the relationship between imputation and explainability to account for the missingness property, whilst aiming to ensure that the imputation method itself is interpretable.[Chapter 8 and Appendix F] The library is available at: <https://pypi.org/project/surrogate-set-imputer/>
7. A novel tool to enable ease of accessibility for data exploration, machine learning and XAI method applications. [See publication 8][Chapter 9] The tool is available at - <https://github.com/983046/ExMed>.

1.3 List of Publications

Here we provide overview of published and submitted works that directly contribute to this thesis. From publications 1-7 and 9 of the listed contributions I am the main author and thus contributed to all the writing of the first draft manuscripts, experiments, code, methods, theory and evaluations. Dr. Xiuyi Fan, Dr. Monika Seisenberger and Dr. Hsuan Fu played a pivotal part in the proof reading and correcting erratum within the manuscripts. Dr. Xiuyi Fan provided the concept of publication 1 with Prof. Shangming Zhou having a large part in amending the writing for the introduction for the manuscript, where this publication was my first exposure to the field of XAI and research publication writing.

1. **A Comparison of Explanations Given on Electronic Health Records [DFB⁺21]**
Jamie Duell, Xiuyi Fan, Bruce Burnett, Shangming Zhou. IEEE-EMBS International Conference on Biomedical and Health Informatics (IEEE BHI 2021, Full Paper).

Here the main contributions of the manuscript is a comparison of XAI methods for Lung Cancer patients, the XAI field was somewhat recently emerging at the time of publication and thus state-of-the-art methods had yet been compared in the medical context for tabular EHRs for Lung Cancer patients. Therefore, the identification of the disagreement problem for XAI methods was crucial.

2. **Towards Polynomial Adaptive Local Explanations for Healthcare Classifiers.** [DFS22] Jamie Duell, Xiuyi Fan, Monika Seisenberger. International Symposium on Methodologies for Intelligent Systems (ISMIS 2022, Full Paper).

This manuscript contributes to the improvement of accuracy of local linear surrogate models as seen in LIME. Most of the manuscripts at the time of this publication focused on the modifications of the LIME neighbourhood function. Here, I instead look at extending the linear model whilst still being able to derive explanations. The contributions of this publication are twofold:

- Provide a more accurate local explanation model by extending the LIME method with local polynomial models.
- Adapt the polynomial models to each local neighbourhood such that we get patient specific explanations with respect to a corresponding neighbourhood.

3. **Counterfactual-Integrated Gradients: Counterfactual Feature Attribution for Medical Records** [DSF23] Jamie Duell, Monika Seisenberger and Xiuyi Fan, IEEE International Conference on Bioinformatics and Biomedicine 2023 Workshop on Machine Learning and Artificial Intelligence in Bioinformatics and Medical Informatics (IEEE BIBM 2023, MABM Workshop Paper)

Here the contributions of the manuscript are threefold.

- In the application we proposed a simple Nearest Counterfactual Neighbour algorithm and provide modifications to the integrated gradients framework to utilise the counterfactual examples, this is an extrapolation on the effects of changing the baseline and target and how one can re-frame research questions from this approach. The explanation is extracted by stopping interpolations at the decision boundary, providing both a counterfactual example and counterfactual feature attribution explanation, this will naturally provide counterfactual instances closer than any other generative method. The code associated with this publication is provided at https://github.com/jamie-duell/Counterfactual-Integrated_Gradients.
- On the theoretical side, we provided a theoretical analysis of property satisfiability and the introduction of new properties for counterfactual feature attribution methods. The benefit of doing so, is to pave the way for further work in the thesis and provide a generalised formal evaluation to determine the strength of such approach.
- In an empirical evaluation we provide evidence for the theoretical claims with experimentation.

4. **Batch Integrated Gradients: Explanations for Temporal Electronic Health Records [DFFS23]** Jamie Duell, Xiuyi Fan, Hsuan Fu, Monika Seisenberger. International Conference on Artificial Intelligence in Medicine (AIME 2023, Short Paper)

This short paper contributed towards the (relatively) new interests of the academic community of XAI focusing on temporal and time-series explainability, more specifically dealing with irregular time intervals. This was introduced as a simple modification of the integrated gradients method. The experiments aim to illustrate to the medical community the usability of temporal explanations and to emphasise the future direction of my own research.

5. **A Formal Introduction to Batch Integrated-Gradients for Temporal Explanations [DSZ⁺23]** Jamie Duell, Xiuyi Fan, Tianlong Zhong, Hsuan Fu, Monika Seisenberger. IEEE International Conference on Tools for Artificial Intelligence (ICTAI 2023, Full Paper)

The contributions of this manuscript are threefold:

- The extension of the prior paper [DFFS23], here we show how one can modify the temporal path, and show that various theoretical properties of XAI hold for Batch-IG [DFFS23].
- The experimental work shows how properties are indeed satisfied and thus supports the theoretical claims of the Batch-IG method.
- The method was experimented on two real world datasets illustrating explanations that can be produced.

6. **QUCE: The Minimisation and Quantification of Path-Based Uncertainty for Generative Counterfactual Explanations [Work in Progress - To be submitted in April to ECAI 2024]** Jamie Duell, Monika Seisenberger, Hsuan Fu and Xiuyi Fan.

Here the contributions of the manuscript are threefold:

- We proposed a new method that generates counterfactual examples that quantify uncertainty.
- We developed a new way to minimise uncertainty for path-based methods to produce more reliable explanations.
- We illustrated that the newly proposed method outperforms existing state-of-the-art methods in minimising uncertainty.

7. **Explaining Incomplete Electronic Health Records [Submitted to the AI in Medicine Journal in November 2023]** Jamie Duell, Xiuyi Fan and Monika Seisenberger.

Here the contributions of the manuscript are threefold:

- We proposed a set of properties that should trivially be satisfied by imputation methods. Here we aim to unify the inherent relationship between explainability and imputation. This relationship is evident across a broad range of XAI methods that satisfy certain properties (in this manuscript we focus on the property of *missingness*).
- We developed a new imputation method inspired by local surrogate prediction models such as LIME, and provide details of implementation.
- The new method was evaluated for both property satisfiability (theoretically and empirically) whilst similarly outperforming a number of state-of-the-art imputation methods in controlled single-value and multiple-value imputation experiments. The library for this method is provided at <https://pypi.org/project/surrogate-set-imputer/>.

8. **ExMed: An AI Tool for Experimenting Explainable AI Techniques on Medical Data Analytics [KED⁺21]** Marcin Kapcia, Hassan Eshkiki, Jamie Duell, Xiuyi Fan, Shangming Zhou, Benjamin Mora. IEEE International Conference on Tools for Artificial Intelligence (ICTAI 2021, Short Paper).

The key contributions of this manuscript are twofold:

- We introduce a comprehensive XAI toolkit (ExMed) designed for domain experts, encompassing data analytics, data preprocessing, data visualization, ML application, and XAI application.
- We demonstrate the utility of ExMed by applying the toolkit to two real case studies: one focused on COVID and the other on lung cancer.

Personal Contribution: As a co-author I produced the initial code for the XAI methods that were later used in the tool. I also provided the pre-processed data for the cancer dataset used in the study. I wrote the first draft of the cancer case study section and contributed corrections to the manuscript throughout.

The ExMed tool can be downloaded via: <https://github.com/983046/ExMed>. Details on the usage are given in the form a small video tutorial.

9. **A Comparison of Global Explanations Given on Electronic Health Records [DFS23]** Jamie Duell, Xiuyi Fan, and Monika Seisenberger 2023 International Conference on Intelligent Autonomous Systems, Suwon, Korea (IAS-18 2023, Full Paper).

The contributions of this manuscript are threefold:

- We assess the quality of counterfactual instances using predictive models.
- We introduce metrics for comparing XAI methods.
- We employ these metrics to quantify the similarity of XAI methods.

10. **Evaluating XAI Explanations on Electronic Health Records [Poster Presentation]** Wei Feng Sim, Jamie Duell and Xiuyi Fan. International Conference on AI in Medicine (AiM 2023).

The contributions of this manuscript are twofold:

- We present metrics for comparing XAI and counterfactual methods.
- We utilize these metrics to measure the similarity between XAI and counterfactual methods.

Personal Contribution: As a co-author I designed the experiments and provided changes to the final manuscript.

1.4 Conference/Workshop Talks

1. **A Comparison of Explanations Given on Electronic Health Records**
Jamie Duell, IEEE BHI Conference 2021, Athens, Greece.
2. **On Explainable Artificial Intelligence for Medical Diagnostics and its Potential Scope for Future Development.**
Jamie Duell, British Colloquium on Theoretical Computer Science, Special session on Explainable AI 2022, Swansea, Wales.
3. **Towards Polynomial Adaptive Local Explanations for Healthcare Classifiers.**
Jamie Duell, ISMIS 2022, Cosenza, Italy.
4. **Rule-PSAT: Relaxing Rule Constraints in Probabilistic Assumption-Based Argumentation [Fan22]**
Jamie Duell, International Conference on Computational Models of Argument 2022, Cardiff, Wales [On behalf of Xiuyi Fan].
5. **Enhancing the Explainability of Electronic Health Record Predictions**
(Jamie Duell, Society for the Study of Artificial Intelligence and Simulation of Behaviour Workshop on Explainability and Transparency in AI 2022, Swansea, Wales.
6. **Introduction of the Explainable Artificial Intelligence models using Python: LIME from Scratch**
Jamie Duell, 2023, Quebec, Canada.
Link: <https://www.fsa.ulaval.ca/evenements/ateliers-ia-explicable-1/>
Interactive Notebook: <https://tinyurl.com/4vdh8cdp>
GitHub Code: <https://github.com/jamie-duell/XAI-Workshop>
7. **Towards Explainable Artificial Intelligence: Batch-Integrated Gradients and Solutions to Missingness.**
Jamie Duell, AIMLAC CDT AI Conference 2023, Swansea, Wales.

8. **Batch Integrated Gradients: Explanations for Temporal Electronic Health Records.**

Jamie Duell, AIME 2023, Portoroz, Slovenia.

9. **A Formal Introduction to Batch Integrated-Gradients for Temporal Explanations.**

Jamie Duell, ICTAI 2023, Atlanta, United States.

Part II

Background

Chapter 2

Explainable Artificial Intelligence

Contents

2.1	Introduction	19
2.2	Feature Attribution	21
2.3	Counterfactual Explanations	29
2.4	Conclusion	30

2.1 Introduction

eXplainable Artificial Intelligence (XAI) aims to go beyond the black-box limitations of Artificial Intelligence (AI) methods. Traction towards the field of XAI became prominent upon the introduction of Local Interpretable Model-Agnostic Explanations (LIME) [RSG16]. Figure 2.1, provides an illustration of the search-term prevalence of: “*Explainable AI*” and “*Explainable Artificial Intelligence*” respectively. These insights are extracted from Google Trends¹. Here we observe that the XAI related search terms increased greatly, posterior to the publication of the LIME paper in 2016, this began to rise rapidly after the release of SHapley Additive exPlanations (SHAP) [LL17] in 2017.

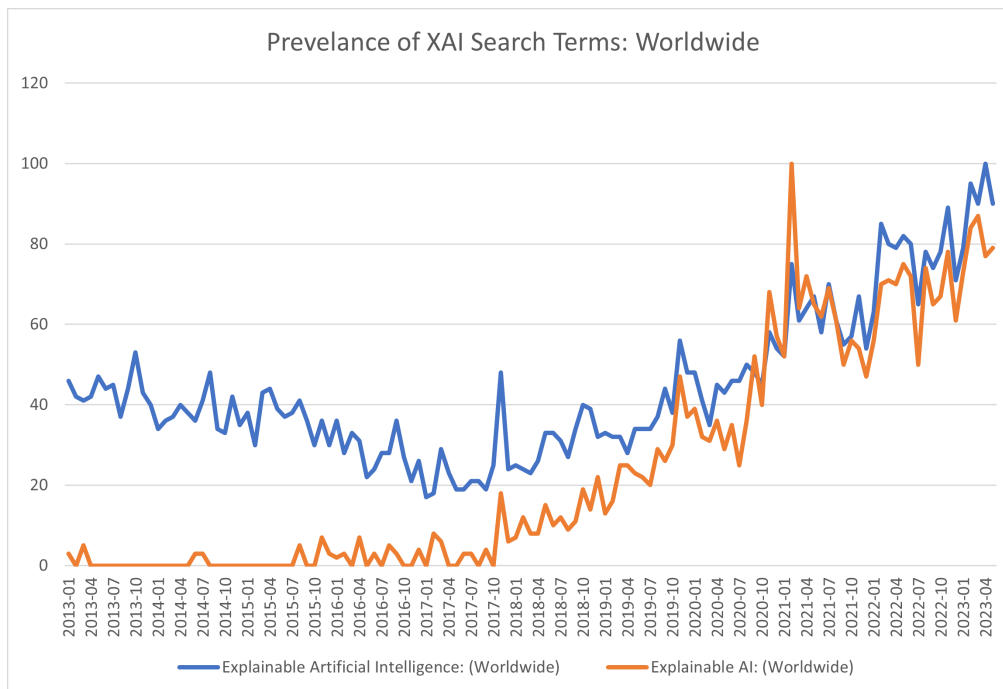
In this chapter, I provide a taxonomy for XAI methods. This taxonomy offers insights into the various ways that XAI methods can be categorized. Specifically, I introduce the *scope* and *model type* of XAI methods. Such a taxonomy enables us to address the following questions:

1. Model type - How is the model constructed? (see Section 2.1.1)
2. Scope - what does the explanation focus on? (see Section 2.1.2)

Following this, in sections 2.2 and 2.3 I provide a general overview of the XAI methods that are utilised throughout the body of this thesis.

¹<https://trends.google.com/trends/>

Figure 2.1: Prevalence of the search terms “Explainable Artificial Intelligence” and “Explainable AI”, given between the years 2013-2023. These results are extracted from Google Trends. Empirically, there is evidence of an increase in the search terms post 2016.



2.1.1 XAI Model Type

XAI models can be disseminated into model type subclasses. Here, I consider three forms of model types, these being: *model-agnostic*, *model-specific* and *model-intrinsic* approaches, these can be defined as:

- **Model-Agnostic** methods provide an explanation for any black-box model f , and is thus able to explain predictions obtained by any model regardless of the architecture.
- **Model-Specific** methods aim to select a given black-box model f and produce an explanation catered to the model architecture of f . Therefore, this surrounds the manipulation or extraction of explanations from specific AI architectures, such that reasoning is derived w.r.t the model.
- **Model-Intrinsic** methods are inherently interpretable implying there is an observable cause and effect given by a model f , examples include coefficients (linear regression), odds-ratios (logistic regression) or rules (decision trees). An analogous term often used for model-intrinsic methods is *glass-box* models (see Appendix A for architecture details).

Within the subset of XAI models, these different model types are designed to provide explanations in various forms.

2.1.2 Scope

Upon consideration for the approaches of explainable methods, there is the premise of *scope*, that describes the focus for explanations. In the form of feature-attribution, the definitions for global and local explanations are given as:

1. **Local Explanations** are explanations that focus on an independent instance with a dataset. This is often explaining the prediction for a single instance, for example the LIME method [RSG16] focuses on explaining a single instance.
2. **Global Explanations** focus on an abstract view of explainability that generalises the the entire model, for example, one could consider the SHAP method [LL17] that agglomerates the local explanations for each instance within a dataset and by taking the average, one can describe the expected outcome of a model whilst considering all instances, or simply accumulating all local instances visually.

2.2 Feature Attribution

The feature attribution approach for XAI aims to evaluate, given an instance how each feature contributes towards a given prediction. Feature attribution often illustrates this by quantifying the importance of each feature, such that each features corresponding importance magnitude reflects the importance of each feature.

2.2.1 Local Interpretable Model-Agnostic Explanations

Local Interpretable Model-Agnostic Explanations (LIME) introduced in [RSG16], explores the approximation of black-box decision bounds with respect to a local instance. Intuitively, the LIME method aims to produce a convoluted space around an instance to explain. This convoluted space is the result of perturbations around the point to explain. To obtain an explanation for a given instance locally, the neighbourhood bound aims to restrict the influential features to those near the instance to explain. The use of a simple model which is inherently interpretable is applied to this local neighbourhood, then an explanation is extracted with respect to the point of interest (see Figure 2.2). Formally, given a dataset $X = \langle \mathbf{x}_1, \dots, \mathbf{x}_N \rangle \in \mathbb{R}^{N \times J}$, LIME aims to explain a single $\mathbf{x} \in X$ by producing a “local surrogate data set” \mathcal{Z} such that $\mathcal{Z} \sim \mathcal{N}(\mathbf{x}, \sigma^2)$ is taken from a Gaussian distribution or \mathcal{Z} can also be taken from a uniform distribution, where we obtain $\mathcal{Z} = \langle \mathbf{z}_1, \dots, \mathbf{z}_P \rangle \in \mathbb{R}^{P \times J}$. \mathcal{Z} contains a convoluted area of P perturbed samples surrounding an instance to explain \mathbf{x} . The LIME method is defined as:

$$\text{LIME}(\mathbf{x}) = \arg \min_{g \in G} \mathcal{L}(f, g, \pi_{\mathbf{x}}) + \Omega(g) \quad (2.1)$$

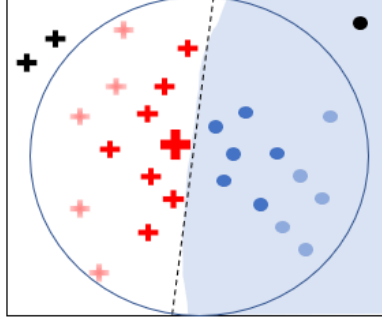


Figure 2.2: Simple illustration for the intuition behind the LIME method to explain the large red cross. Here, we let the circle define the neighbourhood. The background colour depicts the black-box classifier. The dashed line illustrates the local linear model. The colour saturation depicts the importance of each feature (bold has a stronger weight associated). The black points illustrate instances that are outside of the weighted neighbourhood.

here the loss function \mathcal{L} is used to determine the fidelity of the local linear model g from a set of interpretable models G , the fidelity of g is measured with respect to the black-box model f . This is achieved by minimising the loss function, to which end one obtains a model that is “locally faithful” when $\mathcal{L}(g, f, \pi_{\mathbf{x}}) = 0$. The Ω term is a regularization term that is used to penalise the coefficient vector of the linear model. The loss function \mathcal{L} is defined as:

$$\mathcal{L}(g, f, \pi_{\mathbf{x}}) = \pi_{\mathbf{x}}(\mathbf{z})(f(\mathbf{z}) - g(\mathbf{z}))^2. \quad (2.2)$$

For tabular data, the weighted neighbourhood for LIME is given by $\pi_{\mathbf{x}} = \exp(-\frac{\delta(\mathbf{x}, \mathbf{z})^2}{\lambda^2})$, where the kernel width $\lambda^2 = 0.75\sqrt{J}$ and distance function $\delta : \mathbb{R}^J \times \mathbb{R}^J \rightarrow \mathbb{R}$. Informally, the use of this weighting function to the linear equation states that; the data points \mathbf{z} , that are further away from \mathbf{x} in the neighbourhood \mathcal{Z} , are given less importance (weighting) than those that are closer to \mathbf{x} .

To minimise $\mathcal{L}(g, f)$, let $f(\mathbf{z}) = \mathbf{y}$ and solve for β , where β here is the coefficients (weights) for the linear model. Thus, in matrix form we have the solution of the *normal equation* which is given by the following derivation:

$$\begin{aligned} \mathcal{L}(g, f) &= \frac{1}{P} \sum_{\mathbf{z} \in \mathcal{Z}} (f(\mathbf{z}) - g(\mathbf{z}))^2 = (\mathbf{y} - \mathcal{Z}\beta)^2 \\ &\implies (\mathbf{y} - \mathcal{Z}\beta)^T (\mathbf{y} - \mathcal{Z}\beta) \\ &\implies ((\mathbf{y})^T - (\mathcal{Z}\beta)^T) (\mathbf{y} - \mathcal{Z}\beta) \\ &\implies (\mathbf{y})^T \mathbf{y} - 2(\mathcal{Z}\beta)^T \mathbf{y} + (\mathcal{Z}\beta)^T \mathcal{Z}\beta \\ &\implies \mathbf{y}^T \mathbf{y} - \underbrace{2\mathcal{Z}^T \beta^T \mathbf{y}}_{\text{term 1}} + \underbrace{\mathcal{Z}^T \beta^T \mathcal{Z}\beta}_{\text{term 2}} \end{aligned}$$

Thus, given the expanded form, the derivative with respect to β is taken, namely looking at term 1 and term 2. The loss is then set to zero with respect to the coefficients β :

$$\begin{aligned}\frac{\partial L}{\partial \beta} &= -2 \frac{\partial(\mathcal{Z}^T \beta^T \mathbf{y})}{\partial \beta} + \frac{\partial(\mathcal{Z}^T \beta^T \mathcal{Z} \beta)}{\partial \beta} \\ &\implies \frac{\partial L}{\partial \beta} = -2\mathcal{Z}^T \mathbf{y} + 2\mathcal{Z}^T \mathcal{Z} \beta \\ &= \frac{\partial L}{\partial \beta} = -2\mathcal{Z}^T \mathbf{y} + 2\mathcal{Z}^T \mathcal{Z} \beta = 0 \\ &\implies \frac{\partial L}{\partial \beta} = \mathcal{Z}^T \mathbf{y} + \mathcal{Z}^T \mathcal{Z} \beta = 0\end{aligned}$$

therefore, we have the following equality:

$$\mathcal{Z}^T \mathcal{Z} \beta = \mathcal{Z}^T \mathbf{y},$$

given $(\mathcal{Z}^T \mathcal{Z})$ is invertible, multiplying both sides of the equation by $(\mathcal{Z}^T \mathcal{Z})^{-1}$ gives:

$$\beta = (\mathcal{Z}^T \mathcal{Z})^{-1} \mathcal{Z}^T \mathbf{y}$$

Expanding on this derivation, one can include the weight (neighborhood) term, producing the solution for the weighted normal equation for the coefficients of the local neighbourhood of \mathbf{x} :

$$\beta = (\mathcal{Z}^T \pi_{\mathbf{x}} \mathcal{Z})^{-1} \mathcal{Z}^T \pi_{\mathbf{x}} \mathbf{y} \quad (2.3)$$

LIME incorporates the least absolute shrinkage and selection operator (LASSO) regularization for returning up to k features, ordered by the magnitude of the coefficient feature dimensions. Thus, $\Omega(g)$ can be represented as: $\sum_{j=1}^J |\beta^j|$. In matrix form, this is represented as the l_1 norm, such that any l_p norm can be represented as $\|\beta\|_p = \left(\sum_{i=1}^N |\beta_i|^p \right)^{\frac{1}{p}}$. Ordering the regularized coefficient vector by dimension magnitude and restricting up to k returned features produces the LIME explanation. LIME implementation² can vary according to the problem domain.

Due to the arbitrary nature of the neighbourhood function in its definition, which was primarily assigned due to the empirical performance in explanations, there have been many expansions of LIME aiming to address concerns of the neighbourhood, examples of this include, Deterministic-LIME (DLIME) [ZK21] and Stabilized-LIME (S-LIME) [ZHW21]. The DLIME approach utilises hierarchical clustering and fits linear models over the defined clusters. When a new sample is introduced, the cluster that the nearest neighbour of the new sample is used the background dataset for the linear model and associated explanation. S-LIME on the otherhand, provides a solution to determine the number of perturbed samples that are required for a background set that employs enough confidence for the linear model.

²I provide a working tutorial for the implementation of a simple variant of LIME from scratch, where I mitigate LASSO for simplification of the model in: <https://github.com/jamieduell/XAI-Workshop> (resp. <https://tinyurl.com/4vdh8cdp>) as presented in Laval University (see <https://www.fsa.ulaval.ca/evenements/ateliers-ia-explicable-1/>).

2.2.2 SHapley Additive exPlanations

The SHapley Additive exPlanations (SHAP) method was introduced in [LL17]. Informally, the SHAP method aims to determine the marginal contribution for each feature of an instance with respect to a prediction. From a game theoretic perspective, the goal is to assess the individual contributions of each 'player' in a game towards a given 'reward'.

In the context of ML, we can consider the players to be features and the reward to be a prediction. The contribution is calculated by considering all possible coalitions of features and the subsequent attribution towards the prediction. SHAP is an additive feature attribution method that utilises the game theoretic approach of Shapley values. The attribution of a feature through the means of Shapley values is shown in Eq. 2.4

$$\phi^j(f, x_i^j) = \sum_{\mathbf{z} \subseteq \mathbf{x}} \frac{|\mathbf{z}|!(J - |\mathbf{z}| - 1)!}{J!} [f(\mathbf{z}) - f(\mathbf{z} \setminus j)], \quad (2.4)$$

where \mathbf{z} is taken over all possible subsets of \mathbf{x} to attribute the difference in \mathbf{x} with \mathbf{z} and without feature j (denoted as $\mathbf{z} \setminus j$). In reference to equation 2.1 the SHAP method aims to construct a version of a local surrogate model that successfully recovers Shapley values locally. A method designed to approximate Shapley values in a model-agnostic manor is named: Kernel SHAP. Kernel SHAP modifies the regularization term Ω and neighbourhood function $\pi_{\mathbf{x}_i}$, reducing to the form:

$$\Omega(g) = 0, \quad (2.5)$$

$$\pi_{\mathbf{x}_i} = \frac{(J - 1)}{\binom{J}{|\mathbf{z}|_0!(J - |\mathbf{z}|_0)!} |\mathbf{z}|_0(J - |\mathbf{z}|_0)}, \text{ for } 0 \leq |\mathbf{z}|_0 \leq J. \quad (2.6)$$

Simple intuition for understanding Shapley Value and feature attribution can be given under the assumption of *independence*, where independence can be simply defined as:

Independence: Assuming two events A and B are independent, then the probability of both events A and B happening are equal to the product of independent probabilities for A and B . Thus,

$$P(A \cap B) = P(A) \cdot P(B)$$

Shapley values can be accurately calculated from a linear model such that, given a prediction for an instance $f(\mathbf{x}) = \beta^0 + \beta^1 x^1 + \dots + \beta^J x^J$, then recovering Shapley values for a feature j for the instance $\mathbf{x} \in X$ using model f , namely $\phi^j(f(\mathbf{x}))$, then attribution can be directly calculated as:

$$\phi^j(f(\mathbf{x})) = \beta x^j - \mathbb{E}[\beta^j X^j] \quad (2.7)$$

Then, summing over all features j will return:

$$f(\mathbf{x}) - \mathbb{E}[f(X)] \quad (2.8)$$

this is often not plausible and thus to determine attribution over a black-box model f , then one can approximate the Shapley Value with the weighting kernel in equation 2.6. This weighting is applied to a linear model, that is fit over all coalitions of features with and without each feature j .

To produce such coalitions, consider a mapping function $\lambda \in \{0, 1\}^J$, let $\lambda(\mathbf{x}^j) : j(0 \leq j \leq J)$ produce a permuted set of $\mathbf{z} \subseteq \mathbf{x}$, where non-active features are randomly sampled, then fitting a weighted linear model over all permuted samples provides an approximation for Shapley values given any function f . Due to the formalisation of SHAP following the game-theoretic framework for Shapley values, SHAP in theory adheres to a selection of formal properties, these are given as:

- **Efficiency** The efficiency axiom asserts that the marginal contribution of individual features are apportioned correctly to explain the model prediction. Thus the sum of the Shapley values for all features reflects the difference between the model's prediction for a specific instance and its average prediction across all instances.
- **Symmetry** The symmetry axiom states that two features that contribute the same amount to a prediction are given equal feature attribution.
- **Dummy** The dummy axiom states that any feature that has zero influence over the prediction must have a zero feature-attribution value.
- **Additivity** The additivity axiom states that the combined attribution of two features, is equal to the sum of attribution of two features independently (analogous to the assumption of independence).

2.2.3 Explainable Boosting Machine

The Explainable Boosting Machine (EBM) takes the form of a Generalised Additive Model (GAM). First, the concept of a Generalised Linear Model (GLM) is explained, to motivate the approach of the GAM framework. GLMs are developed such that, we remove the Gaussian expectation from linear models and instead generalise to any distribution from the exponential family. Thereby, consider a standard linear equation given by:

$$\mathbf{y} = X^T \beta$$

Where $\beta = \langle \beta^1, \dots, \beta^J \rangle$ is the coefficient vector for J features. For a GLM the use of a link function ζ , incorporates a relationship between the expected value of \mathbf{y} , the probability distribution within the exponential family for \mathbf{y} , namely $E_{\mathbf{y}}$ and with the linear solution, thus yielding:

$$\zeta(E_{\mathbf{y}}(y|\mathbf{x})) = \mathbf{x}^T \beta$$

This falls under the family of interpretable models, as naturally one can interpret the model coefficients by applying a function that cancels out the link function ζ , the link function is represented by ζ^{-1} (e.g. $\zeta = \ln$, $\zeta^{-1} = \exp$), by multiplying both sides of the equation. Therefore, an explanation for an instance \mathbf{x} , is given by:

$$E_{\mathbf{y}}(y|\mathbf{x}) = \zeta^{-1}(\mathbf{x}^T \beta)$$

As previously stated, EBMs are inherently interpretable and are an extension of the GAM, which is defined as:

$$\zeta(E_{\mathbf{y}}(y|\mathbf{x})) = \beta^0 + \sum_{j=1}^J f^j(\mathbf{x}^j).$$

The GAM framework relaxes the linear constraint, by allowing each feature $x^j \in \mathbf{x}$ to have an arbitrary function f^j applied to it. The EBM extension allows for the incorporation of pairwise terms, thus extending to:

$$\zeta(E_{\mathbf{y}}(y|\mathbf{x})) = \beta^0 + \sum_{j=1}^J f^j(\mathbf{x}^j) + f^{j,m}(\mathbf{x}^j, \mathbf{x}^m)$$

where, $m \neq j$. Naturally, the function transformation f , interactions terms and spline-like decomposition of features can decrease interpretability. Although this is true, due to the linear composition of functions, one can still determine the importance of the dependent variable.

2.2.4 Integrated Gradients

Informally, the Integrated Gradients (IG) [STY17] method presents an XAI approach that accumulates gradients on a path from an all-zero instance baseline \mathbf{x}' and an instance to explain \mathbf{x} . The accumulation of gradients over a path from an all-zero baseline acts as a neural state, where the importance of each feature is measured as it is activated over small interpolations. For each feature of the gradient, the sign signifies its positive or negative contribution to the model's prediction for a given instance \mathbf{x} , while the magnitude reflects the degree of importance, thus explaining the instance prediction.

Path-Integrated Gradients (Path-IG) represent a path as a function between instances [STY17]. Formally, given a dataset X , the Path-IG method represented over the j^{th} (feature) dimension of an instance $\mathbf{x} = \langle x^1, \dots, x^J \rangle \in X$, is given by a smooth function $\psi = \langle \psi^1, \dots, \psi^J \rangle : [0, 1] \rightarrow \mathbb{R}^J$ defining a path in \mathbb{R}^J , integrating over α maps to an interpolated point α between a baseline and input, such that $\psi(0) = \mathbf{x}'$ and the instance of interest $\psi(1) = \mathbf{x}$, thus Path-IG is represented as:

$$\text{Path-IG}_j^\psi(\mathbf{x}) := \int_{\alpha=0}^1 \frac{\partial F(\psi(\alpha))}{\partial \psi^j(\alpha)} \frac{\partial \psi^j(\alpha)}{\partial \alpha} d\alpha$$

where F is a differentiable deep network $F : \mathbb{R}^J \rightarrow \mathbb{R}$. The IG method adheres to one unique path, namely a straight line path from an all-zero baseline $\mathbf{x}' = \langle 0^1, \dots, 0^J \rangle$ to an input \mathbf{x} . Thereby, IG is defined as:

$$\text{IG}_j(\mathbf{x}) := (x^j - x^{j'}) \int_{\alpha=0}^1 \frac{\partial F(\mathbf{x}' + \alpha(\mathbf{x} - \mathbf{x}'))}{\partial x^j} d\alpha, \quad (2.9)$$

where $x^{j'}$ is the j^{th} feature dimensions of the baseline \mathbf{x}' . The Riemann approximation $\text{IG}^{\mathcal{R}}(\cdot)$ for a computable implementation of IG is given by:

$$\text{IG}_j^{\mathcal{R}}(\mathbf{x}; K) := (x^j - x^{j'}) \times \frac{1}{K} \sum_{k=1}^K \frac{\partial F(\mathbf{x}' + \frac{k}{K} \times (\mathbf{x} - \mathbf{x}'))}{\partial x^j}.$$

If we consider the average gradient of a path over the j^{th} dimension to be given by $\frac{1}{K} \sum_{k=1}^K \frac{\partial f(\mathbf{x}' + \frac{k}{K} \times (\mathbf{x} - \mathbf{x}'))}{\partial x^j}$, then the average gradient given over a path over all dimensions j can be represented as ∇F . IG assumes the baseline point \mathbf{x}' to be a vector containing all zeros. Therefore:

$$\sum_{j=1}^J \text{IG}_j^{\mathcal{R}}(\mathbf{x}; K) = (\mathbf{x} - \mathbf{x}') \cdot \nabla F = \nabla F \cdot \mathbf{x}.$$

Extrapolating to the non-zero vector case, it can be given that:

$$\sum_{j=1}^J \text{IG}_j^{\mathcal{R}}(\mathbf{x}; K) \approx F(\mathbf{x}) - F(\mathbf{x}')$$

then, we can see that:

$$\nabla F \cdot \mathbf{x} - \nabla F \cdot \mathbf{x}' \approx F(\mathbf{x}) - F(\mathbf{x}')$$

Note, that as K approaches infinity in the approximation, we converge to:

$$\sum_{j=1}^J \text{IG}_j^{\mathcal{R}}(\mathbf{x}; K) = F(\mathbf{x}) - F(\mathbf{x}').$$

While IG exhibits limitations in its application scope, it capitalizes on numerous formal axioms to underpin its methodology. These axioms, which play a foundational role, informally encompass:

- **Completeness:** The difference in prediction between the baseline and input should be equal to the sum of feature attribution values.
- **Sensitivity(a):** For every input and baseline that differ in one feature and the subsequent prediction is different, then feature attribution should only be given to that one feature.

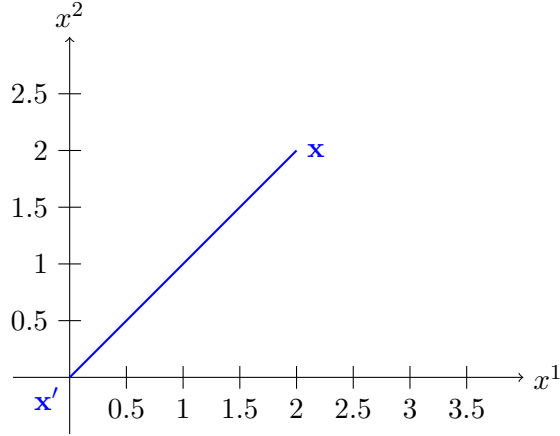


Figure 2.3: Simple intuition of the Integrated Gradients method. Here a straight line is given from the baseline \mathbf{x}' to the instance to explain \mathbf{x} . Here, $\mathbf{x} = \langle 2, 2 \rangle \in \mathbb{R}^2$ and $\mathbf{x}' = \langle 0, 0 \rangle \in \mathbb{R}^2$.

- **Sensitivity(b):** If the neural network is not mathematically dependent on one feature, the feature attribution assigned to that feature should be 0.
- **Implementation Invariance:** Two functionally equivalent neural networks should produce the same feature attribution as an explanation.
- **Linearity:** Given a linear composition of two neural networks F_1 and F_2 that is modelled by a third neural network with weights α and β such that: $F_3 = \alpha F_1 + \beta F_2$, the assigned attribution should be the weighted sum of attributions for F_1 and F_2 with the weights α and β .

The IG framework is thus positioned as a state-of-the-art method which has been later expanded in a variety of works, for further details on the IG framework refer to [STY17]. Examples of the expansion of IG include the Expected Gradients (EG) [EJS⁺21] and Integrated Hessians (IH) [JSL21]. The EG method formulates an approach to enhance the robustness of the IG method, here the EG method does not require a defined baseline, and instead takes the expectation of IG over perturbed baselines taken from the training data distribution \mathcal{D} . The EG method is defined as:

$$\text{EG}^j(\mathbf{x}) = \int_{\mathbf{x}'} \left((x^j - x^{j'}) \times \int_{\alpha=0}^1 \frac{\partial F(\mathbf{x}' + \alpha \times (\mathbf{x} - \mathbf{x}'))}{\partial x^j} \right) p_{\mathcal{D}}(\mathbf{x}') d\alpha$$

Where the integral can be rewritten as an expectation, such that:

$$\text{EG}^j(\mathbf{x}) = \mathbb{E}_{\mathbf{x}' \sim \mathcal{D}, \alpha \sim \mathcal{U}(0,1)} \left[(x^j - x^{j'}) \times \frac{\partial F(\mathbf{x}' + \alpha \times (\mathbf{x} - \mathbf{x}'))}{\partial x^j} \right].$$

The IH method extends IG, to consider the pairwise interaction effects between features. Here the second order partial derivatives are considered with respect to two features j

and k . The IH method when $j = k$ is defined as:

$$\begin{aligned} \text{IH}^{j,k}(\mathbf{x}) &= (x^j - x^{j'}) \times \int_{\alpha=0}^1 \int_{\beta=0}^1 \frac{\partial F(\mathbf{x}' + \alpha\beta \times (\mathbf{x} - \mathbf{x}'))}{\partial x^j} d\alpha d\beta + \\ &\quad (x^j - x^{j'})^2 \times \int_{\alpha=0}^1 \int_{\beta=0}^1 \alpha\beta \frac{\partial^2 F(\mathbf{x}' + \alpha\beta \times (\mathbf{x} - \mathbf{x}'))}{\partial x^j \partial x^k} d\alpha d\beta, \end{aligned}$$

and where $j \neq k$:

$$\text{IH}^{j,k}(\mathbf{x}) = (x^j - x^{j'})(x^k - x^{k'}) \times \int_{\alpha=0}^1 \int_{\beta=0}^1 \alpha\beta \frac{\partial^2 F(\mathbf{x}' + \alpha\beta \times (\mathbf{x} - \mathbf{x}'))}{\partial x^j \partial x^k} d\alpha d\beta.$$

Similar to EG, this can be rewritten as an expectation, namely Expected Hessians (EH). The EH method for $j = k$ is defined as:

$$\begin{aligned} \text{EH}^{j,k} &= \mathbb{E}_{\alpha\beta \sim \mathcal{U}(0,1) \times \mathcal{U}(0,1), \mathbf{x}' \sim \mathcal{D}} \left[(x^j - x^{j'}) \times \frac{\partial F(\mathbf{x}' + \alpha\beta \times (\mathbf{x} - \mathbf{x}'))}{\partial x^j} + \right. \\ &\quad \left. (x^j - x^{j'})^2 \times \alpha\beta \frac{\partial^2 F(\mathbf{x}' + \alpha\beta \times (\mathbf{x} - \mathbf{x}'))}{\partial x^j \partial x^k} \right], \end{aligned}$$

and where $j \neq k$, we have:

$$\text{EH}^{j,k} = \mathbb{E}_{\alpha\beta \sim \mathcal{U}(0,1) \times \mathcal{U}(0,1), \mathbf{x}' \sim \mathcal{D}} \left[(x^j - x^{j'})(x^k - x^{k'}) \times \alpha\beta \frac{\partial^2 F(\mathbf{x}' + \alpha\beta \times (\mathbf{x} - \mathbf{x}'))}{\partial x^j \partial x^k} \right].$$

These alterations to the IG infrastructure enable for faster computation of IG, similarly, providing more robustness to noise through the set of uniform baseline perturbations, as well as providing more information by deriving the interaction effects between each feature.

2.3 Counterfactual Explanations

Counterfactual explanations explore what changes can be made to an instance such that the predicted outcome is altered. Here I briefly introduce the methods from Wachter et al. [WMR18] and Diverse Counterfactual Explanations (DiCE) [MST20], whilst collectively there exist other methods for counterfactual explanations [YLXH22, KTKA20, DCL⁺18], the DiCE and Wachters algorithm exhibit the state of the art in application and provide reference for the developed method in the body of this thesis (See Chapter 5).

2.3.1 Wachters Algorithm

To formulate the counterfactual method, one must construct a simple definition of a classification problem. Thus, given a black-box method f , parameterised by a set of weights β , namely f_β that minimises a loss function \mathcal{L} , here the set of weights can be penalised through a regularization technique $\Omega(\cdot)$, for each data point \mathbf{x} and associated label y , a classifier can be defined as:

$$\arg \min_{\beta} \mathcal{L}(f_\beta(\mathbf{x}), y) + \Omega(\beta).$$

Remark 2.1 For further details on regularization and loss functions see Appendix A.

The counterfactual method introduced in [WMR18] is given as:

$$\arg \min_{\mathbf{x}'} \max_{\lambda} \lambda(f_{\beta}(\mathbf{x}'), y') + \delta(\mathbf{x}, \mathbf{x}'), \quad (2.10)$$

where the λ function aims to produce a target prediction y' , such that $f_{\beta}(\mathbf{x}) \neq f_{\beta}(\mathbf{x}')$ and β is a fixed set of weights from a trained model. The terms ensure that the change in \mathbf{x} is minimal w.r.t the counterfactual variant \mathbf{x}' .

2.3.2 Diverse Counterfactual Explanations

The Diverse Counterfactual Explanations (DiCE) method introduced in [MST20] provides further criterion for the formulation of a counterfactual given by equation 2.10. This induces further constraints to ensure *diversity* in the generated counterfactuals. The hyper-parameters (namely, λ^1 and λ^2) are used to weigh each part of the loss function accordingly. Thus, DiCE is defined as:

$$\begin{aligned} \text{DiCE}(\mathbf{x}) = \arg \min_{\mathbf{x}_{c,1}, \dots, \mathbf{x}_{c,b}} & \frac{1}{b} \sum_{i=1}^b \mathcal{L}(f(\mathbf{x}_{c,i}), y_i) \\ & + \frac{\lambda^1}{b} \delta(\mathbf{x}_{c,i}, \mathbf{x}) - \lambda^2 \text{dpp_diversity}(\mathbf{x}_{c,1}, \dots, \mathbf{x}_{c,b}) \end{aligned}$$

Here, b is the number of counterfactuals generated, $\delta(\cdot, \cdot)$ is an arbitrary distance function, and dpp_diversity a parameter for subset selection with a diversity constraint, enabling for a diverse set of counterfactual examples. This loss function is optimised using gradient descent (see Appendix A).

2.4 Conclusion

This chapter provided an overview of the state-of-the-art XAI methods as of the time of this thesis, thus establishing the required background knowledge on XAI methods for the remaining body of this thesis (for a machine learning background please see appendix A).

Part III

**Evaluating Existing XAI
Methods**

Chapter 3

A Comparison of Model-Agnostic Explanations Given on Electronic Health Records

Contents

3.1	Introduction	33
3.2	Methods and Materials	35
3.3	Results	38
3.4	Explanations	41
3.5	Results	49
3.6	Conclusions	52

3.1 Introduction

Medical and health sciences have seen a pursuit of interest for the use of Machine Learning (ML) algorithms. The complex and high dimensional nature of medical datasets present a big challenge to ML algorithms. This is particularly true when clinicians and public health professionals want to be assured that AI solution should be trustworthy. For the question of “what clinicians want?”, Tonekaboni et al. [TJMG19] identified that merely having a highly accurate ML model is not sufficient to be adopted by clinical staff. Due to the fragile nature of medical data, health data scientists need to provide ML models with both good prediction performance and model *interpretability*. This is exactly the mission of eXplainable Artificial Intelligence (XAI) techniques for real-world applications. In medical diagnosis, explainability/interpretability is needed to enhance robustness of an AI system and enable diagnostics to prevent bias, unfairness, and discrimination, as well as to increase trust by all users in why and how decisions are made.

The explainability of AI systems has been described as early as the 1980s [FS80, Cla87]. As a research prototype developed for diagnosing bacteria infections of the

bloodstream, MYCIN [FS80] demonstrated a potential of explaining which of its hand-coded rules contributed to a diagnosis in a specific case [MLM04]. In 1990s and 2000s, researchers focused on interpretability of rule-based and logic-based inference systems [LAT96, NK99, YWG98, ZG08, ZG09], or improvement of transparency of shallow neural networks by generating meaningful rules from trained neural networks [Fu94, BCR97, TAGD98]. For example, a framework of low-level interpretability and high-level interpretability has been proposed for fuzzy logic inference systems [ZG08] to build an AI system model with a good trade-off between system accuracy and model interpretability. However, these XAI models have a common limitation, that is, they lack of the capacity of dealing with big data or large datasets with high dimensionality. In the age of big data, deep learning has become a popular AI technique since the 2010s [YLH15] due to large processing capabilities of training with big data by utilising using manycore architectures, allowing for parallel processing in high-performance computing (HPC) to produce significant speedups. The victory of quiz show Jeopardy by IBM Watson in February 2011 [Gab11] and ImageNet victory by a computer vision system in October 2012 [AKH12] marked the start of a new AI wave to transform the industry and society. Deep learning is a black box approach which tends to achieve high prediction accuracy without consideration of model interpretability and transparency. Until recently, when XAI became an active research focus in computer science community due to the advances of big data and various regulations of data protection in developing AI systems, such as the GDPR and the EU AI Act¹. For example, according to the GDPR, citizens have the legal right to an explanation of decisions made by algorithms that may affect them (see Article 22), and the EU AI act that requires traceability, transparency and documentation for high-risk AI systems, to promote trustworthy AI solutions. These policies highlight the pressing importance of transparency and interpretability in algorithm design.

Current XAI research efforts focus on developing new approaches for explanations of black-box models by achieving good explainability without sacrificing system performance [LL17, RSG16, RSG18, DFS22]. One typical approach is the extraction of post-hoc explanations. Other approaches are based on hybrid or neuro-symbolic systems, advocating a tight integration between symbolic and non-symbolic knowledge, e.g., by combining symbolic and statistical methods of reasoning. However, very few studies have been done to utilize the XAI techniques to tackle medical diagnostics problems and investigate how the explanations created by XAI techniques can provide insight into medical diagnostics.

The objectives of this study are 1) to demonstrate the explainable visualization of XAI for high-dimensional medical data; 2) to investigate the different model-agnostic methods and the extraction of feature importance, providing commonalities and differentiation amongst each; 3) to determine pros and cons of the XAI models to provide aid to human-experts approaching high-dimensional data. Specifically, we aim to define importance of features in high-dimensional medical data and extract interpretable knowledge that

¹<https://artificialintelligenceact.eu/the-act/>;
<https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai>

acts as a subsidiary tertiary layer to aid medical experts in the decision making process.

3.2 Methods and Materials

We apply and compare three different model-agnostic techniques to obtain the influential input features for each medical question, this work is an extension of previous work carried out on a single case study [DFB⁺21], to validate the results of the initial comparison across a range of medical problems. For the introduction of the model-agnostic models - SHAP, LIME and Scoped Rules (Anchors) [RSG18], local explanations will be provided for a single case-study for each medical problem, this will be used to demonstrate each explainable method with its supported explanations.

3.2.1 Data Pre-processing and Case Study

This study used electronic health records from the Simulacrum [Pub], a synthetic dataset developed by Health Data Insight CiC to mimic some of the data held securely by the Public Health England’s National Cancer Registration and Analysis Service (NCRAS). The Simulacrum data set consists of 1,322,100 synthetic patients allowing for model development by researchers whilst maintaining patient confidentiality. Simulacrum data reflects a high degree of accuracy of the properties found in NCRAS data set, allowing for the development of transferable models from synthetic data sets to real-world data sets.

The Simulacrum faithfully maintains the structural integrity of the NCRAS, including its data structure and interconnections. Moreover, it upholds the statistical patterns observed in NCRAS data, such as the distribution of various data features and the correlations among them, with a claimed high level of accuracy. For instance, it accurately reflects known statistical relationships within the NCRAS data, such as the association between cancer site and gender. For instance, breast cancer is predominantly observed in females, while lung cancer affects males and females roughly equally. These statistical relationships are asserted to be inherent truths within the original NCRAS dataset².

The initial steps of using Simulacrum data set involves data pre-processing to allow for the data to be used effectively in model development. We first created a master table concatenating both Cancer Registration (AV) and Systemic Anti-Cancer Therapy (SACT) tables with a total of 63 columns, the available tables with the associated number of columns is illustrated in Table 3.1. This study aims to tackle medical diagnostics problem in different scenarios by standard ML methods and an XAI framework. Three supervised classification problems are identified for our cohort of lung cancer patients.

- *Predicting the likelihood of reducing a lung-cancer patient’s drug dose. (Binary Classification)*

²<https://digital.nhs.uk/ndrs/data/data-outputs/simulacrum-user-guide/>

Table 3.1: Overview of available Simulacrum data set tables with the corresponding number of columns.

Table Name	No. Associated Columns
sim av patient	6
sim av tumour	16
sim sact patient	2
sim sact tumour	5
sim sact cycle	6
sim sact drug detail	10
sim sact regimen	10
sim sact outcome	8

- *Predicting the likelihood of mortality for a lung-cancer patient. (Binary Classification)*
- *Predicting the survival time of a lung-cancer patient given conditional boundaries. (Multi-class Classification)*

During data pre-processing we identified cases of patients with logical inconsistencies, such as: having a weight or height that is unrealistic for a human, cases where undergoing a regimen after death or vital status being claimed as alive at such time. We treated them as noise within the data and removed them. The dataset is then balanced to the lower bounds class bias for each binary classification output. Here, we explore lung cancer (LC) patients for mod dose reduction (MD) prediction, mortality prediction (DA - Dead/Alive) and survival time (ST) prediction.

The set of input features from the Simulacrum data set is described in the Appendix (Table C.1). The feature “vital status” identifies a patient being in either the “Dead” or “Alive” state is used as the binary classification output for the LC-DA problem. The feature of Mod Dose Reduction being “Yes” or “No” for a patient is used for the binary classification output for the LC-MD problem. Then we propose the multi-class classification problem of ‘Survival Time’: ‘less than 6 months’, ‘between 6 months to 1 year’ and ‘greater than 1 year’. There were 22,860 patients surviving *greater than 1 year*, 24,399 patients surviving *between 6 months to 1 year*, and 61,023 patients surviving *less than 6 months*.

The Simulacrum data set contains many missing values for some features. The most prominent null-value percentiles across the new data set are displayed in Table 3.2. For comparisons, the patient instances for the the lung cancer survival time (LC-ST) problem were limited to a single feature column with a null-value, with all instances > 1 null-value being removed. Conversely, the Lung-Cancer Deceased Alive (LC-DA) and Lung-Cancer Mod Dose Reduction (LC-MD) problems were imputed with the mode, given the missing data despite the number of feature values missing.

Table 3.2: Status of Survival Time Feature Missing Values

Feature Name	Approx. null-value Coverage (%)
<i>ACE</i>	44%
<i>Regimen Outcome Description</i>	25%
<i>CNS</i>	7%
<i>Regimen Time Delay</i>	5%
<i>N Best</i>	2%
...	...

3.2.2 Evaluation of Performance

To measure the performance of baseline models we use the metrics of precision, recall, accuracy, specificity, F1 Score and the area under the receiver operating characteristic (AUROC) curve plot. The above metrics can be described as:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (3.1)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}, \quad (3.2)$$

$$\text{Sensitivity/Recall} = \frac{TP}{TP + FN}, \quad (3.3)$$

$$\text{Specificity} = \frac{TN}{TN + FP}, \quad (3.4)$$

$$\text{F1 Score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \quad (3.5)$$

Sensitivity is a metric that correctly identifies the diseased (malignant) patients, whereas specificity is a metric that accurately identifies healthy (benign) subjects. Since it is a binary classification problem, 2×2 confusion matrix is computed, consisting of four values, namely (i) True Positive – TP (Both actual and predicted class is malignant), (ii) False Positive – FP (actual class is benign but predicted as malignant), (iii) False Negative – FN (actual class is malignant but predicted as benign) and (iv) True Negative – TN (Both actual and predicted class is benign). Based on these four confusion matrix values, all the classification metrics are computed using equation Eqn. 3.1, Eqn. 3.2, Eqn. 3.3, Eqn. 3.4 and Eqn. 3.5.

The ramifications of the false positives and false negatives carry distinct consequences. False positives not only induce psychological distress in patients within medical environments but also lead to an unnecessary investment of time, resources, and procedures. Conversely, false negatives may result in delayed or absent treatment opportunities and forego potential early interventions, constituting a significant setback for patients. Hence, it is imperative to minimize the occurrence of both false positive and false negative rates.

Table 3.3: Baseline performances for logistic regression, XGBoost and EBM tested on each medical problem.

Dataset		Precision (%)	Recall(%)	Accuracy(%)	F1 Score(%)
LC-DA	<i>Logistic Regression</i>	68	68	68	68
	<i>XGBoost</i>	72	72	72	72
	<i>EBM</i>	67	67	67	67
LC-MD	<i>Logistic Regression</i>	68	68	68	68
	<i>XGBoost</i>	76	72	73	74
	<i>EBM</i>	64	62	62	63
LC-ST	<i>Logistic Regression</i>	94	93	93	93
	<i>XGBoost</i>	97	96	96	96
	<i>EBM</i>	89	88	88	88

3.3 Results

Before running XAI methods, we first run a black-box classifier *XGBoost* to generate predictions. We compare the performance of the black-box algorithm against glass-box methods *Explainable Boosting Machine (EBM)* [NJKC19] and *Logistic Regression*. All results are depicted in Table 3.3. Here we have Lung Cancer patients for morality prediction (LC-DA), the reduction of drug dosage for Lung Cancer patients (LC-MD) and Survival Time of Lung Cancer patients (LC-ST). The effectiveness of ML classifiers is pivotal for the practicality of XAI. It is crucial to ensure well-balanced and ample data availability, along with satisfactory performance from the ML model, to effectively utilize explanations. Smaller datasets are prone to sample bias or high variance, thus hindering the model’s ability to generalize effectively to a problem. Similarly, heavily imbalanced classes may lead to under-representation and poor generalization. Moreover, a subpar ML model will inevitably yield inadequate explanations, potentially resulting in misinformation within the explanation. In such cases, the XAI method may attribute importance to features erroneously, or with inherent bias since XAI methods aim to explain the model.

The data set for the LC-DA problem contains 49,456 Lung Cancer patients, extracted from the Simulacrum data set, with 48.94% deceased and 51.06% alive. The models are trained on 70% of the given data and tested on remaining 30%. The data set for the LC-MD problem contains 49,319 patient instances, with 27,752 patients having their drug dose reduced and 21,567 not having their drug dose reduction, with 70% of the data used for training and 30% for testing. The data set for the LC-ST problem contains 2,260 patient instances with 80% used for training and 20% for testing the model.

We compare this against the glass-box methods EBM and Logistic Regression. Therefore, provided in bar charts are used to illustrate a comparison of base performance metrics, these being precision, recall and accuracy of each applied baseline model, as

shown in Figure 3.1. From Figure 3.1 and Table 3.3, it can be seen that the black-

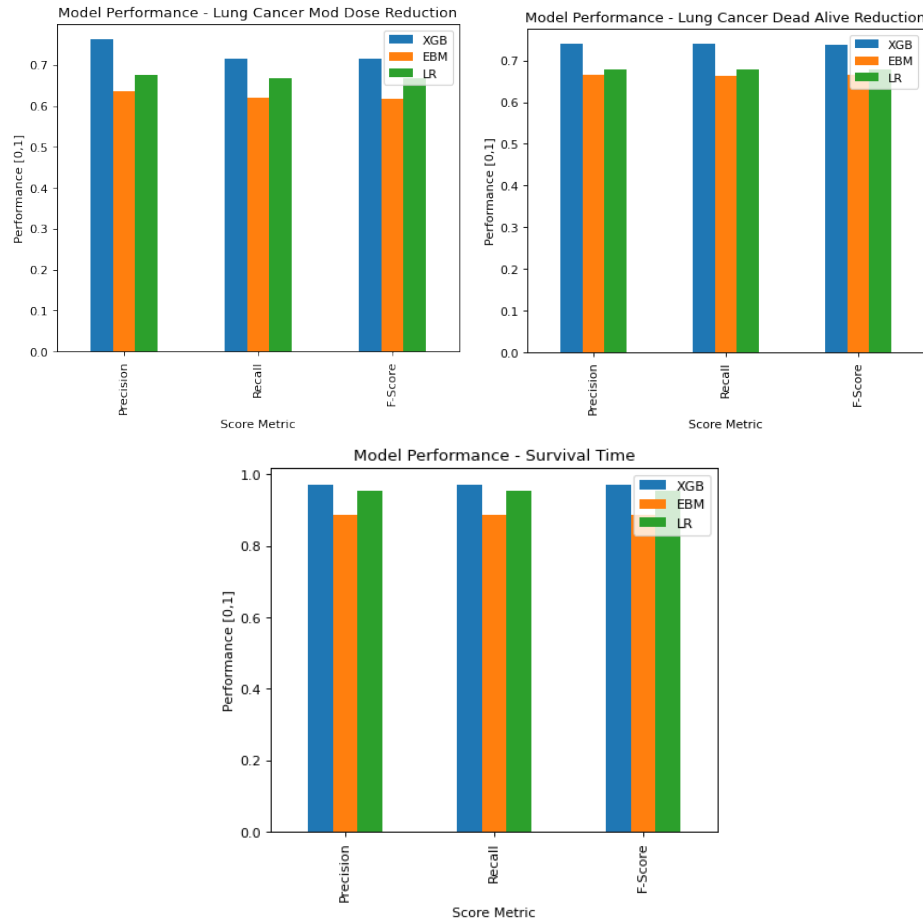


Figure 3.1: Comparison of algorithm performance across each available data set, with performance metrics for Logistic Regression, XGBoost and the EBM method

box XGBoost algorithm is the best performing method across all the given diagnostic tasks, so it can be used as the baseline algorithm for the extension of model-agnostic solutions. We then present an overview of performance using an ROC (receiver operating characteristic) curve and its AUC (area under curve). Figure 3.2 shows the AUROC for LC-DA determining the best fit for the model with the accuracy of an AUC of 0.80 for both classes, naturally averaging to the fit, while the best fit for the model for minimising False Positive and Maximising True positive is around 0.72. Figure 3.3 shows the AUROC for LC-MD determining the best fit for the model with the accuracy of an AUC of 0.83 for both classes and the best fit for the model for minimising False Positive and Maximising True positive is around 0.73. Similarly, Figure 3.4 shows the AUROC for LC-ST determining the best fit for the model with the accuracy of an AUC of 0.98 for survival < 6 Months, with 0.99 for ≥ 6 Months and < 1 Year and 1.00 for the 1 year class, while the best fit for the model for minimising False Positive and

3. A Comparison of Model-Agnostic Explanations Given on Electronic Health Records

Maximising True positive is around 0.96. The micro-average and macro-average ROC curves present two forms of evaluation, the micro-average aggregates contributions of each class, giving an equal weight to each instance, whereas the macro-average treats each class equally, presenting the average performance over all classes and thus can be beneficial with data imbalance.

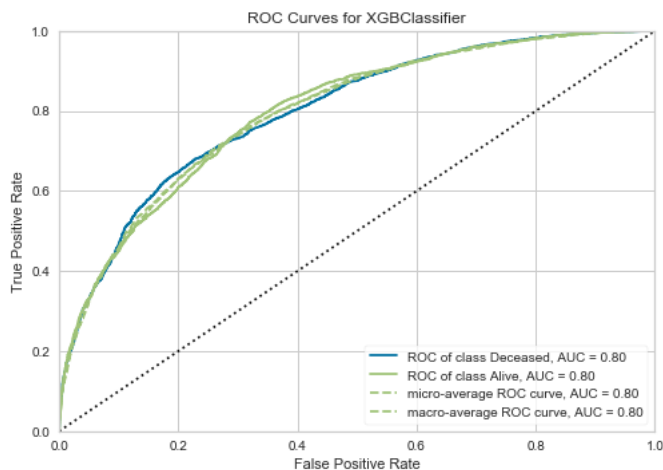


Figure 3.2: AUROC for LC-DA determining the best fit for the model.

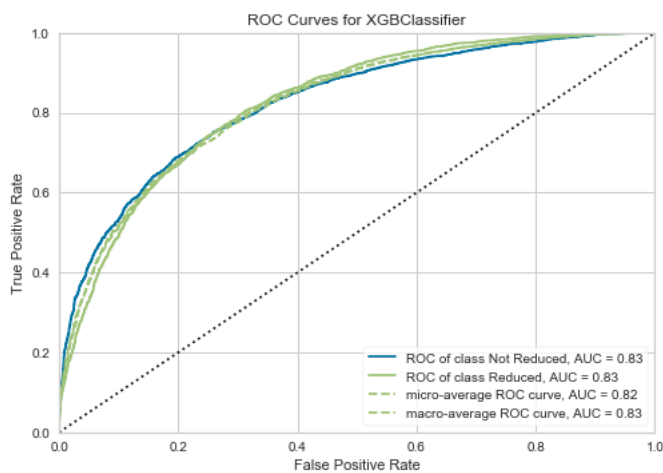


Figure 3.3: The AUROC for LC-MD determining the best fit for the model.

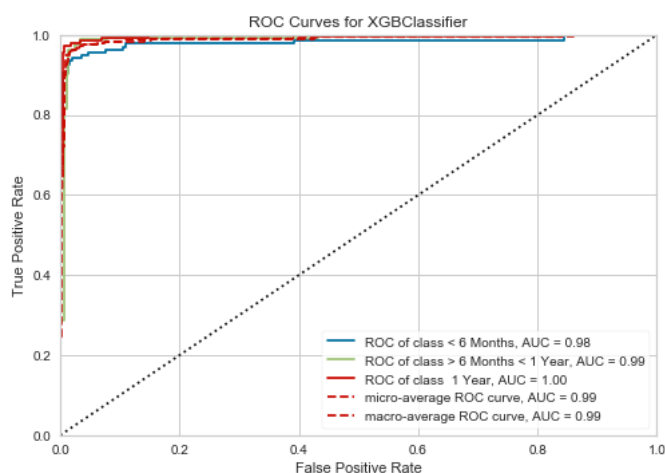


Figure 3.4: AUROC for LC-ST determining the best fit for the model.

3.4 Explanations

3.4.1 Global Explanations: Feature Importance

The SHAP, LIME and Scoped Rules approaches are used to generate feature importance towards classification problems, where a comparative analysis is provided.

3.4.1.1 LC-DA Explanation

The feature importance towards Lung Cancer patients for the binary classification of Dead or Alive (LC-DA) was generated by SHAP on testing data set, as shown in Figure 3.5. The model was trained using XGBoost on a classification problem for Lung Cancer patients. On the x-axis, the magnitude of the given data point distribute to the level of importance for each instance, where 0 is indicative of zero importance. The feature value is represented from low to high based on colour as shown on the right vertical bar, this is shown for each instance over all test data.

It can be seen that the top 3 most important features for the LC-DA problem are M-Best, T-Best and N-Best.

3.4.1.2 LC-ST Global Explanation

Figure 3.6 shows the explanations provided for the multi-class classification LC-ST problem, here we can observe a direct comparison of each features impact on a desired class. The impacts of each feature's positive and negative attribution towards the desired classes can be seen in Figures 3.7, 3.8 and 3.9. These feature importance values allow for the dissection of the models decision. Figure 3.7 reveals that "M Best", "Age" and "Weight" hold the largest impact towards the output class. Figure 3.8 shows that "Weight", "M Best" and "Height" have the most impact on survival in this range, while

3. A Comparison of Model-Agnostic Explanations Given on Electronic Health Records

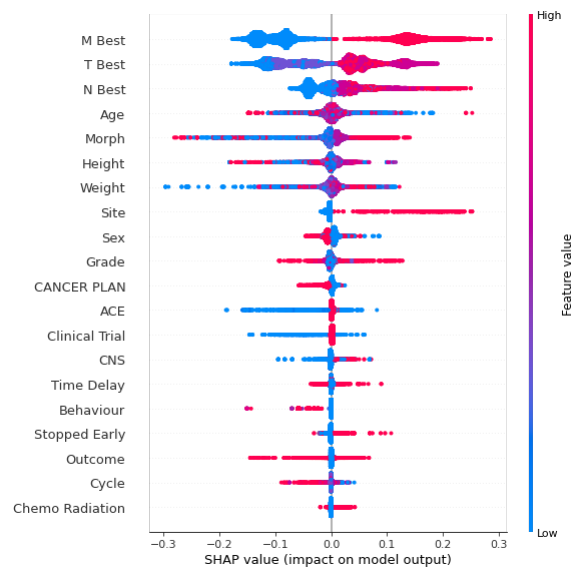


Figure 3.5: SHAP global explanation for the LC-DA problem, the x-axis provides a weighting with positive SHAP values shifting towards survival and negative SHAP values shifting towards deceased.

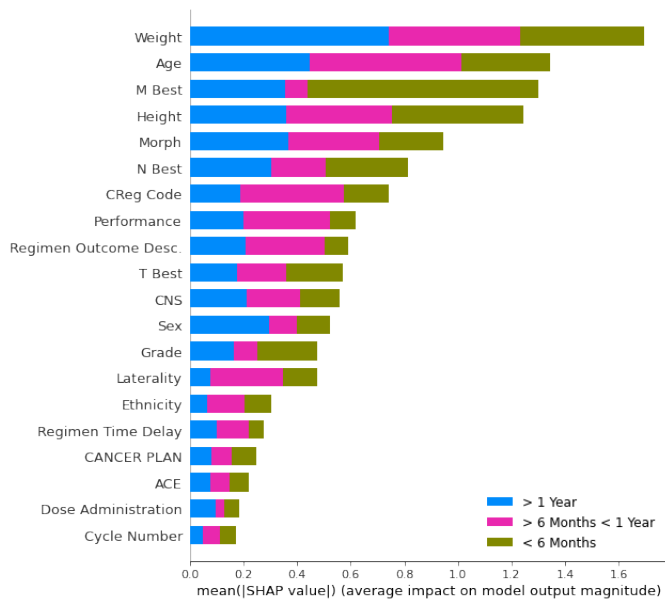


Figure 3.6: Direct comparison of feature attribution towards the output classes. From the bar plot we can determine that “Weight” is the most important feature that corresponds to longer survival, with “M Best” having the most impact for short term survival

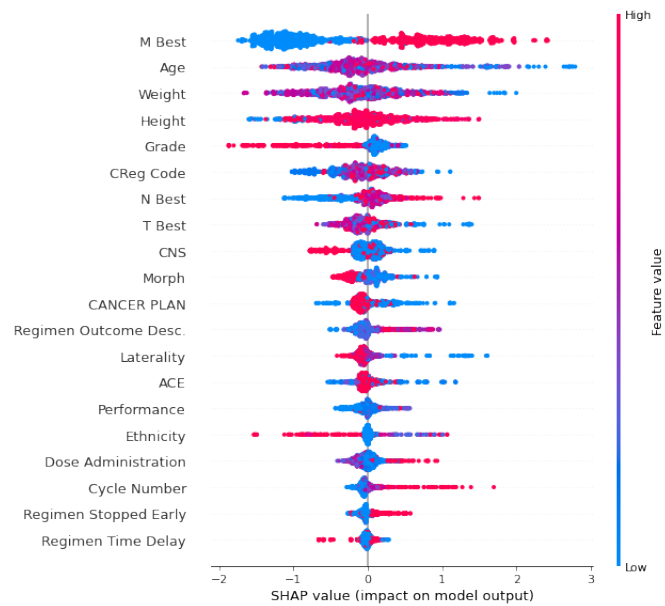


Figure 3.7: SHAP global explanation for the LC-ST problem with feature attribution measured against class[0] *less than 6 months survival*. From this we can observe that the most influential features towards least survival are “M-Best”, “Age” and “Weight”.

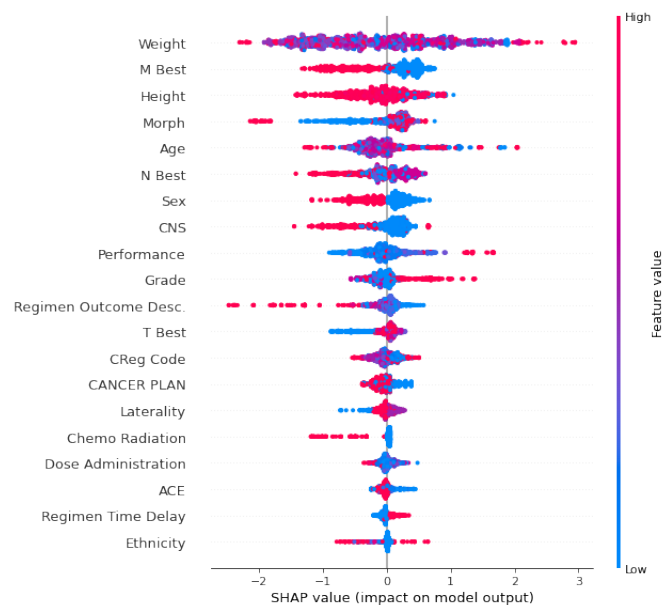


Figure 3.8: SHAP global explanation for the LC-ST problem with feature attribution measured against class[1] *between 6 and 12 months survival*. From this we can observe that the most influential features towards the longest survival time are “Weight”, “M-Best” and “Height”.

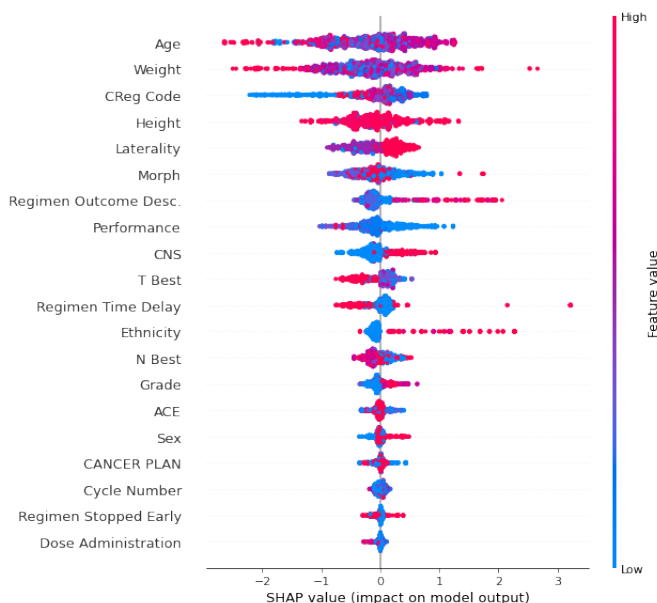


Figure 3.9: SHAP global explanation for the LC-ST problem with feature attribution measured against class[2] *greater than 12 months survival*. From this we can observe that the most influential features towards the survival bracket are “Age”, “Weight” and “CReg Code”.

Figure 3.9 reveals that “Age”, “Weight” and “CReg Code” has the largest impact towards greater than 12 months survival.

3.4.1.3 LC-MD Global Explanation

The global explanation for the LC-MD problem is illustrated in Figure 3.10, where the SHAP values are generated regarding the positive or negative influence shift on the model output based on the associative feature value.

3.4.2 Local Explanations

To conduct a quantitative comparison between the XAI algorithms, we further provided some *local (instance)* explanations. The first case in Table 3.4 was extracted from the binary LC-DA classification problem. The second case in Table 3.5 was taken from the LC-ST multi-class classification problem where we predict the patient instance survival time. The third case in Table 3.6 was taken from the LC-MD binary classification problem.

3.4.2.1 LC-DA Local Explanations

Figures 3.11 and 3.12 present explanation visualisation from the LIME and SHAP methods, respectively, for the patient instance given in Table 3.1. In addition, anchors

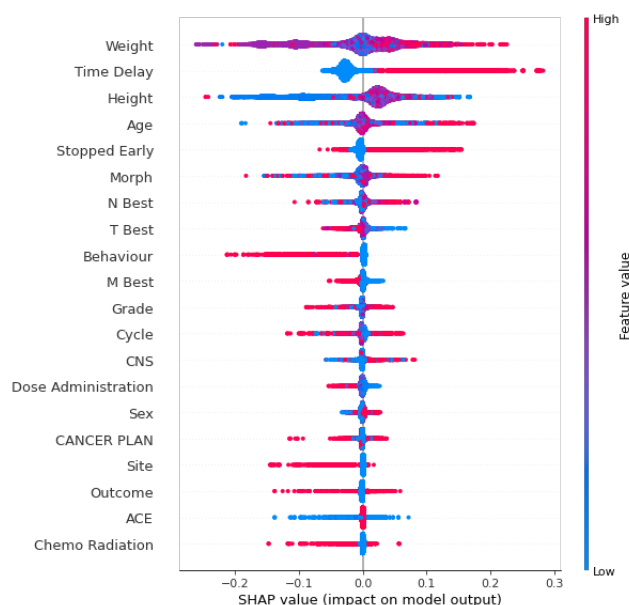


Figure 3.10: SHAP global explanation for the LC-MD problem. We observe that the top 3 most influential features are “Weight”, “Time Delay” and “Height” towards the reduction of drug dose administration.

Table 3.4: A sample patient record in the Simulacrum data set for the LC-DA problem

Age	75	Grade	G3
Sex	Male	Morph	8041
Weight	71.2	Cancer Plan	Curative
Dose Administration	150	Outcome	Treatment completed as prescribed
Drug Group	Etoposide	Administration Route	Oral
Behaviour	Malignant	Regimen Time Delay	No
T Best	4	Regimen Stopped Early	No
N Best	3	Regimen	Cisplatin + Gemcitabine
M Best	1	Clinical Trial	2
Cycle	1	Site	C34
Height	1.57	CNS	99
Chemo Radiation	No	ACE	9

(M Best = 1b) give a conditional conjunction of cases, which identify that given certain elements are true, then the prediction of “Dead” will be true with a coverage of 0.22 and precision value 0.96. In Figure 3.11, LIME provides an explanation with predicted probability of Alive / Deceased, supported by each feature value and the corresponding importance value to which each feature is weighted towards. This demonstrates that the 3 most important features contributing to a patient’s death are “M Best”, “Behaviour” Behaviour of the tumour and “N Best”. Figure 3.12 depicts an explanation by SHAP for a patient, where “M Best”, “N Best” and “Cycle” are considered the most influential features towards the death of the patient. It is noted that all the three methods identify “M-Best” as the most important feature for the classification (the value “1” meaning that cancer has spread to other parts of the body), this provides an empirical ubiquity

across the most important feature for the given patient instance.

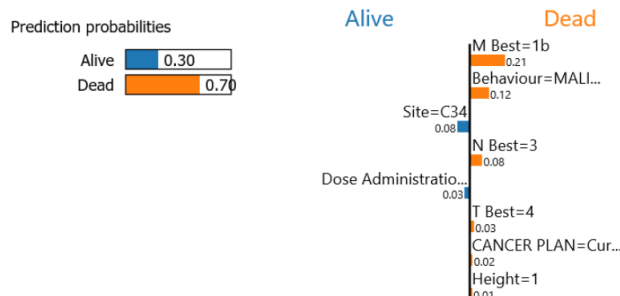


Figure 3.11: An explanation generated by LIME for Alive / Deceased classificaiton.

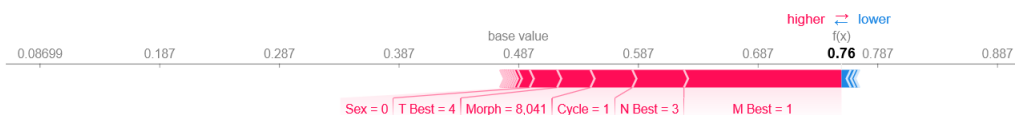


Figure 3.12: An explanation generated by SHAP for a patient: The width of each descriptive block and colour are indicative of the shift in probability to a given case. The colors red and blue denote the direction of prediction shift towards the target classes “Dead” and “Alive,” respectively.

Anchor: M Best = 1b
 Precision: 0.96
 Coverage: 0.22
 Prediction: Dead

Figure 3.13: Anchors give a conditional conjunction of cases, which identify that given certain elements are true; then the prediction will suffice. We see the coverage of these conditions also provided with a precision value $> \tau$

3.4.2.2 LC-ST Local Explanations

We generated an explanation for a local LC-ST patient instance, to provide a demonstration of local explanations for a multi-class classification problem, from this we obtain an explanation where the patient instance introduced in Table 3.5 is predicted to survive between 6 months and 1 year with a high likelihood, and produce the corresponding explanations in Figures 3.14, 3.15 and 3.16. We observe that for the local instance, LIME, SHAP and Anchors share some identifiers towards the prediction *less than 12 months* with “Height” being shared as the most important feature for LIME and SHAP, and “CNS” being the third most important feature for SHAP and LIME, similarly this can be seen with “M Best” in the top 3 most important features for LIME and Anchors, highlighting clear aberration in feature importance ordering, but still agreeing on some features impact.

Table 3.5: A sample patient record in the Simulacrum data set for the LC-ST problem

Age	54	Grade	GX
Sex	Male	Morph	8140
Weight	87	Cancer Plan	Non Curative
Dose Administration	8	Outcome	Progressive disease during Chemotherapy
Behaviour	Malignant	Regimen Time Delay	No
T Best	3	Regimen Stopped Early	Yes
Ethnicity	A	CReg Code	L0201
N Best	1	Performance	0
M Best	1	Clinical Trial	2
Cycle	1	Site	C34
Height	1.88	CNS	99
Chemo Radiation	No	Laterality	Right
Actual Survival (Days)	294	Drug Group	Etoposide

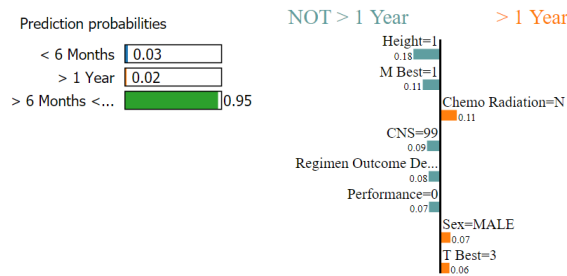


Figure 3.14: Demonstration of a local LIME explanation for the LC-ST problem, We observe that the three most important contributors to the prediction outcome for < 12 Months survival are “Height”, “M Best”, “CNS”.

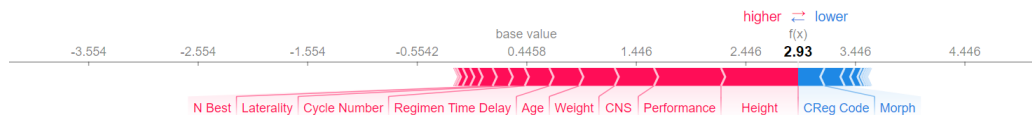


Figure 3.15: Demonstration of a local SHAP explanation for the LC-ST problem. We observe that the three most important instances contributing to the patients predicted survival time are “Height”, “Performance” and “CNS”. From the output we observe a shift towards the defined target class “> 6 Months and < 1 year”.

```

Anchor: Cycle Number <= 2.00 AND Height > 1.74 AND M Best = 1 AND Performance = 0 AND CNS = 99 AND Laterality = RIGHT AND Clinical Trial = 2 AND Regimen Time Delay = N AND Regimen Stopped Early = Y AND Ethnicity = A AND Morph = 8140 AND CANCER PLAN = Non Curative AND Regimen Outcome Desc. = Progressive disease during Chemotherapy AND T Best = 3 AND Age <= 63.00 AND CReg Code = L0201
Precision: 0.77
Coverage: 0.01
Prediction: > 6 Months < 1 Year
    
```

Figure 3.16: Demonstration of a local Anchors explanation for the LC-ST problem. Demonstrating all the Anchors that contribute towards the prediction with the coverage and precision. We can observe “Cycle Number”, “Height” and “M Best” as the first set of anchors provided.

3.4.2.3 LC-MD Local Explanations

Table 3.6: A sample patient record in the Simulacrum data set for the LC-MD problem

Age	64	Grade	4
Sex	Male	Morph	8046
Weight	61	Cancer Plan	Curative
Dose Administration	10	Outcome	Treatment completed as prescribed
Drug Group	Pemetrexed	Administration Route	Intravenous
Behaviour	Malignant	Regimen Time Delay	No
T Best	2a	Regimen Stopped Early	No
N Best	0	Regimen	Carboplatin + Etoposide iv&po
M Best	0	Clinical Trial	N
Cycle	5	Site	C34
Height	1.6	CNS	Y1
Chemo Radiation	No	ACE	9

We generated an explanation for a local LC-MD patient instance, to provide a demonstration of local explanations for another instance of a binary classification medical problem, where we obtain a local explanation using LIME, SHAP and Anchors. From the given explanations we observe aberration in explanations returned.

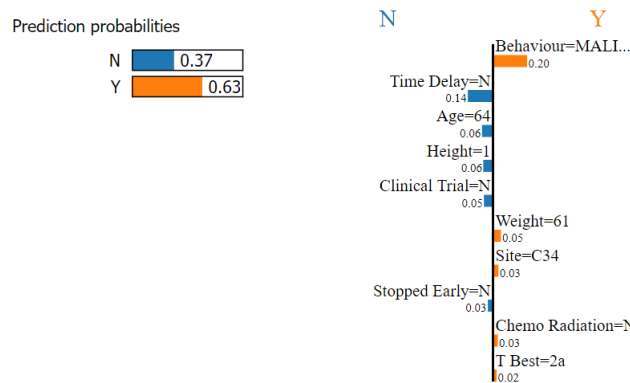


Figure 3.17: Demonstration of a local LIME explanation for the LC-MD problem, from this we can observe that the largest impact toward drug dose reduction being predicted is the tumour “Behaviour” being malignant followed by “Time Delay” and “Age”.

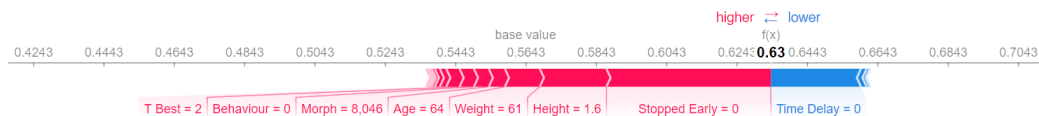


Figure 3.18: Demonstration of a local SHAP explanation for the LC-MD problem. We observe that “Regimen Stopped Early”, “Height” and “Weight” hold the greatest influence towards drug dose reduction being necessary.

```

Anchor: Height > 1.57 AND Morph = 8046 AND CANCER PLAN = Curative AND Site = C34 AND ACE = 9 AND Age <= 79.00 AND Behaviour = M
ALIGNANT AND Dose Administration <= 500.00 AND CNS = Y1 AND Weight <= 63.10 AND T Best = 2a AND Sex = MALE AND Outcome = Treatm
ent completed as prescribed AND M Best = 0
Precision: 0.88
Coverage: 0.00
Prediction: Y

```

Figure 3.19: Demonstration of a local Anchors explanation for the LC-MD problem. From this, we can observe the anchors that the first returned anchors for drug dose reduction are “Height”, “Morph” and the “Cancer Plan”.

3.5 Results

In this chapter, we examined explanations generated by different XAI methods as a tertiary extension for medical diagnostics using electronic medical records. We further presented the shared features by absolute value irrespective of positive or negative attribution towards to the model result. This was evaluated for SHAP and LIME, determining how the two models share a named feature ranking x_k such that $SHAP(x_k) = LIME(x_k)$ for the k th ranking. We analyse this for k features, to determine how many features share the same feature attribution rank. For each data set we analyse $k = 1$, $k = 2$ and $k = 3$, this is evaluated over each data set independently as shown in Figure 3.20. It is worth mentioning that for LIME, SHAP and Anchors, we employed the default hyper-parameter configurations. However, Anchors was omitted from our analysis since the algorithm may not always identify an anchor when τ is set to 0.95, thus this leading to scenarios where no explanation can be provided or only one or two rules are provided. Consequently, the integrity of results for percentile comparison would be compromised. Conversely, SHAP and LIME differ in calculation whilst giving explanations across all instances and thus providing a more meaningful comparison.

We observe inconsistencies amongst the shared features, although there are a high majority of shared features, such as the most important features as shown in the first 1000 instances in Figure 3.21 and for the 400 instances extracted from the LC-ST problem in Figure 3.22. The most important feature is the same feature for a high percentile of the designated population.

Given such information, we can also determine top priority features from the data sets, though they may not be shared for each instance. It is desirable that the identified important features align with domain knowledge representation. Therefore, we provided a comparison across the Lung Cancer problem data sets. The first 1000 instances from the LC-MD and LC-DA and 400 instances from the LC-ST test data were used to extract the most important feature, where $k = 1$ for LIME and SHAP as well as the first anchor extracted from Scoped Rules. The most influential features towards a patient either [“Dead”, “Alive”], a survival time of [“Less than 6 months”, “Between 6 months to 1 year”, “Greater than 1 year”] or reduced drug modification for the current regimen [“Yes”, “No”] were identified.

Commonalities arise across the shared most important feature for each XAI application. Shared importance ordering for features is less common across each model, this differentiating across the data sets. For example, SHAP determines weight as the most important feature in the LC-MD data set, conversely LIME having Time Delay as the

3. A Comparison of Model-Agnostic Explanations Given on Electronic Health Records

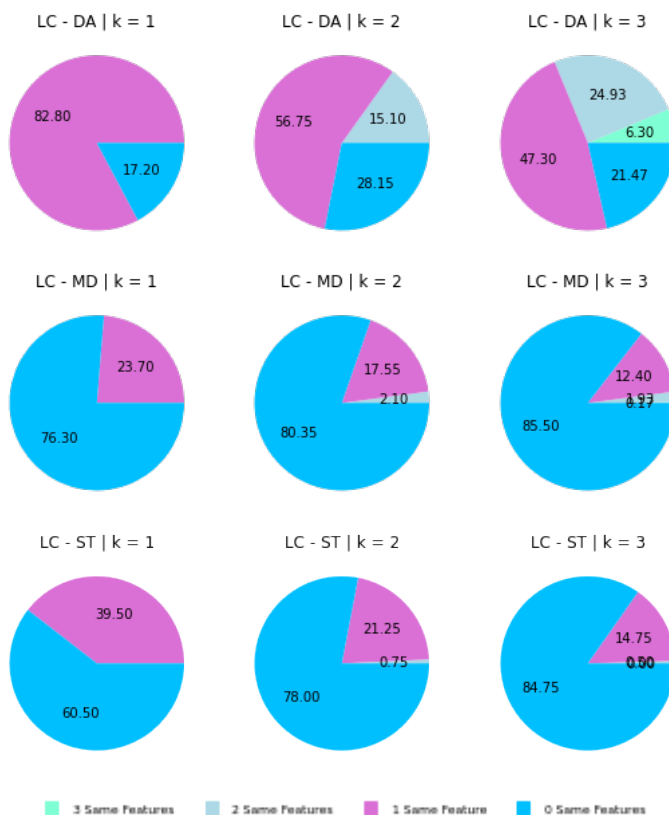


Figure 3.20: Comparing the shared features across each problem using both SHAP and LIME

most important feature. It is observed more balanced dissemination across the three features for the SHAP model. Notably, LIME and Anchors determine the same set of features as important displayed in Figure 3.21.

From Figure 3.21, we can observe the top 3 features for the LC-DA problem are “TNM” staging being the most influential features to prediction for each XAI model. For LC-MD problem, regimen “Time Delay”, “Weight” and “Height” are the three most influential features for LIME and SHAP, whilst not being the condition for Anchors, with SHAP sharing more of an even distribution of importance. Figure 3.22 reveals that the top 3 most influential features are “M Best”, “T Best” and “Age” for LIME and Anchors, whilst SHAP shares “M Best” and “Age” then followed by “Height”.

Figure 3.5 shows the top 3 most important features for the LC-DA problem are M-Best, T-Best, and N-Best. This is a particularly interesting result as it confirms the superiority of TNM based cancer stage classification [LAHYB15]. Cancer staging is a critical step in the diagnosis process with multifarious objectives [LAHYB15], such as helping identify treatment plans, providing indication of prognosis, showing the evaluation of the results of treatment, and facilitating the exchange of information of cancer development. The findings of these 3 top features are also consistent with another

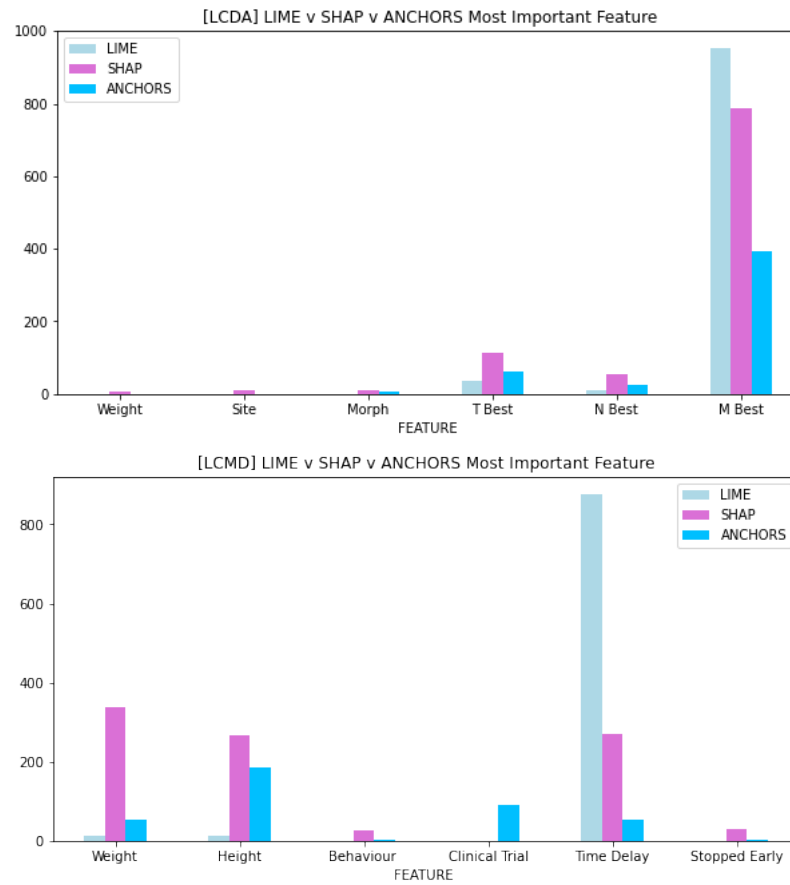


Figure 3.21: Most important feature returned or the first anchor (scoped rules) for the first 1000 instances on the test data set for both LC-DA and LC-MD problems.

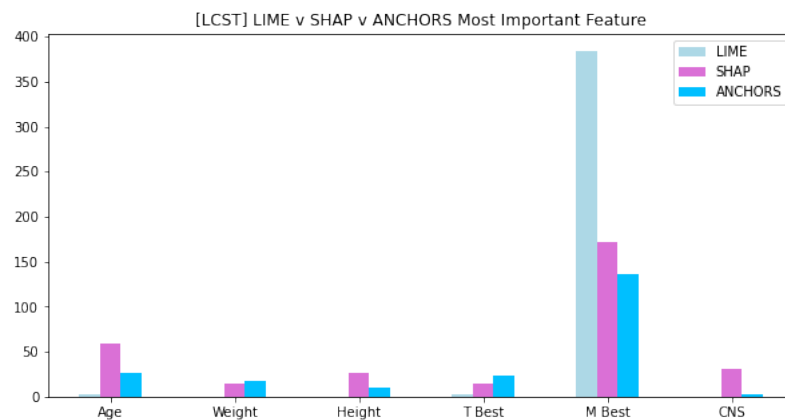


Figure 3.22: Most important feature returned or the first anchor (scoped rules) for the first 400 instances on the test data set for the LC-ST problem.

data-derived cancer prognosis study [WJJ16] indicating that TNM stage remains the most important prognostic features, while being followed by tumor histologic grade, patient sex, age, and performance status. Conversely, note that our study shows that age and cancer morphology are the following important features for predicting mortality of lung cancer patients (Figure 3.5).

There is also differentiation when determining the influential features for predicting minimal survival time where the most influential features are M-Best, age and weight (Figure 3.7), alternatively in predicting the longest survival time (Figure 3.9), we see age, weight and cancer registration code play an important role. Finally, the survival range of 6 months to 1 year lists BMI features and M-Best as the most important predictors for this class.

In the the LC-MD problem (Figure 3.10), we observe that weight is the most influential feature, here we see a negative SHAP value given a medium to low feature value and a positive SHAP value given a larger valued weight. This is then followed by the most influential features being time delay, height and age. It is worth noting that the drug dosage strategy is commonly influenced by weight [dPIZC⁺16], with intervention occurring more often for obese patients [BL17].

This chapter explored the ubiquity of XAI models evaluated on unseen lung cancer patients, this is carried out via a brute force approach that determines the most important features returned by a model and the consistency with domain knowledge is discussed. The k^{th} ranking shows a disparity in the ordering of which features agree, showing that the feature attribution methods determine different features as valuable and important in making the prediction.

Nevertheless, the limitations of this study can be seen in both the theoretical foundations and the limited number of explanation models used. One could further look at shared attribution space and correlations between explanations as a means of similarity (further comparisons are provided in appendix G as to reduce redundant information, as the conclusion of disagreement is reinforced [DFS23]).

3.6 Conclusions

Determining feature priority from a local interpretation provides a sense of clarity for non-domain experts by providing feature contribution to a given case and overall data set. The ability to extract explanations of black-box model predictions is essential for high-risk applications of machine learning; medical implementations being one example, from such information of patterns across features and different explanation models we can determine data set and supporting predictions fairness. Explanation locality allows for new instances to be communicated to the domain-expert with reason. This tertiary layer of knowledge could improve the rate of case deduction and support human-expert reasoning. We observe a clear aberration in the feature attribution priority order amongst the model-agnostic solutions across the three case studies presented, though generally these methods still share an agreement of importance across the top three features.

This is valuable because it highlights the necessity of not depending solely on one XAI model for explanations. This is critical because a singular approach might not excel for every instance, given the early stage of XAI development, necessitating the consideration of a range of methods. Employing a variety of XAI approaches could also reveal inherent biases in models, particularly if there is a general consensus elucidated among different methods regarding the important features identified through feature attribution methods. It is possible for explanations generated by various methods to appear inconsistent or unconvincing. Consideration of the impact of random perturbations is essential in such scenarios. This approach can introduce out-of-distribution data, leading to misguided explanations. Additionally, the influence of human-selected hyper-parameters should not be underestimated. This aspect could potentially allow users of XAI methods to adjust hyper-parameters to achieve results that align more closely with the narrative they intend to convey.

Part IV

Enhancing Explainability - a Model Perspective

Chapter 4

Polynomial Adaptive Local Explanations

Contents

4.1	Introduction	57
4.2	Related Work	58
4.3	Method	59
4.4	Comparative Methods	62
4.5	Results	63
4.6	Conclusion	66

4.1 Introduction

The use of eXplainable Artificial Intelligence (XAI) methods enables transparency for black-box model predictions, thus supplementing a user’s ‘*right for explanation*’ elucidated in Europe’s General Data Protection Regulation (GDPR) [SP17]. As of 2016, there exist variations of XAI surrogate models that explore different approaches to localised explanations, though the premise of XAI greatly predated the recent influx [Frä20]. Perturbation methods have seen success and wide application in the medical domain [DKW⁺21, PZZ⁺21, SKW⁺21, YRC⁺20], popular examples being Local Interpretable Model-Agnostic Explanations (LIME) [RSG16], SHapley Additive exPlanations (SHAP) [LL17] and Scoped Rules (Anchors) [RSG18], where SHAP explores a feature summary through additive marginal contribution evaluation and Anchors and LIME explore local surrogate models from a set of readily interpretable models e.g. linear regression.

In this chapter, our primary objective is to provide local explanations for tabular data, focusing on the context of Electronic Health Records (EHR). EHRs serve as a crucial asset for both population-based health research and individual health analysis. Within the realm of clinical care exploration, the significance of local explanations

cannot be overstated, as they play a pivotal role in establishing trust [TJMG19]. For instance, in the realm of individual health, explanations must inherently incorporate patient-specific information to cater to the unique requirements of each case.

The landscape of individualized health care demands explanations that are tailored to each patient’s context. As a response to this need, tools have emerged that leverage existing eXplainable AI (XAI) methods in conjunction with data exploration and analytic techniques [KED⁺21]. These tools are designed to enhance our understanding and interpretation of complex health data, ensuring that insights gleaned are not only accurate but also comprehensible and meaningful for both healthcare professionals and patients.

In light of the lack of consistency observed across explanations [DFB⁺21], concerns about their reliability have surfaced. In the pursuit of creating local explanations that are not only lucid and effective but also tailored to each patient, we introduce the Polynomial Adaptive Local Explanations (PALE) framework. This model is designed to harmonize the optimization of both a black-box overarching model and the individual local explanations. The primary focus of PALE centers on tabular data, aiming to achieve a level of transparency in patient predictions within a localized context. It accomplishes this by generating explanations that elucidate how each patient and feature collectively influence the outcome. This is achieved through the construction of local surrogate models that adapt to the nuances of each patient’s unique circumstances through scaling polynomial degrees.

The objectives of this chapter are as follows:

1. Produce an end-to-end framework that optimises both the complex model and the local model for each instance;
2. Produce explanations based on the derived scaling polynomial models to understand uni-variate feature impact for local instances;
3. Compare local explanations and local explanation performance across the different XAI methods.

4.2 Related Work

Exploration of local surrogate model explanations saw an effective rise posterior to the efforts of LIME. LIME is a model-agnostic method with a primary focus on local explanation where a local linear model is used on a perturbed set around the instance \mathbf{x}_i . An explanation \mathcal{E} for local point \mathbf{x}_i is defined as

$$\mathcal{E}(\mathbf{x}_i) = \arg \min_{g \in G} \mathcal{L}(f, g, \pi_{\mathbf{x}_i}) + \Omega(g),$$

where we have a local linear model g from a set of interpretable models G , aiming to minimise the error of the local linear model, where perturbations around instance \mathbf{x}_i are subject to a neighbourhood π , where \mathcal{L} measures the fidelity of the local model g

with respect to the complex model f . The Ω term is used to reduce the complexity of the local model g . Perturbations are created around the mean of the data set within one standard deviation following a Gaussian or uniform distribution. See [RSG16] for details.

There are various adaptations and extensions to the LIME framework, focused on improving the reliability of the approach. The authors of Deterministic-LIME (DLIME) [ZK21] extend the LIME framework by producing an adaptive neighbourhood using k -nearest neighbours and hierarchical clustering in an attempt to provide consistent explanations. In [ZHW21] the authors introduce Stabilized-LIME (S-LIME) which also surrounds the improvement of perturbation points for better local explainability, stability in the former DLIME and S-LIME are measured using the Jaccard similarity coefficient. [PMT18] introduces local explanations and example-based local explanations, where weighting is carried out using random forests for supervised neighbourhood selection.

In [BCN20] the authors propose an ensemble approach to LIME, namely LimeOut in order to reduce the reliance of sensitive features, in order to achieve this the authors replicate a similar idea to drop out techniques that are used in neural networks, aiming to maintain model performance. [ZHH⁺21] introduces Bayesian LIME (BayLIME), in efforts to obtain consistency in explanations and maintain model robustness through integration of prior knowledge and the adaptation of Bayesian reasoning.

Extrapolating to local model fits, [SZLF19] introduces Tree-LIME, an approach that replaces the local linear model of LIME with a decision tree based approach for local interpretability. The authors of [BHTL20] draw more comparable intentions, as the authors aimed to fit a quadratic model to extend the LIME local model, the intent to analyse the performance improvement against the linear model. Therefore, the development of this inspired the intent for creating a framework with instance specific explainability to any polynomial degree that fits best for a given case. Feature attribution methods have explored specific feature-types, where we see focus on continuous features, enhancing the idea for the selective perturbation strategy [KLS⁺22].

4.3 Method

This section of the chapter introduces the PALE methodology, and is structured as follows: Section 4.3.1 provides the details of the adaptive model within the PALE framework, also the form of explanation provided. This chapter also illustrates that an explanation can be produced for any instance. The PALE framework also ensures, explanations are only provided such that, the local model has a degree of precision. Section 4.4 provides a set of comparison metrics to compare the introduced PALE method against state-of-the-art and inherently interpretable explanation methods. Section 4.5 provides an example explanation, where the model performance is evaluated in Section 4.5.1.

4.3.1 PALE Framework

Informally, the PALE framework offers explanations tailored to each individual instance. In the context of EHRs, this customization enables the adaptation of the model to the unique features of each patient. To achieve this, the framework generates a perturbed dataset for each instance, creating a convoluted neighborhood around that point. This approach facilitates localized and interpretable insights into model predictions. By introducing small changes to the original instance through perturbations, the framework enables the evaluation of effects concerning minor alterations. Additionally, utilizing a surrogate dataset and model enables a model-agnostic approach, allowing for its application to any complex model.

We propose a end-to-end framework, constructed to optimise the complex model f over all data X , therefore, $f(X)$ denotes our black-box model, where we minimise the residual loss \mathcal{L}_f of the complex model. We optimise the local explainer loss for each i^{th} instance, where $X = \langle \dots, \mathbf{x}_i, \dots \rangle$. We search for the optimal local models $g_m \in G$, where G is a set of polynomial models, for an instance in the local neighbourhood $\pi_{\mathbf{x}_i}$. Local model error is minimised through $\mathcal{L}_{g_{m,i}}$ and weighted using the same neighbourhood setting that is used in the LIME framework, where the optimal m polynomial degree for each instance is obtained. The framework aims to produce local explanations over classification problems, therefore we assume the complex model f to be some classifier.

4.3.1.1 Adaptive Model

Introducing PALE, the generated surrogate data set \mathcal{Z}_i is weighted by some neighbourhood function $\pi_{\mathbf{x}_i}$, for an instance of interest \mathbf{x}_i . The surrogate set can be represented by $\{z', \mathbf{y}\} = \mathcal{Z}_i$, where an instance \mathbf{z}'_s in the surrogate set is defined by $\mathbf{z}'_s \in \mathbb{R}^J$, the surrogate data is given by $\mathcal{Z}_i \in \mathbb{R}^{P \times J}$ and labels $\mathbf{y} \in \{0, 1\}^P$. We let $f(\mathbf{z}'_s)$ for each instance \mathbf{z}'_s be the labels of the surrogate set using the prediction probability as the target for the fit model $g_{m,i}(\mathcal{Z}_i)$. The

We first aim to have a scaling polynomial fit for instance adaptation in order to both provide better localised model performance as well as to provide insight into feature attribution and the affect of feature alteration in the local domain. The framework is composed of two loss functions:

- $\mathcal{L}_f(X; \cdot)$, the loss for the complex model,
- $\mathcal{L}_{g_{m,i}}(\mathcal{Z}_i, f; \beta) + \lambda_p(\beta)$, the loss for the polynomial explainer.

λ_p is a regularization method, $\lambda_p(\beta)$ of our local model in the body of this chapter is l_2 regularization described in Appendix A, \mathcal{L}_f is an arbitrary loss function depending on the complex model selected. We employ $\mathcal{L}_{g_{m,i}}$ to minimize loss, for this chapter we utilise the Root Mean Squared Error (RMSE) to evaluate the localized model performance within a surrogate set \mathcal{Z}_i in the neighborhood $\pi_{\mathbf{x}_i}$. This determines the error in each model to the m^{th} degree polynomial for a prediction $g_{m,i}(\mathbf{z}'_s)$ pertaining to each instance of the surrogate set, and the fidelity to the labels y_s assigned by $f(\mathbf{z}'_s)$. This procedure is

performed individually for every instance \mathbf{x}_i , seeking the optimal m degree polynomial for the optimal set of coefficients $\beta'_{\mathbf{x}_i}$, where

$$\beta'_{\mathbf{x}_i} = \arg \min_m \mathcal{L}_{g_{m,i}}(\mathcal{Z}_i, f; \beta) + \lambda_p(\beta)$$

4.3.1.2 Adaptive Local Explanations

To generate explanations we order $\frac{\partial g_{m,i}}{\partial x_i}$ by absolute value, where the associative value corresponds to the feature importance ranked by its value $|\frac{\partial g_{m,i}}{\partial x_i}|$ for each feature j to gauge a descending order of feature importance. Generalising to a scaling polynomial fit, we can observe the partial derivative for the m^{th} polynomial degree, such that for a single feature x^j we observe the affect of change, where every other feature is kept static $\mathbf{x}_i^{/j}$, therefore:

$$g_{m,i}(x^j + \Delta x^j, \mathbf{x}_i^{/j}) = g_{m,i}(x^j, \mathbf{x}_i^{/j}) + (\Delta x^j) \cdot \frac{\partial g_{m,i}}{\partial x^j}(x_i, \mathbf{x}_i^{/j}),$$

as such we obtain a complete set of polynomial model partial derivative based explanations over the given data set X . We refer to this set of polynomial explanations as $\mathcal{E}_p(X)$, where each row corresponds to a instance \mathbf{x} , and each column corresponds to the features,

$$\mathcal{E}_p(X) = \begin{bmatrix} \frac{\partial g_{1,m}}{\partial x^1} & \frac{\partial g_{1,m}}{\partial x^2} & \cdots & \frac{\partial g_{1,m}}{\partial x^J} \\ \frac{\partial g_{2,m}}{\partial x^1} & \frac{\partial g_{2,m}}{\partial x^2} & \cdots & \frac{\partial g_{2,m}}{\partial x^J} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_{N,m}}{\partial x^1} & \frac{\partial g_{N,m}}{\partial x^2} & \cdots & \frac{\partial g_{N,m}}{\partial x^J} \end{bmatrix}. \quad (4.1)$$

We then use the Hadamard of product of this matrix with the corresponding instances given by each row that are associated to the explanation to utilise the concept of directional derivatives, then we have the PALE explanation over our data X , given by:

$$\text{PALE}(X) = \mathcal{E}_p(X) \odot X = \begin{bmatrix} \frac{\partial g_{1,m}}{\partial x^1} x^1 & \frac{\partial g_{1,m}}{\partial x^2} x^2 & \cdots & \frac{\partial g_{1,m}}{\partial x^J} x^J \\ \frac{\partial g_{2,m}}{\partial x^1} x^1 & \frac{\partial g_{2,m}}{\partial x^2} x^2 & \cdots & \frac{\partial g_{2,m}}{\partial x^J} x^J \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_{N,m}}{\partial x^1} x^1 & \frac{\partial g_{N,m}}{\partial x^2} x^2 & \cdots & \frac{\partial g_{N,m}}{\partial x^J} x^J \end{bmatrix}. \quad (4.2)$$

4.3.1.3 Precision

We introduce a form of local precision, this is a user defined level of precision which is in the range $[0,1]$. The term τ , is a flexible user influenced term that binds whether an instance explanation is returned, to a given precision of local fidelity where a returned explanation given the value for $\tau = 1$ would determine $|(f(\mathbf{x}_i) - g_{m,i}(\mathbf{x}_i))| = 0$. This meaning that the prediction of the local model g accurately represents the point of

interest predicted from our complex model f , meaning $g_{m,i}(\mathbf{x}_i) = f(\mathbf{x}_i)$. This is determined through a term given the complex and local model for an instance of interest and a measure of precision τ , such that,

$$\begin{aligned} \text{Precision}(g_{m,i}, f, \mathbf{x}_i; \mathcal{T}, \tau) &= |(f(\mathbf{x}_i) - g_{m,i}(\mathbf{x}_i))|, \\ \text{s.t. } \text{Precision} &\leq 1 - \tau. \end{aligned}$$

We also allow the user to select a target value, $\mathcal{T} \in \{0, 1\}$ (*1 by default in the binary case*), this will allow for the partial derivative of the local regression to be associated with some user defined \mathcal{T} for an explanation. If the local model does not meet the precision requirements, the instance explanation will not be returned. Therefore, the purpose of this in the applied case is to return only locally precise explanations.

4.4 Comparative Methods

To compare explanations returned by PALE, we evaluate the method against SHAP, a linear model (LIME), and higher degree polynomials and logistic regression explanations.

4.4.1 Jaccard Index

We can explore the Jaccard similarity index for v features, in this body of work we explore $v = 5$. The Jaccard index can be defined by

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

This provides a comparison of the returned sets of feature names between two XAI methods.

4.4.2 Pearson Correlation Coefficient

We also compare the Pearson r correlation coefficient for the sets of explanations, given the absolute values returned from the XAI methods. Where the Pearson r correlation used in comparative XAI work [DFS23] is defined as:

$$r = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2} \sqrt{\sum(Y_i - \bar{Y})^2}} \quad (4.3)$$

where:

- X_i and Y_i are the individual data points in the two variables, and
- \bar{X} and \bar{Y} are the means of the two variables.

4.4.3 Logistic Comparison

To compare PALE to a commonly used approach, we use Logistic Regression. Using this, we provide explanations with respect to the odds ratios (OR), through uni-variate logistic regression analysis on each feature in the perturbed set \mathbf{z}'_i . We introduce the logistic model as the function u_i , where u_i is the local logistic regression model over a surrogate set for instance \mathbf{x}_i . The localised model is a uni-variate model to explore individual feature importance. To achieve this, we introduce a secondary surrogate set \mathcal{Z}' where, $\{z', \mathbf{y}'\} = \mathcal{Z}'$. A feature vector is denoted by $\mathbf{z}'^j \in \mathbb{R}^{P \times 1}$ and associated label is a binary case $\mathbf{y}' \in \{0, 1\}^P$, therefore:

$$u_i(z'^j) = P(\mathbf{y}' | z'^j) = \frac{1}{1 + (\exp(-(\Psi^j \times z'^j)))}. \quad (4.4)$$

We introduce a modified version of OR to center odds at the value 0 for ease of interpretation, the logistic explanation \mathcal{E}_l where Ψ^j is the returned log odds, can be represented by:

$$\mathcal{E}_l(x_i^j) = \exp(\Psi^j) - 1. \quad (4.5)$$

We can use the shift in odds ratio in either \mathbb{R}^+ or \mathbb{R}^- of non-absolute value explanations for each feature j , of an instance. To assess the similarity between the explanation derived and the odds ratio explanation, we apply the sign (sgn) function to the derivative of the polynomial with respect to the returned $\mathcal{E}_l(x_i^j)$. We determine the ratio of shared explanation shift LogCompare for any \mathbf{x}^j over J features as:

$$\text{LogCompare}(\mathbf{x}_i) = \begin{cases} \frac{1}{J} \sum_{j=1}^J \mathbb{1}_{[x^j]}, & \text{if } \text{sgn} \left(\frac{\partial g_{i,m}}{\partial x^j} \right) = \text{sgn} (\mathcal{E}_l(\mathbf{x}^j)), \\ 0, & \text{otherwise.} \end{cases} \quad (4.6)$$

4.5 Results

The data used in this chapter is derived from the Simulacrum dataset (see Chapter 3). We extract a subset of lung cancer patients from the Simulacrum to demonstrate the proposed method. We focus on binary classification problems for the demonstration of this framework. The binary classes we aim to predict are *< 6 Months* and *≥ 6 Months* survival time.

We use an XGBoost model with a 70% train and 30% testing data split as our complex model to demonstrate the explanatory model. The model performance is evaluated using the binary cross-entropy loss function, obtaining the results presented in Table 4.5.1.

4. Polynomial Adaptive Local Explanations

Class	Precision (%)	Recall (%)	F1-Score (%)
< 6 Months	Lung Cancer: 97	Lung Cancer: 97	Lung Cancer: 97
	Skin Cancer: 86	Skin Cancer: 89	Skin Cancer: 87
	Breast Cancer: 96	Breast Cancer: 94	Breast Cancer: 95
	Lymphoma: 98	Lymphoma: 98	Lymphoma: 98
\geq 6 Months	Lung Cancer: 98	Lung Cancer: 98	Lung Cancer: 98
	Skin Cancer: 89	Skin Cancer: 86	Skin Cancer: 88
	Breast Cancer: 96	Breast Cancer: 97	Breast Cancer: 97
	Lymphoma: 98	Lymphoma: 98	Lymphoma: 98

4.5.1 Model Performance



Figure 4.1: RMSE measurements for a subset of 100 Simulacrum patient instances across 4 datasets. We can observe how the increase in polynomial degree improves the local model accuracy for most instances, but in some cases a simpler model will suffice.

In Figure 4.1 we present the performance of each model to the m^{th} degree polynomial across four datasets. We analyse the RMSE returned for the local model $g_{m,i}$ for 100 instances $\mathbf{x}_i : i = \{1, 2, \dots, 100\}$. From this, we determine that an increase in polynomial degree has significant impact on the local model performance over each surrogate set \mathcal{Z}_i .

4.5.2 Explanation Example

In this section, we consider a single patient to explain, exploring the following lung cancer patient instance:

- Age 66, Sex 0, Morph 8140, Weight 85.90, Height 1.67, Dose Administration 8, Chemo Radiation 0.0, Regimen Outcome Description 0.0, Admin Route 1.0, Regimen Time Delay 0.0, Regimen Stopped Early 1.0, Cycle Number 1.0, Grade 1.0, Cancer Plan 0.0, Cancer Registration Code 301.0, T Best 4.0, N Best 2.0, M Best 0.0, Laterality 2.0, CNS 1.0, ACE 9.0, Performance 0.0, Clinical Trial 2.0.

Prediction: ≥ 6 Months,

Actual: ≥ 6 Months.

We explore how higher degree polynomial functions can inform feature attribution on a local level. We use the partial derivative for the 2^{nd} (Figure 4.2) and 3^{rd} (Figure 4.3) degree polynomials, to determine how each feature j interacts with the output for our local model. Evaluating the explanations for the top 5 most important features, we



Figure 4.2: Derivation of the quadratic polynomial term - Simulacrum patient instance. The explanation determines how an instantaneous increase in each feature value x_i influences the local polynomial function $g_{m,i}$ at the location of the instance, where we have $g_{2,i}$. Higher (resp. lower) values on the y -axis represent a large (resp. small) feature importance value.

observe that the quadratic derivative in Figure 4.2 determines *Weight*, *M Best* and the *Regimen Outcome Description* to have a high attribution in the local model. Conversely, when observing the 3^{rd} degree polynomial in Figure 4.3, we see *Cancer Registration Code* followed by *M Best* and *Regimen Outcome Description* as the highest attribution in the local model.

4.5.2.1 Similarity Measures

For the comparison of XAI models, we determine the Jaccard similarity index and Pearson r correlation coefficient, between the explanations present by g_m and the

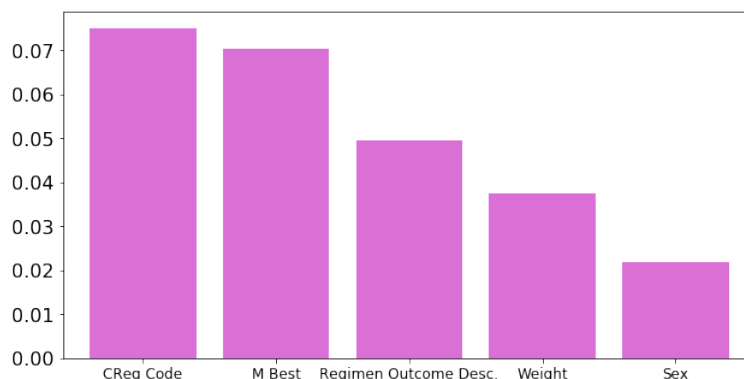


Figure 4.3: Derivation of the cubic polynomial term - Simulacrum patient instance. The explanation determines how an instantaneous increase in each feature value x_i influences the local polynomial function $g_{m,i}$ at the location of the instance, where we have $g_{3,i}$. Higher (resp. lower) values on the y -axis represent a large (resp. small) feature importance value.

explanation given by SHAP. Although the PALE framework extracts the ideal polynomial degree and produces an explanation for each instance, we manually extract explanations for each degree and compare the similarities amongst each degree polynomial and SHAP. In Figure 4.4, it is evident that the highest Jaccard similarity is observed between the third-degree polynomial fit and SHAP. Furthermore, our analysis of Figure 4.4 indicates that, for the given instance, the third-degree polynomial exhibits a stronger correlation with SHAP than lower-degree polynomials, as highlighted by the Pearson r correlation coefficient between each XAI model.

We use LogCompare to the agreement between both the quadratic and cubic explanations for the sign of feature attribution values, as opposed to absolute feature importance values, so we can determine the amount of shared attribution between the logistic model and local polynomial derivations. From this, we obtain $\text{LogCompare}(\mathbf{x}_i) = 0.48$ for the quadratic model explanation and $\text{LogCompare}(\mathbf{x}_i) = 0.65$ for the cubic model explanation. Therefore, we observe in the given case, the cubic explanation has a greater similarity in explanation with the logistic model than that of the quadratic model.

4.6 Conclusion

We use a similar classification problem as seen in [DFB⁺21], [KLS⁺22], where under similar predictions surrounding survival we see great influence from the likes of *M Best*, *Weight*, amongst other features. Therefore, we observe the selection of important features hold a degree of accuracy with clinical knowledge of cancer survival. The contribution of this chapter is an end-to-end framework that optimizes both the local and complex model to provide an explanation of how change to a feature will influence the outcome of the model prediction in the local setting. We emphasise the need for

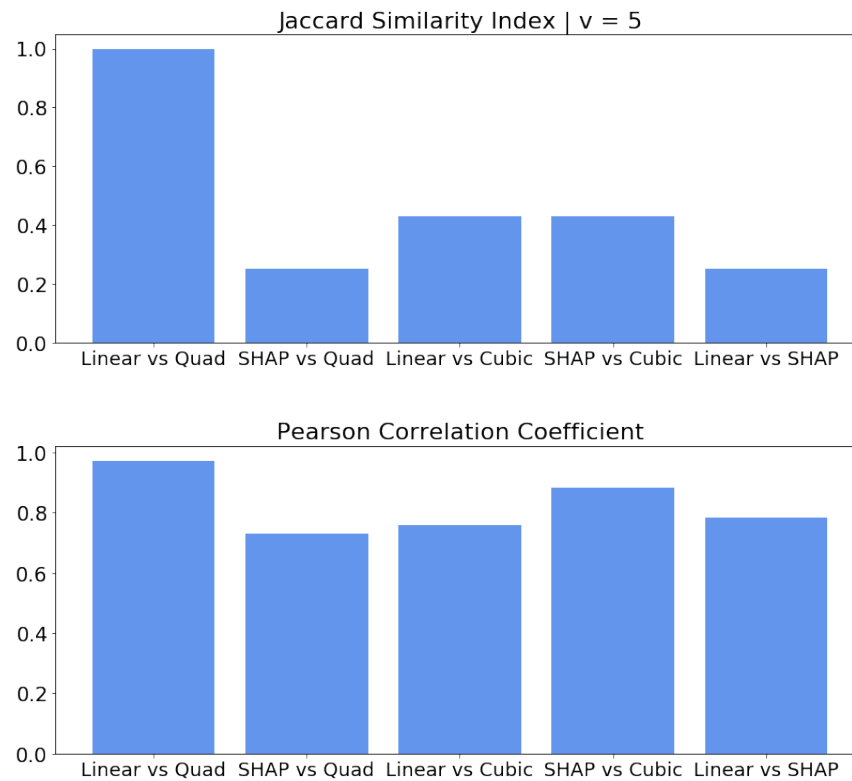


Figure 4.4: A comparison of explanations given by the linear model, quadratic model, cubic model and the SHAP model for a patient instance.

patient specificity, thus we produce an adaptive framework at the local level through adaptive polynomials.

We identify that the uni-variate approach shows single feature interaction with the local model, and although predictions are reliant on the kernel and localised feature perturbations which can lead to explanation instability, with ongoing research being focused in this area for the extension of LIME, we instead focus on improving the interpretable local model by adapting explanations to each local instance to increase local specificity. Extending upon this, the interpretable comparison with the logistic regression model poses questions towards the disagreement of explanations, to further analyse this, we will consider statistical significance against the explanations given. We acknowledge the problem of potential polynomial overfitting despite regularization. Further research will be carried out in order to approach the addressed issues and expand upon the framework.

Chapter 5

Counterfactual Integrated Gradients

Contents

5.1	Introduction	69
5.2	Integrated Gradients	72
5.3	Counterfactual Explanations	72
5.4	Method	74
5.5	Results	78
5.6	Conclusion	83

5.1 Introduction

Explainable Artificial Intelligence (XAI) has been widely applied in medicine in recent years [SP22]. Application areas of XAI vary from [PRMT20] for the early detection of Parkinson’s disease, [DIKT19] for the diagnosis of Alzheimer’s disease, [DFFS23, MUKK20, LSG⁺19] for variations of cancer and [WLW20, KED⁺21] for COVID. Whilst causal effects have been long explored in simple models for medicine [Höf05], the XAI field exemplifies the need for transparency in less interpretable models. In this chapter we exemplify the counterfactual approach of explainability with an emphasis on model-specific explanations. It is clear the approximation approach of PALE introduced in the previous chapter is limited. While model-agnostic methods frequently approximate decision boundaries, developing XAI techniques tailored for neural networks that leverage the network architecture can be advantageous due to the capacity of being able to approximate any continuous function. This assertion is rooted in the *universal approximation theorem* [BG20, MP99], which subsequent research has supported by demonstrating that ANNs can approximate any continuous function within a bounded domain using a finite number of nodes, only two layers and sufficient choice of activation function [MP99, LL20]. This elucidates the significant potential of ANNs. Similarly, customizing approaches tailored to specific models minimizes potential errors of XAI

bias and the possible imperfect approximation of decision boundaries inherent in model-agnostic methods. Therefore, in this part of the thesis I redirect the attention to designing model-specific approaches.

It follows that we can ascertain answers to specific questions by modelling XAI approaches to answer the desired question. Hereinafter in this chapter we explore the counterfactual perspective of XAI. Counterfactuals explore “what-if?” scenarios, enabling the evaluation of causal effects in intervention studies. An example of this is the Average Treatment Effect in medicine, where subsets of patients are assessed with and without a specific treatment [NW23]. As it is practically impossible to observe an individual both with and without an intervention, estimations are made by analyzing cohorts that received the treatment and those that did not. Counterfactual methods and their explanations play a crucial role in determining how changes to an individual can alter predicted outcomes and identify the features that influence such changes [GCHW23]. Consequently, counterfactual analysis allows us to explore a single patient in multiple hypothetical states simultaneously. Importantly, the counterfactual methodology is not limited to discrete changes; it can also be applied to analyze the impact of continuous features.

The counterfactual explanation algorithm *Diverse Counterfactual Explanations* (DiCE) [MST20] and Wachter et al. (Wachter) [WMR18], provide the generation of counterfactuals for a given instance. These counterfactual algorithms can be used to generate an explanation that aims to answer the question:

What changes can be made to an instance, in order to achieve the desired output?

For a prediction instance, these methods provide an explanation using a similar instance of the opposing class as a counterfactual example. The explanation can be example based (it already exists in the data) or generative (constructed with the prediction model). Generally speaking, counterfactual explanations are sample instances that are similar in feature values to the prediction instance but different in prediction; so one can observe what can be changed to obtain a different outcome.

Alternatively, feature-attribution methods provide another form of explanation this is seen with state-of-the-art methods Local Interpretable Model-Agnostic Explanations (LIME) [RSG16], SHapley Additive eXplanations (SHAP) [LL17] and Integrated Gradients (IG) [STY17]. Therefore, utilising feature-attribution in the scope of counterfactuals, will allow for the observation of what features when changed, positively or negatively attribute towards the counterfactual class.

This chapter proposes *Counterfactual Integrated-Gradients* (CF-IG), a technique utilizing the IG feature-attribution method. The integration of feature-attribution with counterfactual examples enables the analysis of positive and negative causal relations between the independent and dependent variable(s), combining the benefits of example-based counterfactuals and feature-attribution techniques. Figure 5.1 illustrates the proposed method. Generally speaking, we consider counterfactual examples to provide insight as to how an instance should look to obtain an alternate outcome; and feature-

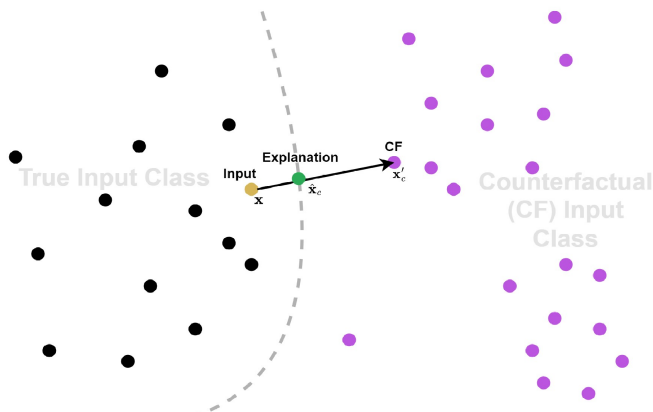


Figure 5.1: Illustrative example of CF-IG. Given an input \mathbf{x} , its linear interpolation to its nearest counterfactual example $\hat{\mathbf{x}}'_c$ in the dataset is shown. The explanation $\hat{\mathbf{x}}_c$ produced by CF-IG is the point crossing the decision boundary (the dotted line) on this interpolation. CF-IG also produces feature-attribution values for its explanations.

attribution in addendum to this can illustrate how each feature independently affected the predicted outcome.

To establish the validity of CF-IG, we compare it against a list of theoretical properties proposed in the literature, specifically focusing on [ABN22] and [VDH20], as these are recent publications on the theoretical underpinnings of XAI. We also compare CF-IG with existing counterfactual methods, namely DiCE and Wachter, and demonstrate that CF-IG is the only method satisfying these theoretical properties.

The key contributions of this paper are to:

1. introduce a new counterfactual generation method, CF-IG, that also considers feature attribution techniques;
2. demonstrate CF-IG satisfying many theoretical properties of XAI methods proposed in the literature;
3. empirically evaluate CF-IG's performance against existing counterfactual explanation methods on Electronic Health Records (EHR) datasets.

The rest of this chapter is organised as follows: Section 5.2 provides a background on the Integrated Gradients method. Section 5.3 introduces counterfactual explanations and associated properties. Section 5.4 introduces the proposed approach and associative metrics to illustrate the performance of the proposed approach. Section 5.5 evaluates the proposed approach against the introduced metrics and the identified properties.

5.2 Integrated Gradients

Integrated Gradients (IG) [STY17] is a model-specific XAI method aimed to produce local feature attribution explanations. IG assumes that the black-box model is differentiable e.g. Neural Networks. Access to differentiable models allows evaluation of the associated gradients. The insight into the model via gradients can be used for explanations. Therefore, consider an instance \mathbf{x} , an explanation along the j^{th} dimension is defined by IG as:

$$E_j(\mathbf{x}) = (x^j - x'^j) \times \int_{\alpha=0}^1 \frac{\partial F(\mathbf{x}' + \alpha(\mathbf{x} - \mathbf{x}'))}{\partial x^j} d\alpha. \quad (5.1)$$

Let $F : \mathbb{R}^J \rightarrow \mathbb{R}$ represent the differentiable function that produces the prediction probability for the target class. We denote f as the classifier function that assigns the class label to \mathbf{x} . Specifically, if F yields a prediction probability such that $F(\mathbf{x}) \geq \tau$: $\tau \in [0, 1]$ for a class y' , then $f(\mathbf{x})$ assigns the class y' , otherwise, it assigns class y . It should be noted that throughout this chapter, τ is set to 0.5.

IG is inspired by path integrals, by taking the path of least action (straight line interpolations) between an all zero baseline \mathbf{x}' of J features and the instance \mathbf{x} . The average gradients along the path represent the attribution at point \mathbf{x} .

5.3 Counterfactual Explanations

In addition to feature-attribution methods such as IG, counterfactual explanation methods such as DiCE [MST20], Wachter [WMR18] and others [LOHdR19, DCL⁺18], are another subset of XAI methods. We can formulate these approaches as giving explanations as follows:

Definition 5.1 (Counterfactual Explanation) Given a dataset $X \in \mathbb{R}^{N \times J}$, an instance $\mathbf{x} = \langle x^1, \dots, x^J \rangle \in X$, and a black-box model f that predicts the class labels of instances. A *counterfactual explanation* to an instance \mathbf{x} is another instance $\mathbf{x}_c \in \mathbb{R}^J$ such that $f(\mathbf{x}) \neq f(\mathbf{x}_c)$.

In this work, we consider two types of counterfactual explanations, *example-based* and *generative* counterfactual explanations, produced by their corresponding methods respectively, defined as follows.

Definition 5.2 (Example-Based Counterfactual) Given a dataset X , an *example-based counterfactual method* φ is a function that for $\mathbf{x} \in X$, $\varphi(\mathbf{x})$ is a counterfactual explanation to \mathbf{x} and $\varphi(\mathbf{x}) \in X$.

Definition 5.3 (Generative Counterfactual) Given a dataset X , a *generative counterfactual method* ϑ is a function that for $\mathbf{x} \in X$, $\vartheta(\mathbf{x})$ is a counterfactual explanation to \mathbf{x} and $\vartheta(\mathbf{x}) \notin X$.

From counterfactual explanations, which are also instances, we can define *counterfactual feature attributions* to capture the relative importance of features that distinguish an instance and its counterfactual counterpart. In other words, counterfactual feature attributions help us to answer the question

“if we were to turn an instance \mathbf{x} to a different outcome, what are the features we should focus on?”

We define counterfactual feature attribution as follows.

Definition 5.4 (Counterfactual Feature Attribution) A *counterfactual feature attribution method* is a function $\Phi : \mathbb{R}^J \times \mathbb{R}^J \rightarrow \mathbb{R}^J$ that takes an instance \mathbf{x} and one of its counterfactual examples \mathbf{x}_c to return feature attribution values $\langle \phi^1, \dots, \phi^J \rangle$.

Theoretical analysis of XAI algorithms enables the assessment of property satisfiability for XAI methods, with desired properties providing guidelines for the development of such methods. Properties we will study in this work are *implementation invariance* [STY17], *relevance*, *representativeness*, and *success* [ABN22]. They are informally introduced as follows:

1. Implementation Invariance. For functionally equivalent XAI methods, the explanation returned is the same.
2. Relevance. Explanations should be possible on unseen instances.
3. Representativeness. Prediction probabilities of the black-box model should be reproducible by the explainer.
4. Success. An explanation is returned for every instance.

In addition, under the same pretense as the representativeness property, we propose the *counterfactual representativeness* (CR) property, which states that for an instance \mathbf{x} and its counterfactual \mathbf{x}'_c must be reproducible by a counterfactual feature attribution method. Formally,

Property 1 (Counterfactual Representativeness). *Given a black-box function F that assigns a prediction probabilities, an instance \mathbf{x} , a counterfactual example \mathbf{x}'_c , a counterfactual feature-attribution method Φ has counterfactual representativeness, if and only if*

$$|F(\mathbf{x}) - F(\mathbf{x}'_c)| = \left| \sum_{j=1}^J \phi^j(\mathbf{x}, \mathbf{x}'_c) \right|. \quad (5.2)$$

Intuitively, Φ has counterfactual representativeness when the sum of all of its feature attributions is the difference between prediction probabilities of the instance and its counterfactual explanation.

Next, we introduce the concept of *counterfactual monotonicity* (CM) as another desirable property as follows.

Property 2 (Counterfactual Monotonicity). *Given a black-box function F , an instance \mathbf{x} , and two counterfactual instances $\mathbf{x}'_{c,1}$ and $\mathbf{x}'_{c,2}$. A counterfactual feature-attribution method Φ is counterfactually monotonic*

$$\begin{aligned} & \text{if } |F(\mathbf{x}) - F(\mathbf{x}'_{c,1})| \leq |F(\mathbf{x}) - F(\mathbf{x}'_{c,2})|, \\ & \text{then } \left| \sum_{j=1}^J \phi^j(\mathbf{x}, \mathbf{x}'_{c,1}) \right| \leq \left| \sum_{j=1}^J \phi^j(\mathbf{x}, \mathbf{x}'_{c,2}) \right|. \end{aligned}$$

In essence, a counterfactual feature attribution method is CM if, for two counterfactual explanations $\mathbf{x}_{c,1}$ and $\mathbf{x}_{c,2}$ of the same instance \mathbf{x} , the difference in prediction probability between $F(\mathbf{x}_{c,1})$ and $F(\mathbf{x})$ is closer than or equal to the difference between $F(\mathbf{x}_{c,2})$ and $F(\mathbf{x})$, then the total attribution associated with transitioning to counterfactual $\mathbf{x}_{c,1}$ must be less than or equal to the attribution assigned to $\mathbf{x}_{c,2}$. It is easy to see that counterfactual representativeness implies counterfactual monotonicity.

5.4 Method

5.4.1 Approach

With example-based counterfactual methods and IG, our CF-IG method employs a technique that involves observing the path integral concerning counterfactual examples, generating both generative counterfactual explanation as well as counterfactual feature attribution explanation. This process involves a series of iterative linear interpolations until the decision boundary for the desired class is crossed. The instance that barely crossed the decision boundary is the generative counterfactual explanation, whereas the extrapolation of the gradient toward the counterfactual class is the feature attribution explanation. By utilizing the path integral approach, we ensure that the property of implementation invariance is maintained for functionally equivalent neural networks, when the same set of hyper-parameters is employed.

Intuitively, we want an instance $\mathbf{x} \in X$ and its counterfactual explanation \mathbf{x}'_c to be “close”. To this end, we let all example-based counterfactual explanation be the set $C = \{\mathbf{x}_c \in X | f(\mathbf{x}_c) \neq f(\mathbf{x})\}$ and define *nearest counterfactual neighbour (NCFN)* as:

$$\mathbf{x}'_c = \arg \min_{\mathbf{x}_c \in C} \delta(\mathbf{x}, \mathbf{x}_c), \quad (5.3)$$

where δ is an arbitrary distance function between two instances.

From NCFN, we can compute generative counterfactual explanations that are closer to the decision boundary. Formally,

Definition 5.5 Given a model f , for an instance \mathbf{x} with its NCFN \mathbf{x}'_c , let γ be the line defined by \mathbf{x} and \mathbf{x}'_c . The *generative counterfactual example* of \mathbf{x} is

$$\hat{\mathbf{x}}_c = \arg \min_{\mathbf{x}_c \in \gamma, f(\mathbf{x}_c) \neq f(\mathbf{x})} \delta(\mathbf{x}, \mathbf{x}_c). \quad (5.4)$$

Intuitively, $\hat{\mathbf{x}}_c$ is the point on γ that barely crosses the decision boundary. It is clear that $\hat{\mathbf{x}}_c$ is a counterfactual explanation to \mathbf{x} . To obtain a feature attribution explanation from $\hat{\mathbf{x}}_c$, we simply revise IG as follows:

$$E_j(\mathbf{x}, \hat{\mathbf{x}}_c) = (\hat{x}_c^j - x^j) \times \int_{\alpha=0}^1 \frac{\partial F(\mathbf{x} + \alpha(\hat{\mathbf{x}}_c - \mathbf{x}))}{\partial x^j} d\alpha. \quad (5.5)$$

This is calculated with the Riemann integration with a hyper-parameter K for the j^{th} dimension, with

$$E_j^{CF}(\mathbf{x}, \hat{\mathbf{x}}_c; K) = (\hat{x}_c^j - x^j) \times \sum_{k=1}^M \frac{\partial F(\mathbf{x} + \frac{k}{K} \times (\hat{\mathbf{x}}_c - \mathbf{x}))}{\partial x^j} \times \frac{1}{K},$$

approximating $E_j(\mathbf{x}, \hat{\mathbf{x}}_c)$, for each feature (dimension) j in both \mathbf{x} and $\hat{\mathbf{x}}_c$.

In short, CF-IG produces an counterfactual feature attribution explanation by integrating with respect to the partial derivative over linear interpolations between the instance and its generative counterfactual instance. The interpolation stops when the prediction changes (when the decision boundary is reached). We let the counterfactual feature attribution $\Phi_{\text{CF-IG}}$ be a vector of all attributions $E_j^{CF}(\cdot; \cdot)$, such that

$$\Phi_{\text{CF-IG}}(\mathbf{x}, \hat{\mathbf{x}}_c) = \langle E_1^{CF}(x^1, \hat{x}_c^1; K), \dots, E_J^{CF}(x^J, \hat{x}_c^J; K) \rangle. \quad (5.6)$$

The complete counterfactual feature attribution explanation for all features is such that

$$E^{CF}(\mathbf{x}, \hat{\mathbf{x}}_c; K) = \sum_{j=1}^J E_j^{CF}(x^j, \hat{x}_c^j; K). \quad (5.7)$$

Thanks to the Riemann integration, the following property holds with respect to $E^{CF}(\mathbf{x}, \hat{\mathbf{x}}_c; K)$.

Proposition 1. $\lim_{M \rightarrow \infty} E^{CF}(\mathbf{x}, \hat{\mathbf{x}}_c; K) = \tau - F(\mathbf{x})$.

Proof. For a straight line path γ between \mathbf{x} and \mathbf{x}'_c , we let $[\mathbf{x}, \mathbf{x}'_c]$ be the interval that the path γ traverses. A function ψ maps points in this interval to some α with $\psi(\alpha) = \langle x^1(\alpha), \dots, x^J(\alpha) \rangle$. Thus integrating with respect to a change in α over \mathbf{x} to \mathbf{x}'_c will represent a straight line interpolation from \mathbf{x} to \mathbf{x}'_c at each point α along γ .

$$\begin{aligned} \int_{\gamma} \nabla F \cdot d\psi &= \int_{\mathbf{x}}^{\mathbf{x}'_c} \nabla F(\psi(\alpha)) \cdot \psi'(\alpha) d\alpha \\ &= \int_{\mathbf{x}}^{\mathbf{x}'_c} \frac{d}{d\alpha} F(\psi(\alpha)) d\alpha \\ &= F(\psi(\mathbf{x}'_c)) - F(\psi(\mathbf{x})) \\ &= F(\mathbf{x}'_c) - F(\mathbf{x}), \end{aligned}$$

where ∇F is the gradient of F , ψ' is the first derivative of ψ at α . Since $\hat{\mathbf{x}}_c$ is at the decision boundary, we know $F(\hat{\mathbf{x}}_c) = \tau$. Considering the point $\hat{\mathbf{x}}_c$ exists on the path γ from \mathbf{x} to \mathbf{x}'_c , then following rules of additivity for integrals we have:

$$\begin{aligned} \int_{\mathbf{x}}^{\hat{\mathbf{x}}_c} \frac{d}{d\alpha} F(\psi(\alpha)) d\alpha + \int_{\hat{\mathbf{x}}_c}^{\mathbf{x}'_c} \frac{d}{d\alpha} F(\psi(\alpha)) d\alpha \\ = \int_{\mathbf{x}}^{\mathbf{x}'_c} \frac{d}{d\alpha} F(\psi(\alpha)) d\alpha \end{aligned}$$

rearranging the equation, we can solve for the intermediary point that occurs at τ , giving an explanation as K approaches infinity:

$$\begin{aligned} E^{CF}(\mathbf{x}, \hat{\mathbf{x}}_c; K) &= \\ \int_{\mathbf{x}}^{\mathbf{x}'_c} \frac{d}{d\alpha} F(\psi(\alpha)) d\alpha - \int_{\hat{\mathbf{x}}_c}^{\mathbf{x}'_c} \frac{d}{d\alpha} F(\psi(\alpha)) d\alpha \\ &= \int_{\mathbf{x}}^{\hat{\mathbf{x}}_c} \frac{d}{d\alpha} F(\psi(\alpha)) d\alpha \\ &= \tau - F(\mathbf{x}) \end{aligned}$$

Therefore, it is easy to see that as K approaches infinity, we have $E^{CF}(\mathbf{x}, \hat{\mathbf{x}}_c; K) = \tau - F(\mathbf{x})$ \square

5.4.2 Utilising Other Counterfactual Generative Methods

Up to this point, we have introduced an approach for CF-IG to generate counterfactual explanations by leveraging example-based counterfactual explanations through NCFN. Alternatively, it is possible to employ other counterfactual methods ϑ to create counterfactual explanations and then apply CF-IG to them. We demonstrate that CF-IG will generate counterfactual explanations that are either equidistant from or closer to the original instance \mathbf{x} compared to explanations produced solely by ϑ . Additionally, given that certain counterfactual explanation methods might not inherently offer feature importance, our CF-IG method can consistently generate feature attribution explanations for these instances.

Proposition 2. *The path from \mathbf{x} to $\hat{\mathbf{x}}_c$ produced by CF-IG is shorter than or equal to the length of the path from \mathbf{x} to the NCFN point \mathbf{x}'_c .*

Proof. This is trivially true as the path connecting \mathbf{x} , $\hat{\mathbf{x}}_c$ and \mathbf{x}'_c is a straight line; and $\hat{\mathbf{x}}_c$ is between the other two points. \square

We can generalize Proposition 2 to show that for a counterfactual explanation \mathbf{x}'_c found by any method, CF-IG finds $\hat{\mathbf{x}}_c$ such that $\Phi(\mathbf{x}, \hat{\mathbf{x}}_c) \leq \Phi(\mathbf{x}, \mathbf{x}'_c)$, formally:

Proposition 3. *For an instance \mathbf{x} and its counterfactual explanation, it holds that*

$$\lim_{M \rightarrow \infty} \mathbf{x}'_c, E^{CF}(\mathbf{x}, \hat{\mathbf{x}}_c; K) \leq E^{CF}(\mathbf{x}, \mathbf{x}'_c; K). \quad (5.8)$$

Proof. Direct from proposition 1, as:

$$\int_{\mathbf{x}}^{\mathbf{x}'_c} \nabla F(\psi(\alpha)) \cdot \psi'(\alpha) d\alpha = F(\mathbf{x}'_c) - F(\mathbf{x})$$

and:

$$\int_{\mathbf{x}}^{\hat{\mathbf{x}}_c} \nabla F(\psi(\alpha)) \cdot \psi'(\alpha) d\alpha = \tau - F(\mathbf{x})$$

since $\tau \leq F(\mathbf{x}'_c)$, as M approaches infinity it is clear that:

$$\begin{aligned} \int_{\mathbf{x}}^{\hat{\mathbf{x}}_c} \nabla F(\psi(\alpha)) \cdot \psi'(\alpha) d\alpha &\leq \int_{\mathbf{x}}^{\mathbf{x}'_c} \nabla F(\psi(\alpha)) \cdot \psi'(\alpha) d\alpha \\ \implies \tau - F(\mathbf{x}) &\leq F(\mathbf{x}'_c) - F(\mathbf{x}) \\ \implies E^{CF}(\mathbf{x}, \hat{\mathbf{x}}_c; K) &\leq E^{CF}(\mathbf{x}, \mathbf{x}'_c; K) \end{aligned}$$

Therefore, the attribution and prediction probability of a point produced by CF-IG, is closer to the origin \mathbf{x} and the associated attribution and prediction probability. \square

Remark 5.6 Explanations given by CF-IG are bi-directional as a path γ is invertible:

$$\int_{-\gamma} \nabla F \cdot d\psi = - \int_{\gamma} \nabla F \cdot d\psi.$$

Therefore, we can similarly interpret the explanations from both directions, going from a factual class to a counterfactual class as well as going from a counterfactual class to a factual class.

5.4.3 Evaluation

In order to evaluate the performance of CF-IG, we propose metrics that measure against the listed properties. To determine the satisfiability of the implementation invariance property, a metric for the consistency of explanations over a set number of R runs is proposed. We let consistency be measured using the Root Mean Squared Error (RMSE) from the first generated explanation:

$$\text{RMSE}(\Phi, \mathbf{x}_i, \hat{\mathbf{x}}_c^i; R) = \sqrt{\frac{\sum_{r=1}^{R-1} (\Phi^0(\mathbf{x}_i, \hat{\mathbf{x}}_c^i) - \Phi^r(\mathbf{x}_i, \hat{\mathbf{x}}_c^i))^2}{R-1}}.$$

Here, Φ is a counterfactual feature attribution method. We let Φ^0 be the initial run of R total runs, and Φ^r be the r^{th} iteration of $\langle \Phi^1, \dots, \Phi^{R-1} \rangle$ iterations, here we let i^{th} counterfactual example $\hat{\mathbf{x}}_c^i$ be found through equation 5.4 with respect to its corresponding \mathbf{x}_i instance. We can then deduce across N instances, where

$X = \langle \mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_N \rangle$ the consistency of the counterfactual feature attribution method, such that:

$$\text{consistency} = \frac{1}{N} \sum_{i=1}^N \text{RMSE}(\Phi, \mathbf{x}_i, \hat{\mathbf{x}}_c^i; R)$$

Similarly, we consider two variants of proximity measures for counterfactual examples that are produced by different counterfactual methods, namely: proximity_{l_2} and $\text{proximity}_{\text{cosine}}$. Here, proximity_{l_2} uses the Euclidean distance metric to compare samples and $\text{proximity}_{\text{cosine}}$ uses cosine distance. The l_2 distance (dist_{l_2}) between two vectors \mathbf{x} and \mathbf{y} is the Euclidean distance given by:

$$\text{dist}_{l_2}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{j=1}^J (x^j - y^j)^2}.$$

Therefore, we define $\text{proximity}_{l_2}(X, Y)$ over two sets X and Y , each containing N instances as:

$$\text{proximity}_{l_2}(X, Y) = \frac{1}{N} \sum_{i=1}^N \text{dist}_{l_2}(\mathbf{x}_i, \mathbf{y}_i).$$

The cosine distance (CS) between two vectors \mathbf{x} and \mathbf{y} is:

$$\text{CS}(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2};$$

We define $\text{proximity}_{\text{cosine}}(X, Y)$ over two sets X and Y as:

$$\text{proximity}_{\text{cosine}}(X, Y) = \frac{1}{N} \sum_{i=1}^N \text{CS}(\mathbf{x}_i, \mathbf{y}_i).$$

5.5 Results

5.5.1 Property Satisfiability

Property satisfiability explicates the underlying theoretical guarantees of a given method, therefore we evaluate the satisfiability of CF-IG, DiCE and Wachter.

5.5.1.1 CF-IG

The property satisfiability of CF-IG is shown in Table 5.1.

Corollary 5.7 Counterfactual feature attribution explanations produced by CF-IG satisfy counterfactual representativeness.

Proof. As a direct consequence of proposition 1 counterfactual representativeness holds for CF-IG. \square

Corollary 5.8 Feature attribution assigned by CF-IG satisfies representativeness.

Proof. Given Proposition 1 and the definition of the CF-IG method given in equation 5.5. By expanding and evaluating each part of the equation that defines CF-IG separately, we have:

$$\begin{aligned} \sum_{j=1}^J E_j^{CF}(\mathbf{x}, \hat{\mathbf{x}}_c; K) = & \\ & \underbrace{\sum_{j=1}^J \hat{x}_c^j \times \sum_{k=1}^K \frac{\partial F(\mathbf{x} + \frac{k}{K} \times (\hat{\mathbf{x}}_c - \mathbf{x}))}{\partial x^j}}_{F(\hat{\mathbf{x}}_c)} \times \frac{1}{K} \\ & - \underbrace{\sum_{j=1}^J x^j \times \sum_{k=1}^K \frac{\partial F(\mathbf{x} + \frac{k}{K} \times (\hat{\mathbf{x}}_c - \mathbf{x}))}{\partial x^j}}_{F(\mathbf{x})} \times \frac{1}{K}. \end{aligned}$$

therefore, $F(\mathbf{x})$ and $F(\hat{\mathbf{x}}_c)$ are reproducible and representativeness (analogous to the efficiency axiom in [LL17]) of both the origin instance \mathbf{x} and $\hat{\mathbf{x}}_c$ hold. \square

Corollary 5.9 CF-IG satisfies counterfactual monotonicity.

Proof. Direct from proposition 3, it is clear that counterfactual monotonicity is satisfied. \square

Proposition 4. *If there are two non-empty classes than an explanation given by CF-IG will be a success and have relevance.*

Proof. Given an instance exists in both classes a NCFN will always be found from equation 5.3, thus an explanation can always be generated through using an instance produced by equation 5.4 as an input to equation 5.5. As we can return an explanation $E(\mathbf{x}, \hat{\mathbf{x}}_c)$ in the form of counterfactual feature attribution an explanation will be a success and maintain relevance. \square

5.5.1.2 DiCE

Table 5.1 shows that DiCE does not satisfy success, this is due to restrictions on producing on valid counterfactual examples, if permutations do not cross the decision bound, then a counterfactual will not be returned. Similarly, DiCE does not reproduce prediction probabilities of the black-box model, therefore does not satisfy representativeness or counterfactual representativeness. From empirical evidence in Section 5.5.2, DiCE does not provide consistent explanations and thus DiCE does not satisfy Implementation Invariance. On the other hand, DiCE does provide a diverse

Table 5.1: Comparison of the property satisfiability of counterfactual methods. Here ‘✓’ indicates the property is satisfied, ‘✗’ indicates the property is not satisfied and ‘-’ indicates that the property is not applicable or cannot directly be evaluated by the method.

Property	Imp. Invar.	Relev.	Rep.	Success	CM	CR
NCFN	✓	✓	-	✓	-	-
DiCE	✗	✓	✗	✗	-	✗
Wachter	✗	✓	-	✗	-	-
CF-IG	✓	✓	✓	✓	✓	✓
DiCE+CF-IG	✗	✓	✓	✗	✓	✓
Wachter+CF-IG	✗	✓	✓	✗	✓	✓

set of counterfactual examples and is useful in producing a diverse set of observable examples. Similarly, DiCE also is capable of producing explanations on unseen instances and therefore satisfies relevance.

5.5.1.3 Wachter

The Wachter method is a generative counterfactual method that produces counterfactual examples. The counterfactual examples that are produced by Wachter are inconsistent as shown in Tables 5.3 and 5.4 thus empirically does not satisfy implementation invariance. It is also not guaranteed that Wachter generates a counterfactual example and therefore does not satisfy success. It is possible to generate counterfactual examples on unseen instances meaning that relevance holds.

5.5.1.4 CF-IG+Wachter & CF-IG+DiCE

Here we evaluate the property satisfiability of CF-IG when combined with counterfactual examples that are generated by Wachter and DiCE.

Corollary 5.10 CF-IG with the addendum of a generative counterfactual example will satisfy relevance, representativeness, counterfactual representativeness and counterfactual monotonicity.

Proof. Replacing the counterfactual example generated by NCFN, namely \mathbf{x}'_c , by using a counterfactual example that is generated using any generative counterfactual method as an input to equation 5.5, will inherently satisfy relevance, representativeness, counterfactual representativeness and counterfactual monotonicity. This is not true for implementation invariance or success as the generated counterfactuals are not the same on each run as evident from Tables 5.2, 5.3 and 5.4, also there is no guarantee that Wachter or DiCE return a counterfactual for a given instance \mathbf{x} . \square

Table 5.2: Comparison of the consistency given across a collection of counterfactual methods that produce attribution/importance values. We observe that the CF-IG method produces consistent explanations given $N = 100$ and $R = 10$.

Consistency	Skin	Breast	Rectal	Lymphoma
DiCE	0.25	0.18	0.21	0.23
CF-IG	0	0	0	0
DiCE+CF-IG	1.59e-08	0.001	0.001	0.017
Wachter+CF-IG	0.0002	8.22e-05	0.0001	0.0003

5.5.2 Experiment

The experiments in this chapter use data derived from the Simulacrum¹ (introduced in Chapter 3). We isolate cohorts of patients based on their ICD-10 code, to generate datasets where patients are separated into the binary sets ≥ 6 months and < 6 months survival. This allows us to pose the question:

Given a set of patient features, which features influence the change in the prediction?

Thus to evaluate consistency when evaluating the aforementioned question, the explanations are produced over a subset of 100 cancer patients (of a total 1750), for 10 iterations across 4 datasets (see Table 5.2). Here, we only consider DiCE, CF-IG, Wachter+CF-IG and DiCE+CF-IG as these methods produce feature-attribution values.

Extrapolating on this, we explore the addendum of CF-IG that utilises generative counterfactual examples. Here we only consider DiCE and Wachter due to the algorithm accessibility for PyTorch and tabular data. Conceptually from proposition 3, CF-IG will produce counterfactuals closer or equal to the distance of any counterfactual example method.

The term *Proximity* that has been used in work surrounding counterfactual examples [VDH20]. In this work, we consider proximity to be the average distance from each input sample to each of their corresponding counterfactual examples to be a measure of proximity, a quantifiable metric. We measure counterfactual examples using proximity_{cosine} and proximity_{l2} (see Tables 5.3 and 5.4), we note that the instances that are compared have been normalised for proximity_{l2} as this is not scale invariant. Here, we observe when the CF-IG method is used in concatenation with DiCE and Wachter, the interpolated example from CF-IG is closer than DiCE and Wachter independently.

Note that the larger aberration in consistency of the DiCE method is due to how feature-importance is assigned. DiCE can often produce importance values of 1 across multiple instances, and it is common that CF-IG produces feature-importance values $\ll 1$ per feature. Similarly, DiCE promotes *diversity*, where further information is given in [MST20].

¹<https://simulacrum.healthdatainsight.org.uk/>

5. Counterfactual Integrated Gradients

Table 5.3: Comparison of the proximity between original and counterfactual instances using the cosine distance between vectors. This is experimented over 100 instances on each dataset.

$\text{Proximity}_{\text{cosine}}$	Skin	Breast	Rectal	Lymph
NCFN	0.00030	0.001	0.001	0.009
DiCE	0.00239	1.21e-06	2.12e-06	0.0002
Wachter	0.00020	0.0001	0.003	7.95e-05
CF-IG	0.00028	0.0007	0.0012	0.009
DiCE+CF-IG	0.00027	5.78e-08	1.81e-06	6.07e-05
Wachter+CF-IG	0.0001	1.55e-05	0.00027	1.75e-05

Table 5.4: Comparison of the proximity between original and counterfactual instances using l_2 distance. This is experimented over 100 instances on each dataset.

Proximity_{l_2}	Skin	Breast	Rectal	Lymph
NCFN	0.00247	0.005	0.0051	0.0135
DiCE	0.00241	0.0007	0.0002	0.002
Wachter	0.0020	0.003	0.008	0.0012
CF-IG	0.00239	0.004	0.0050	0.0134
DiCE+CF-IG	0.0023	0.0005	0.0001	0.0011
Wachter+CF-IG	0.0017	0.002	0.002	0.0005

5.5.3 Explanations

In this section, we demonstrate explanations generated by our methods and propose a straightforward interactive form of counterfactual feature attribution visualization. We implement the standard feature attribution visualization technique, employing positive and negative (green and red) bar plots to represent the corresponding positive and negative feature attribution values (similar to [LL17, RSG16]). Furthermore, to depict the counterfactual, we augment each feature name on the y-axis with the respective change required to achieve the counterfactual explanation. By hovering over each attribution bar, one can access the original feature value and the new feature value for the instance’s counterfactual, along with the specific attribution value.

It is important to note that the attribution aligns with the counterfactual class; however, as mentioned earlier, inverting the explanation values provides attribution towards the original class. To enhance clarity, we present original and counterfactual feature values solely for the features that were altered, as shown in the associated tables. We note that counterfactual feature values produced by CF-IG are continuous, thus post-processing for discrete features may be necessary.

While we demonstrate this method using examples where the counterfactual is determined using the NCFN technique, it is essential to recognize that explanations can also be generated in conjunction with counterfactual examples produced by DiCE,

Wachter, or any other counterfactual example generator. In our experimental setup, we establish the decision boundary for the desired class at a prediction probability of 0.5.

In Table 5.5 we consider both the factual and counterfactual breast cancer patient instances. The explanations generated by CF-IG, CF-IG+Wachter and CF-IG+DiCE are given in Figures 5.2,5.3 and 5.4.

Table 5.5: Encoded breast cancer patients for counterfactual explanations given on for a patient instance. Here for clarity ‘ \approx ’ implies an infinitesimally small change to a feature value. Here we observe pairs of each explainer (separated with a double line) without (resp. with) the addendum of CF-IG, the closest distance to the origin instance for the pairs is in **bold**. For each pair we observe that CF-IG produces values that are closer to the original instance. Here we evaluate the normalised instances and corresponding Euclidean distance.

Features	Instance	NCFN	CF-IG	DiCE	CF-IG+DiCE	Wachter	CF-IG+Wachter
Age	78	78	78	78	78	55.58	59.65
Sex	0	0	0	1	≈ 0	0	0
Morph	6	6	6	6	6	1	2
Weight	84	52.6	57.38	84	84	69.04	71.75
Height	1.67	1.7	1.69	1.67	1.67	≈ 1.67	1.67
Dose Admin.	600	1800	1617	269.39	493.46	193.92	267.59
Outcome	0	6	5	0	0	0	0
Admin Route	5	3	3.30	5	5	1.85	2.42
Time Delay	3	3	3	3	3	1.65	1.89
Stopped Early	1	1	1	1	1	0.93	0.95
Cycle	1	3	2.69	1	1	≈ 1	≈ 1
Grade	2	2	2	2	2	1.68	1.74
Cancer Plan	0	1	0.84	0	0	0	0
Ethnicity	7	0	≈ 1	7	7	1.65	2.73
Creg Code	6	5	5.1	6	6	1.37	2.21
T Best	4	6	5.69	4	4	1.50	1.96
CNS	4	4	4	4	4	1.5	2.29
ACE	4	2	2.3	4	4	3	3.22
Performance	1	1	1	1	1	≈ 1	≈ 1
Clinical Trial	4	1	1.45	4	4	3.95	≈ 4
Normalized Distance	0	1.64	1.39	1.01	0.05	1.41	1.11

5.6 Conclusion

In conclusion, the CF-IG approach introduced in this chapter provides a robust and principled framework that upholds essential XAI principles while addressing the intricate demands of counterfactual explanations within the context of causal analysis. While path-based methodologies serve as strong theoretical foundations [FZTN22], their pragmatic implementation within feature spaces often presents challenges. Notably, CF-IG consistently delivers implementation-invariant explanations that provide insights into underlying black-box models.

The relevance of counterfactual explanations in medical applications cannot be understated. The ability to explore "what-if" scenarios, particularly by modifying features for individual patients, holds profound potential. Consider the impact of

5. Counterfactual Integrated Gradients

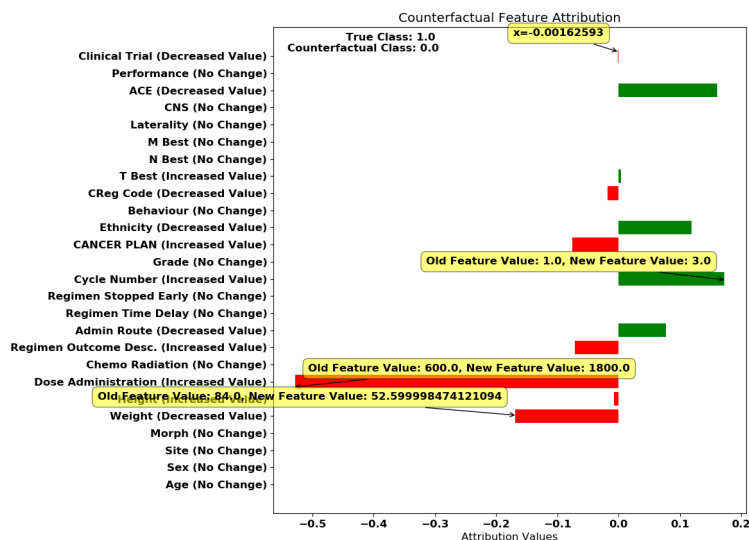


Figure 5.2: Illustrative explanation example of the CF-IG method, highlighting the bar interaction features of the explainer. Here the counterfactual method is applied to a breast cancer patient example. In this explanation we inspect the features: *T Best*, *Dose Administration* and *Weight*. Here we also observe the magnitude of the *Clinical Trial* attribution by value.

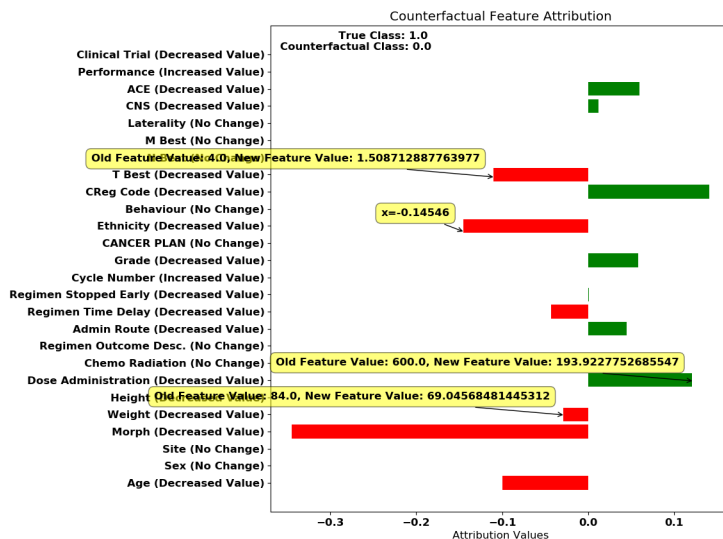


Figure 5.3: Illustrative explanation example of the CF-IG + Wachter method, highlighting the bar interaction features of the explainer. Here the counterfactual method is applied to a breast cancer patient example. In this explanation we inspect the features: *T Best*, *Dose Administration* and *Weight*. Here we also observe the magnitude of the *Ethnicity* attribution by value.

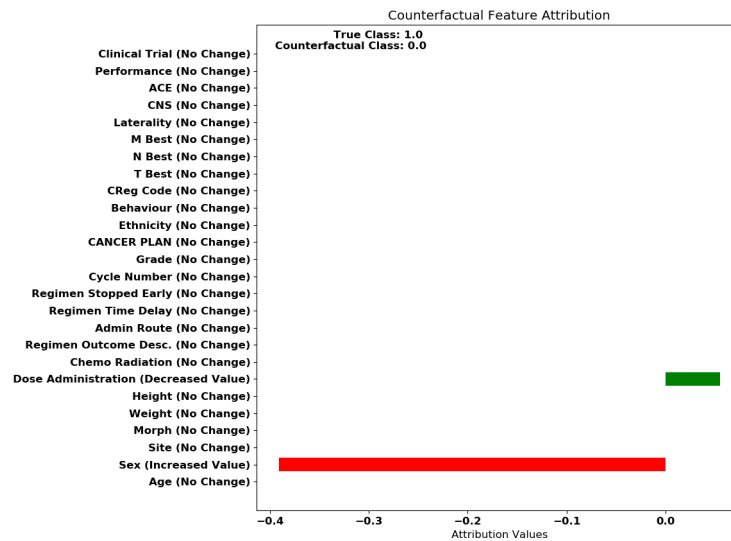


Figure 5.4: Illustrative explanation example of the CF-IG + DiCE method, highlighting the important features identified by the explainer in changing the outcome. Here the counterfactual method is applied to a breast cancer patient example.

optimizing treatment plans based on counterfactual insights, facilitating personalized interventions that account for a patient’s unique attributes and constraints.

In this pursuit, the CF-IG method stands as a promising contribution to advancing XAI in medical domains. With its solid theoretical foundation and empirical validation, CF-IG represents a pathway to more transparent and effective decision-making processes in healthcare. Through the integration of AI-driven insights with the intricacies of patient-centered care, CF-IG offers a glimpse into the future of healthcare, where innovation aligns harmoniously with practicality.

Chapter 6

Formalising Batch-Integrated Gradients for Temporal Explanations

Contents

6.1	Introduction	87
6.2	Method: Batch-Integrated Gradients	89
6.3	Properties for Explainability	91
6.4	Formal Evaluation	93
6.5	Controlled Experiments	96
6.6	Applications	98
6.7	Conclusion	101

6.1 Introduction

In this chapter we formally expand on the previous with modifications to both better define the Batch-IG approach and produce a further analysis of the method. Theoretical analysis of Integrated Gradients (IG) [STY17] has shown that IG calculates attribution in a game theoretic manor, this accounts for continuity by integrating between points. This has been compared (and proven to be equivalent in many cases) to state-of-the-art SHapley Additive exPlanation (SHAP) [LL17] values when calculating SHAP value attribution against a baseline, namely Baseline Shapley (BShap) [SN20] in [FZTN22]. However, whilst both methods are theoretically sound, the IG method is an extension on the discrete SHAP values to a continuous domain. Therefore, in this work we consider the adaptation of path integral methods as to utilise the continuity of data and inherit smooth explanations, whilst subsequently adhering to the game theoretic properties identified in the SHAP method [LL17].

Constructing XAI methods with respect to the temporal nature of data has often been overlooked [SSV21], despite that there are many problems of a temporal nature. Whilst methods such as SHAP have been applied to temporal data [DP21, VABH22], each instance is treated as independent and does not take into account relationships corresponding to time. In a non-technical short paper [DFFS23], the authors utilise the continuity of IG and conceptually introduce the idea of Batch-Integrated Gradients (Batch-IG) method to produce explanations for temporal data. In this chapter we contribute a formal introduction, comparison and evaluation of the method when measured against state-of-the-art explainers and properties of explainability in a temporal setting.

The Out of Distribution (OoD) problem is a clear concern when dealing with linear interpolations between points [PSK22]. Because the baseline for IG contains a vector of all-zeros there is no guarantee that the path is in-distribution. Therefore, in our new approach the idea is to maximise in-distribution interpolations. We propose an extension to this method to redirect paths generated by path integrated gradient methods. Taking the temporal aspect into account gives further insight into the importance of features.

Although, time-series explainability has been explored [CVDS21], local temporal explanations are often not, although they are often preferred in a medical setting [TJMG19]. Thus, the Batch-IG method enables us to answer the question “*how does the change of features over a time period alter the prediction?*”, which to our knowledge has not been explored outside of this method.

The nature of evaluating XAI methods and determining the performance of path based methods is difficult since there is no ground truth to access. To combat this, we propose a set of temporal properties and metrics, to evaluate temporal explainability by constructing controlled experiments for property satisfiability under temporal constraints. We similarly devise a controlled experiment for path evaluations under known data structure and a metric for determining path quality on real-world datasets to determine the quality of interpolations.

The objectives of this chapter are to:

1. introduce properties for temporal explainable AI methods;
2. introduce mathematical foundations and provide a formal evaluation of the Batch-IG method;
3. evaluate the proposed method with other state-of-the-art XAI methods to identify property satisfiability and against known and introduced axiomatic properties;
4. provide controlled experiments to quantitatively analyse the performance of path based methods and produce example explanations on temporal data;
5. evaluate Batch-IG on two real-world case studies.

The rest of this chapter is organised as follows: In Section 6.2, we provide a formal introduction to the Batch-Integrated Gradients method for temporal data. Section 6.3 analyses existing properties for XAI and introduces new properties for temporal data. Section 6.4 provides a formal evaluation of path based methods and discusses how

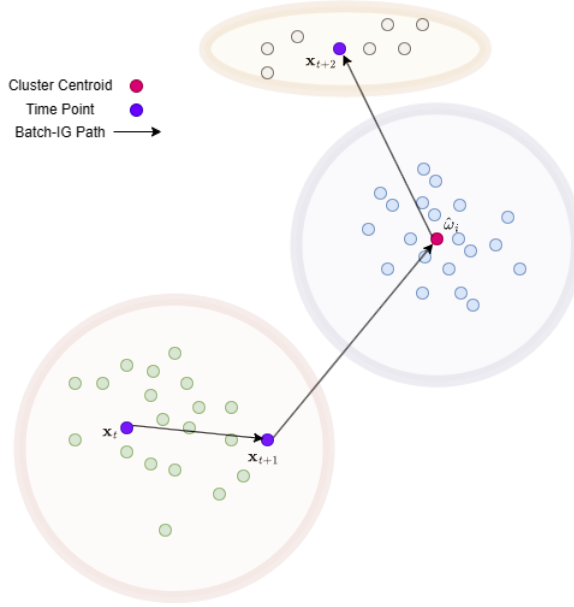


Figure 6.1: Illustrative example of Batch-IG. Given an input at time point \mathbf{x}_t , Interval-IG is demonstrated between time points, and the path for accumulated gradients over linear interpolations is shown to the destination time point \mathbf{x}_{t+2} . Here we show how Batch-IG would traverse through a cluster to minimise out-of-distribution interpolations. Batch-IG also produces feature-attribution values for its explanations.

they conform with the introduced and existing properties. In Section 6.5, we introduce controlled experiments to both, determine the validity of the paths generated by path based methods and measure the faithfulness of XAI methods. Then in Section 6.6, we apply our method to real world education and a public health scenario. The purpose of this chapter is to formally introduce and evaluate the Batch-IG method.

6.2 Method: Batch-Integrated Gradients

To construct the proposed Batch-IG method, we formulate the framework with three steps, (1) cluster the data such that there exists at least one centroid, (2) search for an *optimal* path utilising the cluster centroids (if necessary), (3) generate an explanation along a path between time points. This, in the unique setting will utilise clusters if path traversal is greatly out of distribution, in the general case, step 2 will be omitted. Thus, we only consider the direct path between time points.

Given a dataset $X = \langle \mathbf{x}, b, y \rangle$, where \mathbf{x} is an instance, y is the output prediction probability for \mathbf{x} and b is an index that refers to an instances assigned time batch χ_b . Informally, each instance indexed with the same value of b will belong to the same time-batch $\chi_b \subseteq X$. Time points are represented as $\mathbf{x}_t \in \chi_b$, where $t \in [1 : T]$. The notation $[z_1 : z_2]$ for brevity denotes a set of sequential natural numbers from z_1 to z_2 , such that $z_1 < z_2$.

$$\text{Interval-IG}_j(\mathbf{x}_t, \mathbf{x}_{t+1}) := \begin{cases} (x_{t+1}^j - x_t^j) \times \int_{\alpha=0}^1 \frac{\partial F(\mathbf{x}_t + \alpha \times (\hat{\omega}_i - \mathbf{x}_t))}{\partial x^j} + \frac{\partial F(\hat{\omega}_i + \alpha \times (\mathbf{x}_{t+1} - \hat{\omega}_i))}{\partial \hat{\omega}_i^j} d\alpha, \\ \text{if there exists a } \hat{\omega}_i; \\ (x_{t+1}^j - x_t^j) \times \int_{\alpha=0}^1 \frac{\partial F(\mathbf{x}_t + \alpha \times (\mathbf{x}_{t+1} - \mathbf{x}_t))}{\partial x^j} d\alpha, \\ \text{otherwise.} \end{cases} \quad (6.2)$$

We consider a path that either traverses through the cluster centroid nearest to the proceeding time point \mathbf{x}_{t+1} , or a path to traverse direct between time points \mathbf{x}_t and \mathbf{x}_{t+1} . Therefore, we consider a set of clusters C to be given by an arbitrary cluster algorithm (e.g. k-means) $\mathcal{C}(\cdot)$ on a instances of a dataset $\mathbf{x} \in X$, where each cluster contains a unique cluster centroid. We let $C_i \in C$ refer to a single cluster of instances. We let $\omega_i \in C_i$ be the cluster centroid of C_i . We let a set of centroids be denoted Ω , where $\Omega = \langle \omega_1, \dots, \omega_c \rangle$.

Definition 6.1 (δ -distance) The δ -distance is a function $\delta(\cdot, \cdot)$ that takes two vectors \mathbf{a} and \mathbf{b} and quantifies the distance between both vectors given any distance measurement, such that:

$$\delta(\cdot, \cdot) : \mathbf{a} \times \mathbf{b} \rightarrow \mathbb{R}. \quad (6.1)$$

From this, we need a centroid $\omega_i \in \Omega$ that can be associated with given time points \mathbf{x}_t (\mathbf{x}_{t+1} resp.), using δ we look for a ω_i that conditionally satisfies the following:

$$\hat{\omega}_i = \arg \min_{\omega_i \in \Omega} \delta(\omega_i, \mathbf{x}_{t+1}) : \delta(\omega_i, \mathbf{x}_t) \leq \delta(\mathbf{x}_{t+1}, \mathbf{x}_t) \wedge \delta(\omega_i, \mathbf{x}_{t+1}) \leq \delta(\mathbf{x}_{t+1}, \mathbf{x}_t)$$

here we let δ be the Euclidean distance in this body of work. Informally, we want to find the nearest cluster centroid that is closest to the target time point \mathbf{x}_{t+1} . The conditional addendum to equation 6.3 informally states that traversing the centroid does not yield too much of a detour from \mathbf{x}_t to \mathbf{x}_{t+1} , whilst aiming to be close to \mathbf{x}_{t+1} .

Remark 6.2 If, there are no points that satisfy the condition given in equation 6.3, we instead obtain a direct path from \mathbf{x}_t to \mathbf{x}_{t+1} .

We first introduce Interval-IG in equation 6.2 that subsequently produces an explanation over the j^{th} (feature) dimension of a time interval given by two time points \mathbf{x}_{t+1} and \mathbf{x}_t . Interval-IG serves as a fundamental component in the construction of Batch-IG, providing a single path between two time points. This characteristic allows Interval-IG to offer explanations specifically between two time points. In the context of a time batch χ_b , Interval-IG generates at least $T - 1$ paths and up to $2(T - 1)$ paths. The capability of Interval-IG to analyse attribution between two time points makes it particularly advantageous in scenarios where such analysis is desired.

Table 6.1: Qualitative evaluation of properties that are satisfied by Batch-IG, DeepSHAP, SHAP, LIME and Gradient \times Input when considering the temporal nature of data. Namely, for calculation of attribution for all methods besides Batch-IG, we take the difference in attribution between $t + 1$ and t and determine whether the properties still hold.

Property	Batch-IG	DeepSHAP	SHAP	LIME	Gradient \times Input	IG
Time Completeness	✓	✓	✓	✗	✗	✓
Time Sensitivity	✓	✗	✗	✗	✗	✗
Relevance	✓	✓	✓	✓	✓	✓
Representivity	✓	✓	✓	✗	✗	✓
Irreducibility	✓	✗	✗	✗	✗	✗
Explainability	✓	✓	✓	✓	✓	✓
Success	✓	✓	✓	✓	✓	✓

Extending upon this definition, the Batch-IG method over the j^{th} (feature) dimension of a time-batch $\chi_b \subseteq X$ is defined:

$$\text{Batch-IG}_j(\chi_b) := \sum_{t=1}^{T-1} \text{Interval-IG}_j(\mathbf{x}_t, \mathbf{x}_{t+1}) \quad (6.3)$$

Here T is the number of instances in χ_b . The explanation vector is written as $\Phi_{\text{Batch-IG}} = \langle \text{Batch-IG}_1(\chi_b), \dots, \text{Batch-IG}_J(\chi_b) \rangle$.

6.3 Properties for Explainability

The implementation of Batch-IG utilise two important properties in its explanations: *Time Completeness (TC)* and *Time Sensitivity (TS)*. Roughly speaking, TC states that: given a batch, the change to predictions between its initial and the next point can be completely explained by Batch-IG explanations, such that the sum of attributions between two points is equal to the difference prediction probability of the two points; TS states that for features that are not changed in a batch, their corresponding Batch-IG explanations are zero valued.

This is important as *independence* is assumed, the features that do not change over time should have no attribution (see Section 6.5), and similarly, the attribution should represent the change in the model between two points. Therefore, we formally introduce TC as follows:

Property 3 (Time Completeness). *Given a black-box function f , such that J is the number of features, and a explanation method $\phi^j : \mathbb{R} \rightarrow \mathbb{R}$ for each feature j , then an*

explainer is time complete with respect to a black-box function if:

$$f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) = \sum_{j=1}^J \phi^j(\mathbf{x}_{t+1}) - \sum_{j=1}^J \phi^j(\mathbf{x}_t).$$

The second property, TS, can be established as follows:

Property 4 (Time Sensitivity). *Only features that are dynamic can have a non-zero attribution. Thereby, for attribution along the j^{th} dimension where j is dynamic, we have $\phi^j(\mathbf{x}_{t+1}) - \phi^j(\mathbf{x}_t) \neq 0$ for all j that are dynamic and our function mathematically depends on.*

TS allows us to conceptualise *dynamic* and *static* features in this work, with a similar notion to *uncontrollable* and *controllable* features seen in the Controllable fActor Feature Attribution (CAFA) method [KLS⁺22] and *actionability* seen in counterfactual methods [VDH20, PSSR⁺20].

In the case of time, where we are dealing with the same object in a different state, we introduce *dynamic* (F^d) and *static* (F^s) features, similar to that introduced by the authors of [KLS⁺22], such that $F^d \cap F^s = \{\}$, we refer to

$$F^d = \{\mathbf{x}^j | x_t^j \neq x_{t+1}^j \in \chi_b\}$$

(resp. $F^s = \{\mathbf{x}^j | x_t^j = x_{t+1}^j \in \chi_b\}$),

such that, if the feature value x_t^j at one time point is not equal to the feature value of x_{t+1}^j at the next time point, then we can consider this feature dynamic (resp. static).

Properties for explainability introduced in [ABN22] allude towards a framework for certain properties that describe desirable XAI methods in the classification landscape. We give a non-formal introduction to the desired properties that we explore in this work:

1. **Success:** An explanation is considered a success, if the explainer produces an explanation for every instance.
2. **Explainability:** An explainer should provide informative explanations, and therefore an all-zero explanation is not recommended.
3. **Irreducibility:** Irreducibility states that an explanation should not contain irrelevant information.
4. **Representivity:** Representivity states that prediction probabilities should be reproducible by the explainer.
5. **Relevance:** Relevance indicates that explanations should be possible on unseen instances.

6.4 Formal Evaluation

We provide a formal evaluation, to determine property satisfiability of Batch-IG when considering the temporal aspect of the data.

Proposition 5. *Interval-IG returns the difference in prediction probabilities between two time points \mathbf{x}_t and \mathbf{x}_{t+1} .*

Proof. Consider two points along a path γ on a closed interval $[\mathbf{x}_t, \mathbf{x}_{t+1}]$, where $F(\mathbf{x}_t)$ (resp. $F(\mathbf{x}_{t+1})$) represents a function f applied to \mathbf{x} at two time points \mathbf{x}_t and a proceeding \mathbf{x}_{t+1} . Let ψ represent a path γ as a function at α on a path traversing the interval $[\mathbf{x}_t, \mathbf{x}_{t+1}]$ where $\psi(\alpha) = \langle x^1(\alpha), \dots, x^J(\alpha) \rangle$, we have

$$\int_{\gamma} \nabla F \cdot d\psi = F(\mathbf{x}_{t+1}) - F(\mathbf{x}_t)$$

as

$$\begin{aligned} \int_{\gamma} \nabla F \cdot d\psi &= \int_{\mathbf{x}_t}^{\mathbf{x}_{t+1}} \nabla F(\psi(\alpha)) \cdot \psi'(\alpha) d\alpha \\ &= \int_{\mathbf{x}_t}^{\mathbf{x}_{t+1}} \left(\frac{\partial F}{\partial x^1} \frac{dx^1}{d\alpha} + \dots + \frac{\partial F}{\partial x^J} \frac{dx^J}{d\alpha} \right) d\alpha \\ &= \int_{\mathbf{x}_t}^{\mathbf{x}_{t+1}} \frac{d}{d\alpha} F(\psi(\alpha)) d\alpha \\ &= F(\psi(\mathbf{x}_{t+1})) - F(\psi(\mathbf{x}_t)) \\ &= F(\mathbf{x}_{t+1}) - F(\mathbf{x}_t). \end{aligned}$$

□

Proposition 6. *Any path given by Interval-IG through a cluster centroid from equation 6.3, namely $\hat{\omega}_i$ between two time points, will equal the difference in prediction probability of the two time points \mathbf{x}_t and \mathbf{x}_{t+1} .*

Proof. Considering a path γ and associated path function ψ specifying a path between points \mathbf{x}_t , $\hat{\omega}_i$ and \mathbf{x}_{t+1} , from the additive property for integrals we have:

$$\begin{aligned} \int_{\mathbf{x}_t}^{\hat{\omega}_i} \frac{d}{d\alpha} F(\psi(\alpha)) d\alpha + \int_{\hat{\omega}_i}^{\mathbf{x}_{t+1}} \frac{d}{d\alpha} F(\psi(\alpha)) d\alpha \\ = \int_{\mathbf{x}_t}^{\mathbf{x}_{t+1}} \frac{d}{d\alpha} F(\psi(\alpha)) d\alpha \\ = F(\mathbf{x}_{t+1}) - F(\mathbf{x}_t) \end{aligned}$$

□

It follows that Batch-IG returns the prediction probability that is equal to the difference in prediction probability between the first time point \mathbf{x}_1 and final time point \mathbf{x}_N in a batch χ_b . Direct from proposition 6 it is easy to see that, the integral can be generalised over a time batch χ_b containing T time points, such that:

$$\int_{\mathbf{x}_1}^{\mathbf{x}_T} \frac{d}{d\alpha} F(\psi(\alpha)) d\alpha = F(\mathbf{x}_T) - F(\mathbf{x}_1) \quad (6.4)$$

It follows that the Batch-IG method can return the prediction probability at time points independently. From the definition of Interval-IG given in equation 6.2, since Interval-IG always arrives at \mathbf{x}_{t+1} . Simply distributing the difference $(x_{t+1}^j - x_t^j)$ out and summing over the j (feature) dimensions gives:

$$\underbrace{\sum_{j=1}^J x_{t+1}^j \times \int_{\alpha=0}^1 \frac{\partial F(\mathbf{x}_t + \alpha \times (\mathbf{x}_{t+1} - \mathbf{x}_t))}{\partial x^j} d\alpha}_{F(\mathbf{x}_{t+1})}$$

and

$$\underbrace{\sum_{j=1}^J x_t^j \times \int_{\alpha=0}^1 \frac{\partial F(\mathbf{x}_t + \alpha \times (\mathbf{x}_{t+1} - \mathbf{x}_t))}{\partial x^j} d\alpha}_{F(\mathbf{x}_t)}$$

respectively. This can be carried out for all N time points within a time batch. Thus, the **Representivity** property is satisfied. The information retained by path would in turn satisfy the **Explainability** property in that any instances that differ in both features and prediction probability do not return an empty explanation.

Notably, the integral towards any intermediate point $\hat{\omega}_i$ has a path length less than or equal to the path length between an initial time \mathbf{x}_t and the final point in time \mathbf{x}_{t+1} , as from equation 6.3, it is known that a path length for two points \mathbf{x}_t and \mathbf{x}_{t+1} can be given by:

$$\begin{aligned} \delta(\mathbf{x}_t, \mathbf{x}_{t+1}) &= \|\psi'(\alpha)\| d\alpha \\ &= \int_{\mathbf{x}_t}^{\mathbf{x}_{t+1}} \sqrt{\left(\frac{dx^1}{d\alpha}\right)^2 + \dots + \left(\frac{dx^J}{d\alpha}\right)^2} d\alpha. \end{aligned} \quad (6.5)$$

Under equation 6.3 the following Euclidean distance inequality holds for a intermediate point $\hat{\omega}_i$, where:

$$\hat{\omega}_i : \|\omega_i - \mathbf{x}_t\|_2 \leq \|\mathbf{x}_{t+1} - \mathbf{x}_t\|_2 \wedge \|\omega_i - \mathbf{x}_{t+1}\|_2 \leq \|\mathbf{x}_{t+1} - \mathbf{x}_t\|_2$$

Then, by substituting in any point that satisfies the requirements of $\hat{\omega}_i$ into equation 6.5, we have:

$$\delta(\hat{\omega}_i, \mathbf{x}_{t+1}) \leq \delta(\mathbf{x}_t, \mathbf{x}_{t+1}) \wedge \delta(\hat{\omega}_i, \mathbf{x}_t) \leq \delta(\mathbf{x}_t, \mathbf{x}_{t+1}).$$

6.4.1 Computable Property Satisfiability

TC holds for Batch-IG explanations as follows. To satisfy TC, we must show that this holds for Interval-IG, as this then generalises to Batch-IG, thus we must first approximate the integral with a Riemann sum of Interval-IG $_j$, for simplicity we show this for the straight line path between time points and therefore Interval-IG $_j^{\mathcal{R}}$ is defined:

$$\text{Interval-IG}_j^{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1}) := (x_{t+1}^j - x_t^j) \times \frac{1}{K} \sum_{k=1}^K \frac{\partial F(\mathbf{x}_t + \frac{k}{K} \times (\mathbf{x}_{t+1} - \mathbf{x}_t))}{\partial x^j}. \quad (6.6)$$

Extrapolating from equation 6.6 and simplifying to a straight line between time points we let $\theta_K^j = \frac{1}{K} \sum_{k=1}^K \frac{\partial F(\mathbf{x}_t + \frac{k}{K} \times (\mathbf{x}_{t+1} - \mathbf{x}_t))}{\partial x^j}$, we have the straight line path integral approximation of Batch-IG over the j^{th} feature dimension and by utilising the concept of directional derivatives, as K approaches infinity, we have:

$$\left(F(\mathbf{x}_{t+1}) - F(\mathbf{x}_t) \right) - \left(\sum_{j=1}^J \theta_K^j x_{t+1}^j - \sum_{j=1}^J \theta_K^j x_t^j \right) = 0. \quad (6.7)$$

as shown by proposition 6, it is clear this holds given any intermediary points. To obtain a computable solution, one can obtain a minimal value of K , given a positive real number τ close or equal to 0 as a hyper-parameter with the following steps. Firstly, we define a precision parameterised on K , $\Delta P(\cdot; K)$, as

$$\Delta P(f, \chi_b; K) = \sum_{t=1}^{N-1} \left(\left(|F(\mathbf{x}_{t+1}) - F(\mathbf{x}_t)| \right) - \left(\left| \sum_{j=1}^J \theta_K^j x_{t+1}^j - \sum_{j=1}^J \theta_K^j x_t^j \right| \right) \right). \quad (6.8)$$

Then, we can find a minimum value for K namely \hat{K} , that satisfies

$$\hat{K} = \arg \min_K \Delta P(F, \chi_b; K) : \Delta P(\cdot) \leq \tau.$$

With \hat{K} substituting K in Equation 6.7, we have

$$\sum_{t=1}^{T-1} \left(\left| \sum_{j=1}^J \theta_{\hat{K}}^j x_{t+1}^j - \sum_{j=1}^J \theta_{\hat{K}}^j x_t^j \right| \right) \leq \sum_{t=1}^{N-1} \left(\left(|F(\mathbf{x}_{t+1}) - F(\mathbf{x}_t)| \right) + \tau \right). \quad (6.9)$$

Next, we show how TS holds for Batch-IG for all features that are unchanged along the j^{th} dimension. Dynamic and static features will be inherently defined by Interval-IG $_j^{\mathcal{R}}$ for each j , as the weights along the j^{th} dimension equal 0 where $x_{t+1}^j = x_t^j$, therefore:

$$0 \times \frac{1}{K} \sum_{k=1}^K \frac{\partial F(\mathbf{x}_t + \frac{k}{K} \times (\mathbf{x}_{t+1} - \mathbf{x}_t))}{\partial x^j} \implies \text{Interval-IG}_j^{\mathcal{R}}(F_{x_t^j, x_{t+1}^j}^s) = 0. \quad (6.10)$$

Thus, it follows that any two adjacent time points of Batch-IG satisfies the **time sensitivity** property. Satisfying time sensitivity implies that there is no requirement

for human input, as the method will differentiate dynamic from static features and instead rely on the data within a time batch. Refer to Table 6.1 for each XAI methods adherence to the aforementioned time-oriented properties. Following this, it is clear that for any property that holds for path methods hold constrained to some degree of τ , or are otherwise intrinsically enforced (see equation 6.10).

6.5 Controlled Experiments

In this section, we evaluate the performance of different XAI methods by, (1) measuring the faithfulness corresponding to a generated synthetic dataset, to determine attribution recovery over time points, and, (2) provide a comparison of path based methods by evaluating the ease of fitting the assigned paths to the data.

6.5.1 Faithfulness

We generate a simple synthetic data set where the importance of features are known. Therefore, we define the input data set to be a $\mathbb{R}^{D \times 2}$ matrix of instances, where $D = 50,000$ and a label for an instance at time point t is given by

$$p_t = 2 \sin(x_t^1) + 4 \sin(x_t^2).$$

Considering an instance given at time-point t and $t+1$ respectively, keeping the subscript notation for time points, we have the time batch χ_b containing $\mathbf{x}_t = \langle 4_t, 2_t \rangle$ and $\mathbf{x}_{t+1} = \langle 4_{t+1}, 1_{t+1} \rangle$. Generating the true labels we have $p_t \approx 2.1235$ and $p_{t+1} \approx 1.8522$, the change in value equates to $\Delta p_t \approx -0.2713$, therefore the difference in attribution is given by the difference in the second term of the equation at both time points, namely $\Delta p_t = 4 \sin(1) - 4 \sin(2)$, as the first term is the same.

We compare Batch-IG to DeepSHAP, SHAP, LIME, Gradients \times Input and IG in the controlled setting to determine if the returned attribution recovers the difference in prediction whilst correctly assigning attribution from our example (see Table 6.2). We observe that Batch-IG indeed identifies feature change impact most accurately, exceeding all other methods. Thus, we show that empirically only Batch-IG implementations satisfy **irreducability** over time. We provide a table of property satisfiability in Table 6.1.

6.5.2 Comparison of Path Based Methods

To compare the quality of given paths, we use the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) to analyse the model performance given the different paths that are generated by the path based methods: IG and Batch-IG. This will allow us to see how well the attributed paths can fit to the given data. The use of a Gaussian Mixture Model (GMM) enables the analysis of AIC and BIC, and to see how well a generated path can conform to the cluster densities and be able to fit in distribution with a degree of confidence.

Table 6.2: We demonstrate attribution recovery for an instance, such that we know the ground truth. Therefore, the difference in predictions should be fully recovered by the attribution given to x^2 . We take the difference in attribution between $t + 1$ and t for example $\Phi(x^2) = \Phi(x_{t+1}^2) - \Phi(x_t^2)$.

XAI Method (Model)	$f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t)$	$\Phi(x^1)$	$\Phi(x^2)$
DeepSHAP (ANN)	-0.2737	0.0008	-0.2744
Batch-IG (ANN)	-0.2737	0	-0.2737
LIME (XGBoost)	-0.2705	0.0835	0.1352
SHAP (XGBoost)	-0.2705	0.0002	-0.2708
Gradient \times Input (ANN)	-0.2737	-0.0002	1.8969
IG (ANN)	-0.2737	0.0049	-0.2783

Table 6.3: Table containing the BIC and AIC scores for different path based methods. From this we observe that the lowest AIC and BIC in the given synthetic datasets is given by Batch-IG, indicating superior performance in the given cases (*note: the step sizes for each model equate to the same with respect to the Riemann approximations across the entire path.*)

	BIC	AIC
Generated set 1	IG: 9054.83	IG: 8964.50
	Batch-IG: 8615.70	Batch-IG: 8525.37
Generated set 2	IG: 8786.99	IG: 8696.67
	Batch-IG: 8663	Batch-IG: 8572.67
Generated set 3	IG: 8867.69	IG: 8777.37
	Batch-IG: 8745.42	Batch-IG: 8655.09

To construct this experiment, we generate 3 clusters with known labels $\{0, 1, 2\}$. Each generated dataset contains 2000 instances with 2 independent feature dimensions, with a cluster standard deviation of 0.6. Whereby, there exists a cluster between the starting points \mathbf{x}_t and end point \mathbf{x}_{t+1} . Thus, we can utilise the optimised variant in the unique case.

Upon generating the clusters, we fit a neural network to the data, posterior to this we can then generate our path integrals between points. To test this, we select a target point \mathbf{x}_{t+1} (i.e. the point to explain) to be the centroid of one cluster. We then use our prior time point \mathbf{x}_t (for Batch-IG) to be the furthest cluster centroid, and the all zeros baseline is used for IG. The GMM fit enables us to extract AIC and BIC values (the *lower* the better) seen in Table 6.3.

Table 6.4: The student example corresponding to the generated explanations given in Figure 6.2.

	Week	Diff.	Sol.	Dec.	Needs	Active	Insp.	Alert	Surp.	Cur.	Conf.	Anx.	Joy.
Week 1	1	3	6	7	7	4	4	4	3	4	3	1	3
Week 2	2	6	5	5	5	4	5	5	5	5	5	4	4
Week 4	4	1	7	7	7	5	5	5	4	5	3	2	4

6.6 Applications

To demonstrate applications of Batch-IG, we introduce two datasets, namely: Education Data used in [ZLF⁺23] and COVID data used in [KED⁺21]. We observe that the education research does not include student based analysis with respect to week traversal, and similarly, the latter is restricted to explaining a single time point with current XAI methods. Thereby, we omit the limitations by producing temporal explanations for further analysis for given samples. We then provide an evaluation of the Batch-IG generated paths against IG.

6.6.1 Education Data

The education data contains 8 weeks of anonymized student records over a single module. From this, we consider the multiclass classification problem:

Given a set of 17 independent variables, what factors contribute to a change in students determination over given weeks?

Whereby, the dependent variable follows a 5-point Likert scale (0 = not at all, 5 = extremely). Explainers such as SHAP would give us an independent explanation corresponding to each week, as opposed to traversing explanations corresponding to weekly time points. Similarly, for details corresponding to the Likert scale with respect to independent variables, see [ZLF⁺23].

The dataset is split into a training set containing all weeks ≤ 4 (179 instances), and the respective testing dataset contains weeks > 4 (99 instances), where a simple neural network has been fit over the multi-class classification problem.

Upon this, we can consider the following example: *“Given a student that has attended the first 3 weeks, and naturally became more determined (determined with a Likert score of 4 on week 1 to 5 on week 2 and 4), which factors lead to this student being more determined?”*.

For demonstration, we consider a student over 4 weeks (see Table 6.4). We generate the following explanations given in Figure 6.2. One example we can infer from the given explanation is that the increase in anxiety between weeks 1 and 2, lead to a decrease in determination, whereas, from weeks 2 to 4, there’s a decrease in anxiety leading to an increase in determination and increase in activity between weeks 2 to 4 increased how determined the student was. Similarly, an increase in inspiration in weeks 1 to 2 had the most impact during those weeks in the student being more determined.



Figure 6.2: Attribution over time transitions, between weeks 1, 2 and 4 for a student for predicting the how determined they are, where the accumulated average gradients per time-interval are associated with Likert value 5 for determined (extremely determined.).

6.6.2 COVID Data

The COVID dataset used in [KED⁺21] is collated via the Public Health England website¹, we omit details regarding the curation of the dataset as this can be found in [KED⁺21]. From this we consider the binary classification problem:

Given a set of 18 independent control measures and the temperature and humidity on a day, given any region can we determine which factors contributed to an increased rate of infection?

Here each temporal data point corresponds to a day τ , at time t . The independent variables for example “*School Closure = 0*” indicates that there has been no control measures implemented as of yet, respectively “*School Closure = 1*” indicates the first 0-5 days that the subsequent control measure has been placed by each increment of

¹<https://www.gov.uk/government/organisations/public-health-england>

1 corresponds to a 5 day period. The binary classification problem is regarding an increase in rate of infection $R_t \geq 1 = 1$ or a decreased rate of infection $R_t < 1 = 0$.

Consider a time-batch corresponding to a single region “East Midlands”. For a simple example we form an analysis of two days in the East Midlands, whereby the rate of infection (R_t) for both days is class 1 ($R_t \geq 1$), but the prediction probability towards class 1 increased over the day transition (see Table 6.5). We use our introduced method to identify which factors had positive or negative influence between control measures and the weather, over the day transition.

We provide illustrative explanations as an example in Figure 6.3. In this transition between days, we can see many new control measures transitioning to 1, the first day of the first week of implementation. What we can observe is that the implementation of cafe and restaurants control measure has the largest influence towards the class 0 ($R_t < 1$), meaning that this feature had the most influence in decreasing infection rate, inline with findings seen in [KED⁺21].

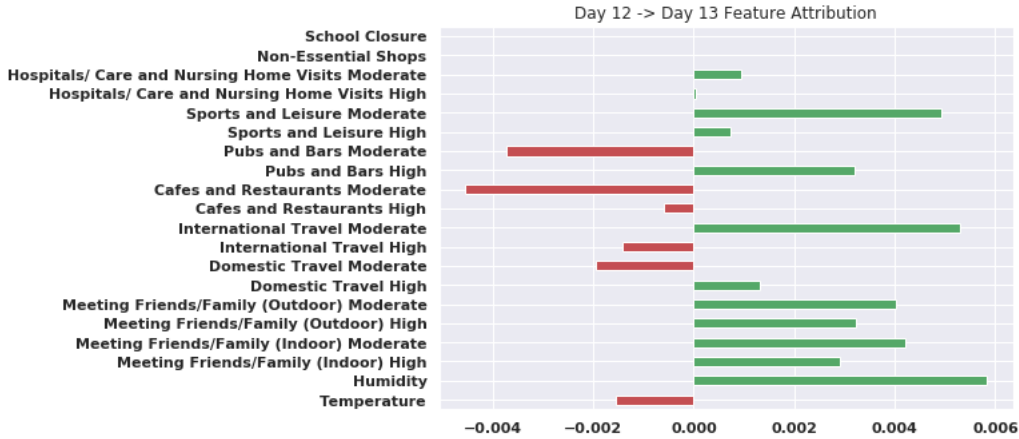


Figure 6.3: Attribution over time transitions, between the days 12 and 13, whereby most features had transitioned into the next period of time that the control measures had been in place.

6.6.3 Performance

We evaluate the root mean squared error (RMSE) of the difference in prediction probabilities assigned to a generated path and the probabilities in predicting the generated path with a separate independently trained network. More formally, we let $F^0(\psi(\alpha))$ generate prediction probabilities at a point α towards a target class, and let $F^l(\psi(\alpha))$ be a separate $l(0 \leq l \leq L)$ network that generates prediction probabilities. Therefore, we have a collection of L networks, $\langle F^1, \dots, F^L \rangle$, here $L = 5$, that generate probabilities for each time point along a given path ψ where f^0 generates the target probabilities over $0 \leq \alpha \leq T$ time points. Thereby, our

Table 6.5: The public health COVID data corresponding to the explanations given Figure 6.3.

	Temperature	Humidity	Meeting Friends/Family (Indoor) High
Day 12	4.38	73.29	0
Day 13	5.4	67.12	1
	Meeting Friends/Family (Indoor) Mod	Meeting Friends/Family (Outdoor) High	Meeting Friends/Family (Outdoor) Mod
Day 12	0	0	0
Day 13	1	1	1
	Domestic Travel High	Domestic Travel Mod	International Travel High
Day 12	0	0	0
Day 13	1	1	1
	International Travel Mod	Cafes and Restaurants High	Cafes and Restaurants Mod
Day 12	0	0	0
Day 13	1	1	1
	Pubs and Bars High	Pubs and Bars Mod	Sports and Leisure High
Day 12	0	0	0
Day 13	1	1	1
	Sports and Leisure Mod	Hospitals/ Care and Nursing Home Visits High	Hospitals/ Care and Nursing Home Visits Mod
Day 12	0	0	0
Day 13	1	1	1

Table 6.6: RMSE comparing Batch-IG against the IG path over 5 functionally equivalent Neural Networks for predicting the path interpolation values on the Education dataset.

RMSE	$l=1$	$l=2$	$l=3$	$l=4$	$l=5$
Batch-IG	1.0678	0.2588	0.1524	0.2141	0.0742
IG	1.5464	1.0257	0.4863	0.6990	0.5747

Table 6.7: RMSE comparing Batch-IG against the IG path over 5 functionally equivalent Neural Networks for predicting the path interpolation values on the COVID dataset.

RMSE	$l=1$	$l=2$	$l=3$	$l=4$	$l=5$
Batch-IG	0.0827	0.4328	0.3072	0.0731	0.3202
IG	0.2431	0.6192	0.5680	0.2027	0.4138

RMSE metric is defined as:

$$\text{RMSE}(F^0, F^l|\psi) = \sqrt{\frac{\sum_{\alpha=1}^n F^0(\psi(\alpha)) - F^l(\psi(\alpha))^2}{T}}$$

We can observe that Batch-IG had a lower RMSE over all network evaluations across both datasets for the generated paths.

6.7 Conclusion

In this chapter, we formally introduced and evaluated Batch-IG, as an extension to minimise the OoD problem seen with linear interpolations with a temporally influenced baseline instance when compared to the IG method. We enabled the ability for temporal explanations and provided an analysis against state-of-the-art

methods on both controlled and real data. We extrapolated an analysis to property satisfiability through theoretical and empirical analysis. We finally provided real world case study explanations produced by our method. Our comparison shows that the paths produced by our method out-performed IG in the given cases.

Chapter 7

The Minimisation and Quantification of Path-Based Uncertainty for Generative Counterfactual Explanations

Contents

7.1	Introduction	103
7.2	Axioms for Path-Based Explainers	105
7.3	Proposed Model: QUCE	107
7.4	Experimental Setup	115
7.5	Quantitative Evaluation	116
7.6	Conclusion	118

7.1 Introduction

Given the prevalence of big data and increased computability, the application of Deep Neural Network (DNN) methods are a commonality. However, the intricacies and depth of DNN architectures lead to results that lack inherent interpretability. In pivotal domains such as healthcare and finance, interpretability is crucial and thus the application of eXplainable Artificial Intelligence (XAI) to extract valuable insights from the DNN models is widespread [BMP23, CMR⁺23].

The Path-Integrated Gradients (Path-IG) [STY17] formulation presents axiomatic properties that are upheld solely by path-based explanation methods. The Out-of-Distribution (OoD) problem is prevalent in the application of path-based explanation methods [DSZ⁺23]; here the intuition is that traveling along a straight line path can incur irregular gradients and thus provide noisy attribution values

[KVA⁺21]. Another known limitation of many Integrated Gradient (IG) [STY17] based approaches is the selection of a baseline reference; thus the Adversarial Gradient Integration (AGI) [PLZ21] method relaxes this constraint by generating baselines in adversarial classes. We note that AGI utilizes the path-based approach for generating counterfactual examples, and for this reason will be a primary baseline for our proposed method throughout this paper.

Counterfactual explanations [Gui22] are often presented in the form of counterfactual examples [WMR18, KMMTS21]; here the goal is to provide a counterfactual example belonging to an alternative class with respect to a reference example. Counterfactual approaches aim to answer the question:

“Given an instance, what changes can be made to change the outcome for that instance?”

Naturally, this allows for empirical observation as to which changes could provide an alternative outcome. The argument for using counterfactual methods is often developed from a causal lens [Höf05, PGS⁺20]. It follows that to better evaluate this causal relationship, a promising avenue is to unify feature attribution with counterfactual examples, as demonstrated by the Diverse Counterfactual Explanations (DiCE) [KMMTS21] method. Naturally, given quantitative approaches to feature attribution calculation such as these, ideally feature attribution methods should adhere to desirable axioms across XAI literature [STY17, ABN22]. Thus, we aim to utilize state-of-the-art feature attribution assignment as to satisfy key axioms in our model development. Another concern with production of counterfactual examples is the production of realistic paths to successfully create a counterfactual example; therefore we shall be exploring uncertainty.

Uncertainty quantification is not often considered when producing explanations, although some approaches have explored this. Examples include [SHSL21] where post-hoc model-agnostic approaches such as Local Interpretable Model-Agnostic Explanations (LIME) [RSG16] and kernel SHapley Additive exPlanations (SHAP) [LL17] are adapted into a Bayesian framework to model the uncertainty of the explanations produced. Since path-based methods are implementation invariant with respect to the model, the explanations will be consistent and thus there will be no variance in the explanations produced. In this way uncertainty quantification in the form of repeated runs of the XAI algorithm as elucidated in [MCR⁺23], while applicable to post-hoc approximation XAI methods, will not suffice for implementation-invariant models. Autoencoder-based frameworks have also been used to measure uncertainty for both machine learning predictions and explanations [ABA⁺21], with integration of uncertainty seen in the production of counterfactual examples [MGM⁺22, GFBG21]. Inherently, autoencoder approaches provide a more suitable basis for attribution settings by evaluating uncertainty in explanations with respect to the uncertainty inherent in the data.

The standard autoencoder approach evaluates the reconstruction error, which is often utilized in work surrounding anomaly detection [TMG23, AFF23]; instead, we explore the use of a variational autoencoder (VAE) for variational inference, and thus investigate counterfactuals generated with respect to our approximation of the true data distribution.

To address the above constraints, we propose the Quantified Uncertainty (Path-Based) Counterfactual Explanations (QUCE) method. The focus of the proposed method is three-fold. We aim to

- minimize uncertainty and thus maximize the extent to which the generated paths and counterfactual examples are within distribution;
- relax the straight-line path constraints of Integrated Gradients;
- provide uncertainty quantification for counterfactual paths and counterfactual feature attribution.

In this chapter, we focus on the minimization of uncertain paths for counterfactual generation with quantifiable uncertainty measures on the generated counterfactual. QUCE’s learning process relaxes IG’s straight-line path restrictions as part of the generative process. This instead allows us to answer the question:

“Given an instance, what is a realistic path we can obtain to change the outcome of that instance, and how certain is it?”

Intuitively, it is unclear in many scenarios if one single best path toward an alternative outcome exists; for example a patient’s treatment path may be unclear [BSKM16], or there may be many viable paths to achieve the same outcome [BSSG20]. Therefore, QUCE utilises both a single and multiple-paths approach, so we can observe a generalized explanation over all paths for an instance in obtaining a desired class and likewise inspect many example paths. From the multiple-paths approach, we are able to gauge a general approximation over many piecewise linear paths of the most important features in obtaining a desired counterfactual outcome, each path independently aiming to minimize the uncertainty in its generative process and thus provide greater in-distribution interpolation. Similarly, we present the optimisation over the key metrics – *proximity* [Gui22], *validity* [Gui22] and *uncertainty* [Sag22]. We provide an overview of some generative counterfactual methods and evaluate which account for the aforementioned metrics in Table 7.2. We provide a simple illustrated example in Figure 7.1.

7.2 Axioms for Path-Based Explainers

Here we informally introduce axioms used in XAI literature. In the seminal work of [STY17], the authors introduce a set of axioms which play a foundational role

7. The Minimisation and Quantification of Path-Based Uncertainty for Generative Counterfactual Explanations

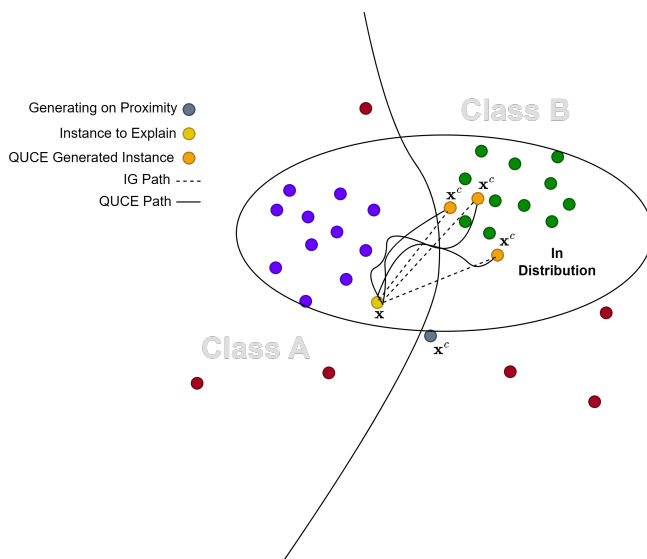


Figure 7.1: A simple illustration demonstrating QUCE generated examples and associated paths. It illustrates paths and counterfactual examples weighted towards being in distribution. The dotted line is the straight line IG path towards the QUCE generated examples. The bold line is the QUCE generated path. The grey point is an example of an instance generated on proximity and opposing class alone.

Table 7.1: An overview of generative counterfactual methods and their consideration of key metrics. Here we observe of the three metrics QUCE is the method that accounts for all three.

Properties	Proximity	Validity	Uncertainty
QUCE	✓	✓	✓
DiCE	✓	✓	✗
AGI	✓	✓	✗

for the development of path-based feature attribution methods. Informally, these encompass:

- **Completeness:** The difference in prediction between the baseline and input should be equal to the sum of feature attribution values.
- **Sensitivity(a):** For every input and baseline that differ in one feature and for which the subsequent prediction is different, feature attribution should only be given to that one feature.
- **Sensitivity(b):** If the neural network is not mathematically dependent on one feature, the feature attribution assigned to that feature should be 0.

- **Implementation Invariance:** Two functionally equivalent neural networks should produce the same feature attribution as an explanation.

The above axiomatic guarantees are for path-based explainers and for this reason we utilise the path-based formulation in this chapter.

7.3 Proposed Model: QUCE

7.3.1 Generating Counterfactuals with QUCE

To generate counterfactuals we propose a three-part objective function with a composite weighting vector $\Lambda = \langle \lambda_1, \lambda_2, \lambda_3 \rangle$, where each $\lambda \in \Lambda$ is an independent tolerance (weight) used to determine the influence of each part of the joint objective function presented in equation 7.1. By minimizing the objective function, we obtain the generated counterfactual \mathbf{x}_c . Informally, our objective function is composed of three parts:

- \mathcal{L}_{pr} , for the maximization of the probability towards the desired class;
- \mathcal{L}_δ , to minimize the distance between the instance and a generated counterfactual;
- \mathcal{L}_ϵ , to minimize the uncertainty of both the generated paths and generated counterfactual examples.

Combining these terms with our weighting vector, we have

$$\mathcal{G}(\mathbf{x}) = \arg \min_{\mathbf{x}_c} \lambda_1 \mathcal{L}_{pr} + \lambda_2 \mathcal{L}_\delta + \lambda_3 \mathcal{L}_\epsilon. \quad (7.1)$$

Having constructed our generative objective function, we provide further notation to illustrate the learning process. First, we consider an iterative learning process such as gradient descent on \mathbf{x} to produce a path from \mathbf{x} to a generated \mathbf{x}_c . Thus, we aim to minimize the developed function $\mathcal{G}(\mathbf{x})$ through the gradient descent approach (variants include, e.g., SGD and ADAM). We initially let $\mathbf{x}_c = \mathbf{x}$; \mathbf{x}_c is updated via

$$\begin{aligned} \mathbf{x}_c &\leftarrow \mathbf{x}_{\Delta_i}, \\ \Delta_i &= \varphi \nabla_{\mathbf{x}_c} (\mathcal{G}(\mathbf{x})), \\ \mathbf{x}_{\Delta_i} &= \mathbf{x} - \Delta_i. \end{aligned}$$

Let \mathbf{x} be updated on a loop over $i (0 \leq i \leq n)$ iterations; when $i = n$ we let have our \mathbf{x}_c indicating our generated point. Here φ represents the ‘‘learning rate,’’ a small positive multiplier value $\varphi \in [0, 1] : \varphi \ll 1$. We store each update on \mathbf{x}_c as a vector $\mathbf{x}^\Delta = \langle \mathbf{x}_{\Delta_0}, \dots, \mathbf{x}_{\Delta_n} \rangle$.

7.3.2 Finding Counterfactuals

7.3.2.1 Valid Counterfactuals

A key concept in finding counterfactual examples is ensuring that the counterfactual is indeed *valid*, and thus we aim to produce counterfactual examples that belong to a counterfactual class. Thus, given a target counterfactual class $\mathcal{T} \in \{0, 1\}$, probabilistic function $F : \mathbb{R}^J \rightarrow [0, 1]$ and probabilistic decision threshold $\tau \in [0, 1]$ we aim to find an instance \mathbf{x}_c that satisfies

$$F(\mathcal{T}|\mathbf{x}_c) \geq \tau. \quad (7.2)$$

Here τ is a probability threshold for the target class \mathcal{T} . Thus, we need a generator \mathcal{G} that satisfies the condition in equation 7.2. To achieve this, we can maximize the likelihood of an instance belonging to a class \mathcal{T} . The maximum log likelihood criterion is defined as:

$$\mathcal{L}_{pr} = \left[\log[F(\mathcal{T}|\mathbf{x}_c)] \right]$$

such that, we only accept counterfactuals where $F(\cdot|\cdot) \geq \tau$. For optimisation we can rewrite this as a minimization problem, instead minimizing the negative log-likelihood:

$$\mathcal{L}_{pr} = \left[-\log[F(\mathcal{T}|\mathbf{x}_c)] \right].$$

This constitutes the constrained optimisation problem with respect to some target class \mathcal{T} .

7.3.2.2 Proximity for Counterfactuals

Given an instance, in the production of counterfactual examples we often aim to find a counterfactual example that is “similar” in feature space to the instance. This is often termed *proximity*.

In this work, we use the l_2 norm as the proposed model focuses on producing counterfactuals from continuous features, defining proximity as follows:

Definition 7.1 (Proximity) Given an instance \mathbf{x} and its counterfactual example \mathbf{x}_c , the proximity between the two instances is given by

$$\mathcal{L}_\delta = \left[\frac{1}{2} \|\mathbf{x}_c - \mathbf{x}\|^2 \right]. \quad (7.3)$$

7.3.2.3 Minimally Uncertain Counterfactuals

To maximize the certainty of counterfactual examples, we examine their complement—namely, the uncertainty associated with a counterfactual example. To explore this, we establish the concept of *counterfactual uncertainty*. Informally, we consider uncertainty to be the Evidence Lower Bound (ELBO) as measured by a Variational Autoencoder (VAE) framework. The objective function ELBO is comprised of two components, namely the Kullbeck–Leibler (KL) divergence and a reconstruction loss. This is defined as the following:

$$\text{VAELoss}(\mathbf{x}) = \mathbb{E}_{q_\theta}[\log q_\theta(\mathbf{z}|\mathbf{x}) - \log p_\psi(\mathbf{z})] - \mathbb{E}_{q_\theta} \log p_\psi(\mathbf{x}|\mathbf{z})$$

where p and q are probability distributions and \mathbf{z} is the latent representation of \mathbf{x} . The aim is to find a θ that successfully models the true training data distribution ψ , and thereby satisfying the following minimization problem:

$$\{\theta^*, \psi^*\} = \arg \min_{\theta, \psi} \text{VAELoss}(\mathbf{x}) \quad (7.4)$$

Intuitively, ELBO aims to produce a p and q that are as close as possible while simultaneously optimizing the mapping of \mathbf{z} back to its original representation \mathbf{x} , through learning to model the distributions with the parameters θ and ψ . Thus it suffices to say that if the reconstructing loss is minimized, the decoded \mathbf{z} should successfully map back to \mathbf{x} . Posterior to the pre-trained VAE, we now have a fixed representation shaping our p and q distributions with our parameters θ^* and ψ^* and thus we can provide counterfactual uncertainty as:

Definition 7.2 (Counterfactual Uncertainty) Following equation 7.4, we simply take the loss without minimization to be the counterfactual uncertainty, thus given our fixed parameters θ^* and ψ^* , we have

$$\mathcal{L}_\epsilon = \mathbb{E}_{q_{\theta^*}}[\log q_{\theta^*}(\mathbf{z}|\mathbf{x}_c) - \log p_{\psi^*}(\mathbf{z})] - \mathbb{E}_{q_{\theta^*}} \log p_{\psi^*}(\mathbf{x}_c|\mathbf{z}).$$

7.3.3 Uncertainty in Counterfactual Explanations

Definition 7.2 allows for the evaluation of new generated instances with a measure of how “good” the fit of the new instance is with respect to the training data distribution, and similarly how well a path fits into the data distribution.

From definition 7.2 we have a quantifiable measure of uncertainty for the generated counterfactual \mathbf{x}_c . Expressing this in vector form as a difference, which is needed for modifying and updating our generated \mathbf{x}_c to adjust for underlying uncertainty, we define *Feature-wise Counterfactual Uncertainty* by simply looking at the reconstruction error, defined for simplicity as follows:

Definition 7.3 (Feature-wise Counterfactual Uncertainty) Given counterfactual uncertainty, we can rewrite a reconstruction error equivalent at a feature level by calculating a vector of differences $\epsilon_{\mathbf{d}}$ such that

$$\epsilon_{\mathbf{d}} = \langle |d^1|, \dots, |d^J| \rangle : \langle d^1, \dots, d^J \rangle = \mathbf{d}; \quad (7.5)$$

$$\text{where } \mathbf{d} = \mathbf{x}_c - \hat{\mathbf{x}}_c. \quad (7.6)$$

With this representation, we can then successfully update \mathbf{x}_c by both adding and subtracting this vector of feature-wise counterfactual uncertainty as given by the reconstruction error, and thus we can calculate *Counterfactual Explanation Uncertainty*.

Definition 7.4 (Counterfactual Explanation Uncertainty) Given a CFA Φ_{CF} and the feature-wise counterfactual uncertainty $\epsilon_{\mathbf{d}}$, the counterfactual explanation uncertainty is given by

$$\Phi_{CF}^{\epsilon_{\mathbf{d}}} = \Phi_{CF}(\mathbf{x}_c \pm \epsilon_{\mathbf{d}} | \mathbf{x}). \quad (7.7)$$

7.3.4 Path Explanations

The Path-Integrated Gradients [STY17, PLZ21, KVA⁺21, YWB23] formulation is the only approach to our knowledge within the landscape of feature attribution methods that satisfies all the feature attribution axioms in Section 7.2. Therefore we adopt the path integral formulation and relax the straight-line constraint seen in IG. To achieve this, recall the set of learned updates on \mathbf{x} , namely \mathbf{x}^{Δ} . It follows that we can produce explanations over \mathbf{x}^{Δ} with respect to a neural network F .

Formally, let the function F be a continuously differentiable function, the QUCE explanation takes the path integral formulation such that given a smooth function $\psi = \langle \psi^1, \dots, \psi^J \rangle : [0, 1] \rightarrow \mathbb{R}^J$ defining a path in \mathbb{R}^J , where $\psi(\alpha)$ is a point along a path at $\alpha \in [0, 1]$ with $\psi(0) = \mathbf{x}_{\Delta_0}$ and $\psi(1) = \mathbf{x}_{\Delta_n}$, the single-path QUCE explainer is defined as

$$\Phi_{\text{QUCE}}(\mathbf{x}^{\Delta}) := (\mathbf{x}_{\Delta_n} - \mathbf{x}_{\Delta_0}) \times \left(\int_{\mathbf{x}_{\Delta_0}}^{\mathbf{x}_{\Delta_n}} \nabla F(\psi(\alpha)) \cdot \psi'(\alpha) d\alpha \right). \quad (7.8)$$

It follows that explanation uncertainty with respect to a single generated counterfactual \mathbf{x}_c is given as

$$\Phi_{\text{QUCE}}^{\pm \epsilon_{\mathbf{d}}}(\mathbf{x}_c) := ((\mathbf{x}_c \pm \epsilon_{\mathbf{d}}) - \mathbf{x}_c) \times \left(\int_{\mathbf{x}_c}^{\mathbf{x}_c \pm \epsilon_{\mathbf{d}}} \nabla F(\psi(\alpha)) \cdot \psi'(\alpha) d\alpha \right). \quad (7.9)$$

We present the QUCE framework in algorithm 1. We now show through the proof of proposition 7 that the QUCE explanations are easily computed.

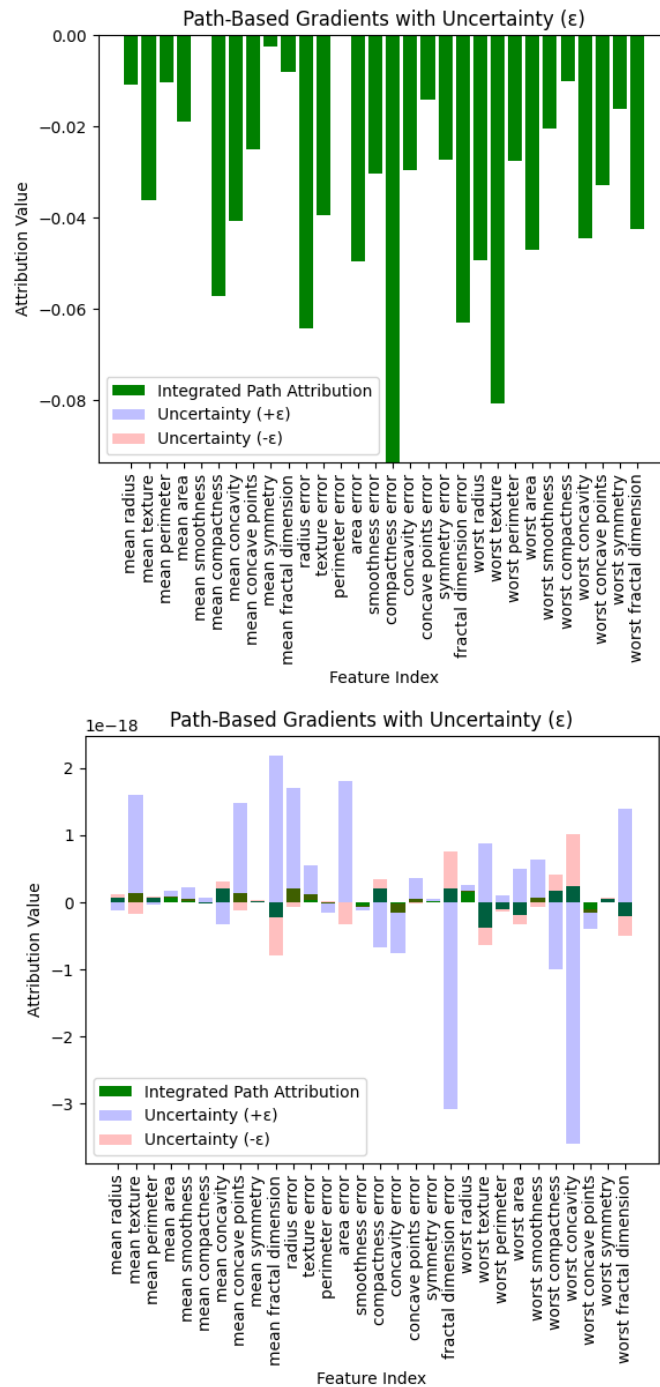


Figure 7.2: Here we illustrate two explanations produced on the Wisconsin Breast Cancer Dataset given by the proposed QUCE method. We observe how each feature influenced the change in the prediction in attempting to generate a counterfactual example. We see the left explanation has almost no uncertainty in generated explanation, whereas the right image demonstrates a large degree of uncertainty in the generated counterfactual explanation.

Proposition 7. *The QUCE explainer has a computable Riemann approximation solution for each feature.*

Proof. Proof provided in the supplementary material. □

Both attributed values from equation 7.9 illustrate uncertainty in feature attribution values given by the QUCE explainer. In proposition 8 and its associated proof we show the implications of weighting uncertainty.

Definition 7.5 (λ -tolerance) An increase in λ -tolerance refers to the reduction of any component λ of Λ . Likewise, a decrease in λ -tolerance refers to the increase of any component λ of Λ .

Naturally, as the weight for uncertainty decreases, we lean towards an increased tolerance of the effects of uncertainty in our explanations. The reasoning behind this is that we may sometimes accept a higher degree of uncertainty, depending on the purpose of generating counterfactuals and the volatility of the task.

Proposition 8. *Increasing the λ -tolerance of uncertainty provides a more flexible search space for possible paths to a generative counterfactual example.*

Proof. This is trivial subject to λ_3 approaching zero. Proof provided in the supplementary material. □

Due to the stochastic nature of our model under nondeterministic variants of gradient descent (e.g. optimization with stochastic gradient descent [SG71]), and with potentially multiple minima (e.g. we may have two points equally “close” to the decision bound with different values), we consider a set of generated counterfactual examples to be given as $C = \langle \mathbf{x}_{c,1}, \dots, \mathbf{x}_{c,b} \rangle$, where b is the number of generated counterfactual examples over some set \mathbf{x} . Given C , we can accumulate attribution over many counterfactuals by avoiding the specification of \mathbf{x}_c ; our lower limit is implicitly assumed to be our instance to explain \mathbf{x} , so that we have

$$\Phi_{\text{exQUCE}}(\mathbf{x}) := \int_{\mathbf{x}_c} \left(\Phi_{\text{QUCE}}(\mathbf{x}^\Delta) \right) p_C(\mathbf{x}_c) d\mathbf{x}_c \quad (7.10)$$

where we integrate over p_C the distribution of C for all $\mathbf{x}_c \in C$ and we can instead rewrite the integral as an expectation as follows:

$$\Phi_{\text{exQUCE}}(\mathbf{x}) := \mathbb{E}_{\mathbf{x}_c \sim C, \alpha \sim \mathcal{U}(0,1)} \left[\Phi_{\text{QUCE}}(\mathbf{x}^\Delta) \right]. \quad (7.11)$$

Here we let $\alpha \sim \mathcal{U}(0, 1)$ indicate interpolation over α for m counterfactual steps in the generator function. Informally, we get the expectation of the gradients over the piecewise linear path between counterfactual steps of the generator. We take

a similar approach to the Expected Gradients [EJS⁺21] formulation, except that instead of sampling from a background set of baselines, we sample from a set of generative counterfactual examples. We make two arguments as to why we use this approach:

- In explaining a counterfactual outcome, we do not know the specific path taken and thus we can average over many paths.
- We can invert the path to explain \mathbf{x} and therefore we can have many generative baselines. This relaxes the specified baseline of many existing path-based explanation methods.

To further extend on the axiomatic guarantees of path-based explainers, we show via proposition 9 that completeness holds when working with the many-paths approach for expected values.

Proposition 9. *Given the function $\Phi_{exQUCE}(\mathbf{x})$, the expected difference in prediction probabilities between generated counterfactuals in the set C with respect to the prediction probability given by $F(\mathbf{x})$, the following equality holds:*

$$\mathbb{E}_{\mathbf{x}_c \sim C, \alpha \sim \mathcal{U}(0,1)} \left[\Phi_{QUCE}(\mathbf{x}^\Delta) \right] \quad (7.12)$$

$$= \mathbb{E}_{\mathbf{x}_c \sim C} \left[F(\mathbf{x}_c) - F(\mathbf{x}) \right] \quad (7.13)$$

Proof. This is a direct consequence of the *completeness* axiom; the proof is provided in the supplementary material. \square

It follows that the expected value approach is equally computable and is a direct extension of proposition 8. We illustrate this in corollary 7.6’s simple proof.

Corollary 7.6 The expected QUCE variant has a computable Riemann approximation solution for each feature.

Proof. This follows from proposition 7; the proof is provided in the supplementary material. \square

Going further, we demonstrate monotonic relationships for generated counterfactual instances that are given by the multiple paths approach. This is a further consequence of the completeness axiom and is expressed in corollary 7.7.

Corollary 7.7 Given two sets of counterfactual examples C^1 and C^2 for an instance \mathbf{x} ,

$$\begin{aligned} \text{if } \mathbb{E}_{\mathbf{x}_c \sim C^1} \left[F(\mathbf{x}_c) - F(\mathbf{x}) \right] &\leq \mathbb{E}_{\mathbf{x}_c \sim C^2} \left[F(\mathbf{x}_c) - F(\mathbf{x}) \right], \\ \text{then } \mathbb{E}_{\mathbf{x}_c \sim C^1} \left[\Phi_{\text{exQUCE}}(\mathbf{x}) \right] &\leq \mathbb{E}_{\mathbf{x}_c \sim C^2} \left[\Phi_{\text{exQUCE}}(\mathbf{x}) \right]. \end{aligned}$$

Proof. This is a direct consequence of proposition 9 and the *completeness* axiom. \square

Given we can compute many-paths explanations, it follows that we can also take the expected gradients for the explanation uncertainty computed by QUCE along each path, such that

$$\Phi_{\text{exQUCE}}^{\pm \epsilon_{\mathbf{d}}}(\mathbf{x}_c) := \mathbb{E}_{\mathbf{x}_c \sim C, \alpha \sim \mathcal{U}(0,1)} \left[\Phi_{\text{QUCE}}^{\pm \epsilon_{\mathbf{d}}}(\mathbf{x}_c) \right]. \quad (7.14)$$

Algorithm 1 Quantified Uncertainty Counterfactual Explanations (QUCE)

X is a dataset, F is a deep network, $\text{VAELoss}(\cdot)$ is a pretrained VAE on X , $\mathcal{G}(\cdot)$ is the joint objective function for QUCE, \mathcal{T} is the target class, τ is a probability threshold towards target class, K is the number of Riemann steps, n is the number of gradient descent updates, φ is a learning rate.

$i=1$

$\mathbf{x}^{\Delta} = []$

procedure QUCE(\mathbf{x})

 pass the instance \mathbf{x} into the function \mathcal{G}

 initialise an instance $\mathbf{x}_c = \mathbf{x}$

while $i \leq n$ **do**

 update \mathbf{x}_c with $\mathbf{x}_c - \varphi \nabla_{\mathbf{x}_c}(\mathcal{G}(\mathbf{x}))$

 append updated \mathbf{x}_c to \mathbf{x}^{Δ}

 increment i

end while

if $F(\mathcal{T}|\mathbf{x}_c) \geq \tau$ **then**

 pass the output \mathbf{x}_c into VAELoss and return $\epsilon_{\mathbf{d}}$

 take the K -step Riemann integral approximation over \mathbf{x}^{Δ}

 take the K -step Riemann integral approximation between \mathbf{x}_c and $\mathbf{x}_c \pm \epsilon_{\mathbf{d}}$

return \mathbf{x}_c , explanation vector, two uncertainty explanation vectors for $\mathbf{x}_c \pm \epsilon_{\mathbf{d}}$

end if

end procedure

7.4 Experimental Setup

7.4.1 Datasets

7.4.1.1 The Simulacrum

The Simulacrum¹ is a synthetic dataset used in this study, the Simulacrum is a large dataset developed by Health Data Insight CiC and derived from anonymous cancer data provided by the National Disease Registration Service, NHS England. We produce five subsets of patient records based on ICD-10 codes corresponding to lung cancer, breast cancer, skin cancer, lymphoma and rectal cancer. These datasets are organised as survival time classification problems, where patients are predicted a survival time of either at least 6 months or less than 6 months.

7.4.1.2 COVID Rate of Infection

The COVID rate of infection dataset contains details on control measures, temperature, humidity and the daily rate of infection for different regions of the UK. Details on data collection are provided in [KED⁺21]. This dataset is a binary classification task identifying an increased rate of infection against a non-increased rate of infection.

7.4.1.3 Wisconsin Breast Cancer

The Wisconsin Breast Cancer (W-BC) [WS95] dataset, provided in the scikit-learn library², is a binary classification dataset that classifies malignant and benign tumours given a set of independent features from breast mass measurements.

7.4.2 Baseline Methods

For comparison, we consider a selection of methods that aim to generate counterfactual examples and also a collection of path-based explainers.

7.4.2.1 Diverse Counterfactual Explanations

DiCE, a counterfactual generator, provides feature attribution values for an instance with respect to its counterfactual examples. We use the DiCE method as a comparison for generating counterfactual examples, as DiCE is not a path-based explainer, we can only compare the generated counterfactuals.

¹<https://simulacrum.healthdatainsight.org.uk/>

²https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html

7.4.2.2 Integrated Gradients

The IG feature attribution method produces explanations for instances in a given dataset. To achieve this, the approach is to integrate over the gradients of a straight-line path from an all-zero baseline vector to the instance to be explained. We modify this in our experiments so that our baseline becomes the instance to be explained, while the target instance is the counterfactual generated by QUCE, so we can evaluate the straight path solution against the QUCE-generated path.

7.4.2.3 Adversarial Gradient Integration

The Adversarial Gradient Integration (AGI) [PLZ21] approach provides an alternative form of feature attribution that also produces generative counterfactual examples. The AGI method is the only path-based generative counterfactual method currently available to our knowledge and thus forms the primary focus of our comparison. We use the Individual AGI algorithm presented in their paper.

7.5 Quantitative Evaluation

To evaluate the implementation of generative counterfactual examples, we propose using the VAE loss to determine how well the counterfactual examples fit to the underlying data distribution.

7.5.1 Path-Based Uncertainty Comparison

Table 7.2: Comparison of the average path uncertainty on the generated counterfactual instances. This is experimented over 100 instances from the training and testing sets of each dataset. Here we have 1000 steps (path interpolation instances) for the Riemann approximation of every path-based approach, thus effectively 100×1000 instances. Here the lower value the better. The proposed QUCE method shows superior performance on average when comparing counterfactual path-based approaches.

Path \mathcal{L}_ϵ	Lung	Breast	Skin	Lymph	Rectal	COVID	W-BC
Train							
QUCE	0.92±0.32	0.82±0.26	0.94±0.41	0.74±0.06	0.86±0.28	1.36±0.15	0.82±0.16
IG-QUCE	0.95±0.32	0.84±0.27	0.96±0.41	0.76±0.06	0.86±0.29	1.39±0.11	0.84±0.20
AGI	1.94±1.86	1.49±0.95	1.80±1.47	0.92±0.28	2.19±2.23	2.01±0.15	0.93±0.36
Test							
QUCE	0.82±0.34	0.91±0.28	0.82±0.31	0.69±0.19	0.80±0.29	0.61±0.07	0.83±0.29
IG-QUCE	0.83±0.33	0.91±0.28	0.82±0.31	0.70±0.19	0.79±0.29	0.67±0.05	0.85±0.33
AGI	1.22±1.16	1.83±0.97	1.34±1.20	0.89±0.28	1.57±1.35	0.82±0.11	0.96±0.55

To evaluate the QUCE method, we provide a comparison of uncertainty along a path. To do this, we use a pre-trained VAE, feeding all generated instances

along any path into the VAE to determine the reconstruction error for all given instances along a path. The intuition behind this is that a smaller reconstruction error is associated with a path that better follows the data distribution and is therefore more “realistic”.

In Table 7.2 we evaluate the path uncertainty across 100 instances from both the training and test data. From this, we observe that the QUCE method provides paths that better follow the data distribution when compared against both IG and AGI on average. Here we reiterate that IG is used with the generated QUCE instance, which already aims to minimize VAE loss, and is therefore used to show the minor differences in relaxing the straight-line path requirements, although this may not always be necessary.

7.5.2 Counterfactual Uncertainty

Table 7.3: Comparison of the average reconstruction error between original instances and their generated counterfactual examples. This is experimented over 100 instances on each dataset. Here we observe that the proposed QUCE method performs best across all datasets.

Counterfactual	\mathcal{L}_ϵ	Lung	Breast	Skin	Lymph	Rectal	COVID	W-BC
Train								
QUCE	1.01	0.78	0.97	0.71	0.85	1.25	0.93	0.93
DiCE	1.97	1.02	1.48	1.10	1.27	1.57	3.63	
AGI	2.93	2.07	2.61	1.01	3.48	2.40	1.22	
Test								
QUCE	0.93	0.91	0.86	0.67	0.83	0.76	1.08	1.08
DiCE	1.95	1.09	1.33	1.18	1.22	0.92	3.00	
AGI	1.68	2.75	1.87	1.02	2.38	0.84	1.41	

We use counterfactual uncertainty as a measure to evaluate generative counterfactual examples given by the QUCE method. This is a simple measure of the average reconstruction error across the generative counterfactual examples across each instance in a dataset. We measure this against the DiCE and AGI methods, as both provide counterfactual examples in their generative process.

In Table 7.3 we present the counterfactual uncertainty over 100 instances from both the training and test datasets over each of the datasets. Here, we observe that QUCE provides a lower value with respect to uncertainty measurements compared to both the DiCE and AGI methods, implying that the instances generated by QUCE better follow the data distribution and can thus be thought of as more likely, subject to the dataset. Further experiments on the deletion

game seen in [YAWM23, AJ23], reconstruction error and a theoretical evaluation against further explainability axioms presented in [ABN22] are provided in the supplementary material.

7.6 Conclusion

In this chapter, we provide a novel approach that combines generative counterfactual methods and path-based explainers, minimizing uncertainty along generated paths and for generated counterfactual examples. We provide an analysis of the proposed QUCE method on path uncertainty, generative counterfactual example uncertainty, and proximity. Our approach provides paths that are less uncertain in their interpolations, so that more reliable gradients and explanations can be extracted. Similarly, we provide a clear explanation of uncertainty, including when and where it exists, as seen in the example explanations provided in Figure 7.2.

Part V

Enhancing Explainability - a Data Perspective

Chapter 8

Explaining Incomplete Data

Contents

8.1	Introduction	121
8.2	Background	124
8.3	Nominative Properties of Imputation and Explanation Methods . .	127
8.4	Imputation Method	130
8.5	Evaluation	132
8.6	Experiment Results	133
8.7	Conclusion	140

8.1 Introduction

The importance of eXplainable Artificial Intelligence (XAI) has surged since the GDPR’s “right to an explanation” mandate [SP17], compelling the need for explanations in inherently opaque AI models. This necessity significantly impacts the medical domain, resulting in widespread applications of XAI in medicine [TG21, ZWL22, DFB⁺21].

Feature attribution represents a collection of common techniques in XAI, with a variety of approaches found in the literature [ZHH⁺21, KLS⁺22]. These methods seek to explain predictions by quantifying the influence of individual features. Common strategies for feature attribution include leveraging interpretable models within local contexts [RS22, RSG16, LL17, ZK21, PMT18] and utilizing gradient-based methods [SCD⁺17, STY17, DFS22, DFFS23].

In the seminal work by Lundberg et al. [LL17], a pivotal paper in the field of feature attribution, the authors emphasize the “**missingness property**” shared by many attribution methods. Essentially, if a feature in a prediction instance lacks a value, it is assigned negligible (zero) importance in the explanation. While this property is generally desirable, leading state-of-the-art methods like Local

Interpretable Model-Agnostic Explanations (LIME) [RSG16], SHapley Additive exPlanations (SHAP) [LL17], and others [SGK17, BBM⁺15, ŠK14] adhere to it, it can be problematic when dealing with data containing missing values.

Consider a prediction instance where a feature is missing a value; according to the missingness property, this feature receives no attribution. This deficiency can substantially impact both the prediction and the explanation, posing challenges in providing a realistic explanation and corresponding prediction. Furthermore, the prevalence of incomplete data in Electronic Health Records (EHR) is a well-documented issue [WCNK13, HAD21]. Handling missing feature values is an ongoing and pervasive research challenge. In the literature, various imputation methods have been proposed [HC20], aiming to fill in missing data and construct complete datasets. A natural consequence of poor imputation is the chance of producing a counterfactual patient state or invoking bias, thus leading to erroneous conclusions.

For instance the k -Nearest Neighbour (kNN) imputation [JAB21] shows a consistently good performance on a collection of numeric regression and classification datasets. The Multivariate Imputation by Chained Equations (MICE) imputation method has seen applications in several medical research [HSP⁺19, RLS15], with both kNN and MICE being used simultaneously in various studies [LLY⁺23].

Similarly, other popular methods include Generative Adversarial Imputation Nets (GAIN) [DFsY⁺21] a method that utilises Generative Adversarial Networks (GAN) [GPAM⁺14], SoftImpute [MHT10] a method utilising soft-thresholded singular value decomposition and MissForest [SB11] a method that utilises the random forest method. Across the recent works of [JCL⁺22, DFsY⁺21], it is observed that MiCE, MissForest, GAIN and SoftImpute exhibit a competitive state-of-the-art performance.

However, existing feature imputation algorithms are far from perfect. Consider a patient instance taken from the SEER breast cancer dataset [TEN19]¹ as shown in Table 8.1. Suppose that Feature 7, “Tumor Size” is missing in the dataset. Different imputation methods will give different estimation to the missing value, which will result in different explanations, as summarised in Table 8.2. Note that *Surrogate Set Imputer (SSI)* is the imputation method introduced in this work. We see that the SSI method produces imputed values that are closest to the ground truth, which in turn gives better explanations. In light of the recent prominence of state-of-the-art XAI methods emphasizing the importance of local surrogate models, we extend this concept to the development of a novel imputation technique. We leverage “local neighborhood” information, as to generate locally faithful imputations. We use a simple model as to enhance the interpretability of our results. Subsequently, we assess the quality of these imputations by comparing

¹The SEER dataset at <https://ieee-dataport.org/open-access/seer-breast-cancer-data> is a classification dataset containing 11 numerical attributes.

Feature Index	Feature Name	Feature Value	SHAP Explanation
1	Age	57	-1.74
2	T STAGE	0	-0.12
3	N STAGE	0	0.45
4	6TH STAGE	0	1.37
5	GRADE	1	-2.45
6	A STAGE	1	0.14
7	TUMOR SIZE	Missing	0
8	ESTROGEN STAT.	0	-3.33
9	PROGESTERONE	0	1.55
10	REGIONAL NODE E.	22	0.94
11	REGIONAL NODE P.	1	1.35
Prediction: 81 Months Survival Time			

Table 8.1: A patient instance taken from the SEER breast cancer dataset. For illustration, we will consider Feature 7, **Tumor Size** to be missing in our comparison of imputation algorithms. Such missingness yields 0 feature attribution explanation.

Imputation Method	Imputed Value	SHAP Explanation
SSI	10.78	9.22
Iterative	34.21	-5.87
kNN	21	0.80
GAIN	23.29	0.12
SoftImpute	25.48	3.43
MissForest	18.30	5.14
Ground Truth	12	11.76

Table 8.2: Imputation results from different methods. The ground truth is Feature 7, Tumor Size with a feature value of 12. The SSI method produces the closest imputed value of 10.78. Similarly, the SSI method yields an explanation that is closest to the one computed with the feature value ground truth.

them to ground truth data. In our pursuit of creating imputations for missing data, we identify specific properties that must be satisfied to achieve improved explainability.

This study makes several contributions:

1. We introduce a set of properties that connect explanations and imputations, providing a comprehensive framework for explaining predictions with incomplete data.
2. We present an innovative method for feature imputation, designed to address the challenges of missing data and interpretability.
3. We propose metrics for evaluating both explanations and imputations, allowing for a thorough assessment of their performance.

The rest of this chapter is structured as follows. Section 8.2 provides background information on feature attribution methods and introduces one popular imputation technique. In Section 8.3, we define a set of desirable properties that imputation methods should adhere to, especially concerning feature attribution. Section 8.4 outlines the details of our proposed imputation method. Section 8.5 examines theoretical properties of the our method. To assess the performance of these methods and their impact on feature attribution, Section 8.6 presents a comprehensive collection of experiments, including both existing and newly proposed evaluation metrics.

8.2 Background

In the realm of managing EHRs and deciphering intricate medical data, understanding the significance of individual features within predictive models through a concept known as feature attribution is paramount. Feature attribution techniques shed light on the contribution of each feature to the overall predictive power of a model, aiding the interpretation of complex machine learning algorithms.

Simultaneously, MICE, an imputation method introduced in [ASFL11], is a popular method for addressing missing data within EHRs. MICE employs a series of regression-based steps to impute missing feature values, making it effective for data recovery and analysis in the medical field. More recently GAIN [DFsY⁺21] takes advantage of the GAN approach that has been popularised in the last decade, with ever increasing computation the capacity to perform well on big data has increased for such deep models.

To extrapolate, in this section we briefly introduce feature attribution and imputation methods.

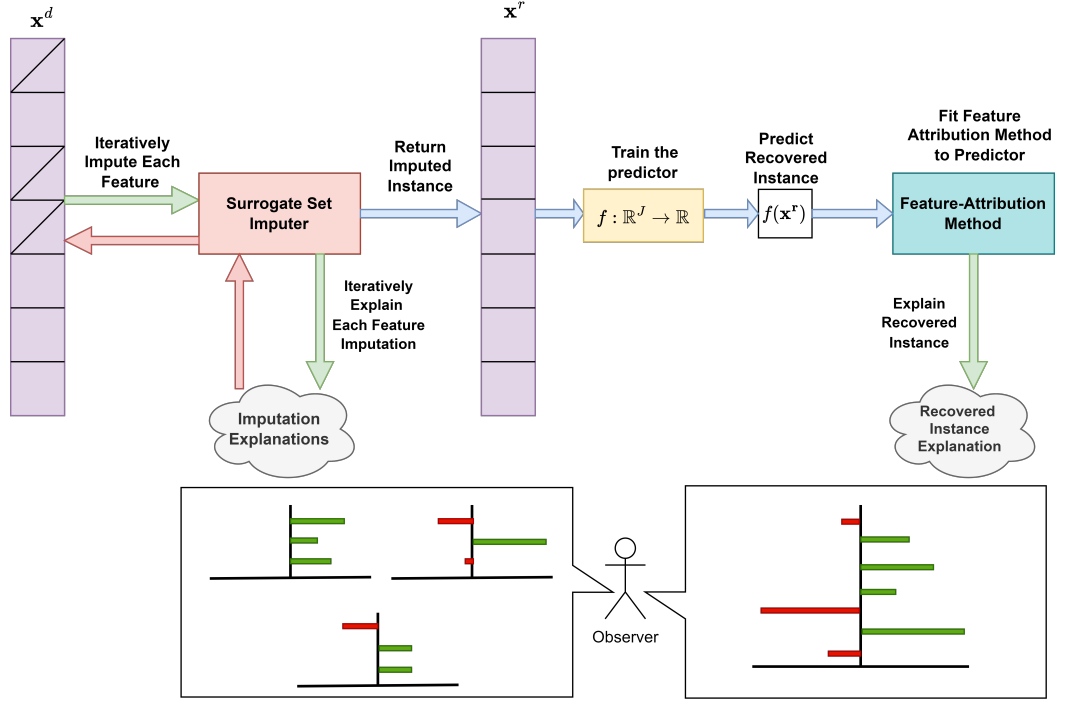


Figure 8.1: The framework of the Surrogate Set Imputer method. **Green** arrows represent returned outputs; **Red** arrows represent an iterative process; and the **Blue** represent transitions in the pipeline once the **Green-Red** iterative processes are completed.

8.2.1 Feature Attribution Methods

When dealing with a probabilistic classifier or regressor f for any instance $\mathbf{x} = \langle x^1, \dots, x^J \rangle$, we obtain a prediction probability for a given predicted class or regression value, represented as $f : \mathbb{R}^J \rightarrow \mathbb{R}$, where $f(\mathbf{x})$ signifies the probability that \mathbf{x} belongs to a specific class or produces a regression value. With the use of an explainable method Φ , we can generate explanations for an instance \mathbf{x} , where $\Phi : \mathbb{R}^J \rightarrow \mathbb{R}^J$. In other words, one has $\Phi(\mathbf{x}) = \langle \phi^1, \dots, \phi^J \rangle$, which each ϕ^j is a corresponding attribution value of a feature x^j in \mathbf{x} .

The process of attributing features when predicting a single instance is commonly known as “local” explanations, while the mean additive attribution across all instances is referred to as “global” explanations. One notable feature attribution method for explainable artificial intelligence (XAI) is Shapley Additive Explanations (SHAP) [LL17], which draws inspiration from game theoretic Shapley Values. We will not delve into further introduction here as SHAP has been continuously cited and introduced in XAI literature.

Local Interpretable Model-Agnostic Explanations (LIME) is another method that focuses on providing local explanations. LIME generates explanations by

creating a surrogate dataset through perturbations and then applying weighted linear regression with a modified k -lasso regularization over the surrogate set to produce explanations. Similar to SHAP, we omit further introduction, as it is well-documented in XAI literature; for more details, refer to [RSG16].

Later in our discussion, we will refer to the “missingness” property that feature attribution methods adhere to. As current state-of-the-art research suggests [BDGP22], additive feature attribution methods like LIME and SHAP, among others [SGK17, BBM⁺15, ŠK14], conform to this missingness property.

8.2.2 Iterative Feature Imputation Method

Multivariate Imputation by Chained Equations (MICE) is a very popular method used for Electronic Health Record (EHR) imputation. MICE [VBGO11] was introduced in [ASFL11], where the algorithm’s details can be found. The MICE imputation method considers multiple possibilities for the missing feature by employing multiple regression models. As stated in [ASFL11], there are six steps to the MICE method, and we provide an overview of each step.

1. Fill all missing feature values with a designated value, such as the mean.
2. Set the placeholder for one feature back to missing.
3. Regress the observed values given in step 2 for the feature on all other features within the imputation model, treating all other features as independent variables.
4. Replace the missing feature values with the predicted values.
5. Repeat steps 2 and 4 for each feature missing a value; this constitutes one iteration of the process. Once all feature values have been imputed,
6. Repeat steps 2 and 4 for a specified number of iterations.

Given a flawed instance \mathbf{x}^d with a missing feature value, with MICE, we can obtain a recovered version of the instance \mathbf{x}^r , which includes the imputed feature value. When this process is iterated for each instance and feature value in the set, we can obtain a recovered set X^r , where $\mathbf{x}^r \in X^r$. MICE is used as one of our comparison baselines in this work.

8.2.3 Generative Feature Imputation Method

The Generative Adversarial Imputation Nets (GAIN) [DFsY⁺21] approach, informally the GAIN architecture follows that of a GAN, such that there are two components, a generator G and discriminator D . The component G in the GAIN

architecture is a generator, the generator observes components of a true data vector and imputes synthetic data for the corresponding missing values from what is observed. It follows that the component D takes the completed vector and disseminates between what is imputed and what is not, here D is given “hints” on what may be imputed, such that G is forced to learn and fool D , thus completing the imputation process. The “hints” are given in the form of a hint matrix, which can be seen as a mask over a subset of values to impute.

8.3 Nominative Properties of Imputation and Explanation Methods

The robustness of imputed instances and their associated explanations is a crucial aspect in the field of data analysis and interpretability. It revolves around the idea that when we replace missing or defective data with imputed values, the explanations derived from these imputed instances should ideally align with those obtained from the original, complete data. This alignment ensures that the insights and conclusions drawn from the imputed data remain reliable and consistent with the overall understanding of the dataset.

To assess the robustness of imputation methods, we begin by examining certain fundamental properties that should be upheld by any feature attribution and imputation techniques. These properties serve as a baseline for evaluating the performance and reliability of such methods. Furthermore, we introduce specific metrics designed to quantify and assess the quality of both the explanations provided by these techniques and the imputations they generate. This comprehensive evaluation framework allows us to gauge the effectiveness of imputation methods in maintaining the integrity and consistency of the data and its associated interpretations.

Let \mathbf{x} be an instance. If any feature value in \mathbf{x} is missing, then we write \mathbf{x} as \mathbf{x}^d , to denote that this is a defected instance. For performance evaluation, we assume there is an oracle that gives us \mathbf{x} which contains the true values for the missing values of \mathbf{x}^d . For a measurable representation, we assume all missing values to be zero and all complete values to be non-zero.

Definition 8.1 (Instance defect) Given a defected instance \mathbf{x}^d and an oracle instance \mathbf{x} , instance defect ($ID_{\mathbf{x}}$) is

$$ID_{\mathbf{x}} = \|\mathbf{x}\|_0 - \|\mathbf{x}^d\|_0. \quad (8.1)$$

Instance defect is the amount of missingness that exists for an instance \mathbf{x} .

To address instance defects, we explore various imputation methods to fill in the missing values. Let’s consider an imputation method λ , where $\mathbf{x}^r = \lambda(\mathbf{x}^d)$. When applied to a defected instance using this method, we obtain the recovered

sample \mathbf{x}^r , which now includes imputed feature values. This process demonstrates the potential for imputation to reduce instance defects.

Property 5 (Imputation Gain). *For a given oracle instance \mathbf{x} and its corresponding recovered instance \mathbf{x}^r , we can express the relationship as follows:*

$$\|\mathbf{x}\|_0 - \|\mathbf{x}^r\|_0 \leq ID_{\mathbf{x}}. \quad (8.2)$$

The concept of imputation gain asserts that the imputation process should not exacerbate the defects in any instance \mathbf{x} . Consider an original instance \mathbf{x} such that $\|\mathbf{x}\|_0 = |\mathbf{x}|$, where $|\cdot|$ represents the cardinality of a vector or the absolute value for a single value, and $\|\cdot\|_0$ denotes the l_0 norm of a vector, which is essentially the count of non-zero terms. Now, suppose we have a defected instance \mathbf{x}^d such that $\|\mathbf{x}^d\|_0 \leq \|\mathbf{x}\|_0$. When we generate a recovered sample \mathbf{x}^r through imputation, we expect an imputation gain, specifically $\|\mathbf{x}\|_0 - \|\mathbf{x}^r\|_0 \leq ID_{\mathbf{x}}$. This conclusion assures us that imputation will not worsen instance defects and, in fact, demonstrates an imputation gain.

In the context of one-to-one imputation for two defected instances, we can establish the property of imputation monotonicity as follows.

Property 6 (Imputation Monotonicity). *Given an oracle instance \mathbf{x} and two defected instances with their respective recovered instances, we have the following relationship:*

$$\begin{aligned} \text{if } \|\mathbf{x}\|_0 - \|\mathbf{x}_1^d\|_0 &\leq \|\mathbf{x}\|_0 - \|\mathbf{x}_2^d\|_0, \\ \text{then } \|\mathbf{x}\|_0 - \|\mathbf{x}_1^r\|_0 &\leq \|\mathbf{x}\|_0 - \|\mathbf{x}_2^r\|_0. \end{aligned}$$

Imputation monotonicity implies that in the case where one defected instance \mathbf{x}_1^d with respect to the oracle instance \mathbf{x} has fewer defects than \mathbf{x}_2^d , the corresponding recovered instance \mathbf{x}_1^r will have fewer or an equal number of defects compared to \mathbf{x}_2^r .

Now, let's introduce the concept of Explanation Defect and Explanation Gain:

Definition 8.2 (Explanation Defect) Given a complete instance \mathbf{x} , a defected instance \mathbf{x}^d , and a feature attribution method Φ , where $\Phi : \mathbf{x} \rightarrow \mathbb{R}^J$ for J features, the explanation defect ($ED_{\mathbf{x}}$) is defined as:

$$ED_{\mathbf{x}} = \|\Phi(\mathbf{x})\|_0 - \|\Phi(\mathbf{x}^d)\|_0. \quad (8.3)$$

Property 7 (Explanation Gain). *Given a feature attribution method Φ , the explanation gain is defined as the difference between the l_0 norm of the explanation of a complete instance and the l_0 norm of the explanation of a recovered instance:*

$$\|\Phi(\mathbf{x})\|_0 - \|\Phi(\mathbf{x}^r)\|_0 \leq ED_{\mathbf{x}}. \quad (8.4)$$

For a feature attribution method Φ adhering to the missingness property introduced in [LL17], consider an explanation of a complete instance, where the l_0 norm of $\Phi(\mathbf{x})$ equals the cardinality of the vector, i.e., $\|\Phi(\mathbf{x})\|_0 = |\mathbf{x}|$. Now, if we have an explanation for a defected instance, such that $\|\Phi(\mathbf{x}^d)\|_0 < \|\Phi(\mathbf{x})\|_0$, then the explanation gain can be defined as $\|\Phi(\mathbf{x})\|_0 - \|\Phi(\mathbf{x}^d)\|_0 \geq 1$. Consequently, for an explanation of an imputed instance $\Phi(\mathbf{x}^r)$ obtained through any imputation method λ , the explanation gain is bounded by $ED_{\mathbf{x}}$.

Finally, let's introduce Explanation Monotonicity:

Property 8 (Explanation Monotonicity). *Given any instance \mathbf{x} , two defected instances, and a feature attribution method Φ , we have the following relationship:*

$$\begin{aligned} \text{if } \|\Phi(\mathbf{x})\|_0 - \|\Phi(\mathbf{x}_1^d)\|_0 &\leq \|\Phi(\mathbf{x})\|_0 - \|\Phi(\mathbf{x}_2^d)\|_0, \\ \text{then } \|\Phi(\mathbf{x})\|_0 - \|\Phi(\mathbf{x}_1^r)\|_0 &\leq \|\Phi(\mathbf{x})\|_0 - \|\Phi(\mathbf{x}_2^r)\|_0. \end{aligned}$$

Explanation Monotonicity states that if one defected instance has fewer defects in its explanation compared to another defected instance, then the corresponding recovered instance's explanation will also have fewer or an equal number of defects compared to the other recovered instance.

Property 9 (Recovery). *Given an imputation method λ , a defected instance \mathbf{x}^d , and an oracle instance \mathbf{x} , the property of recovery states:*

$$\lambda(\mathbf{x}^d) = \mathbf{x}.$$

Property 10 (Explanation Recovery). *Suppose we have a feature attribution method represented as Φ , a recovered instance \mathbf{x}^r , and an oracle instance \mathbf{x} . Explanation recovery dictates that when we apply the feature attribution method Φ to the recovered instance \mathbf{x}^r , we should obtain an explanation that is identical to that of the oracle instance \mathbf{x} :*

$$\Phi(\mathbf{x}^r) = \Phi(\mathbf{x}).$$

The existence of these properties is pivotal as they serve as indicators of the reliability and accuracy of imputation and feature attribution methods. When these properties hold true, they ensure that the imputation method effectively addresses the instance defect without introducing additional errors. Furthermore, they signify that less imputation is required when an instance has fewer defects.

Similarly, the properties indicate that the explanation provided for a defected instance contains fewer features with attribution than any recovered instance from an imputation method. Consequently, both the values of instance features and the explanations for those features increase when \mathbf{x}^d is successfully recovered. In essence, these properties establish that imputation methods should be able to faithfully and consistently recover and explain defected instances, thus ensuring data integrity and interpretability in various applications.

8.4 Imputation Method

Given a dataset X with N instances and J features, let \mathbf{x} be an instance in X , then considering some defected instance \mathbf{x}^d , through an imputation method on a defected instance $\lambda(\mathbf{x}^d)$ we obtain a recovered sample \mathbf{x}^r . To impute the defected instance \mathbf{x}^d , we obtain the k -Nearest Complete Neighbours (k -NCN) of \mathbf{x}^d by evaluating each complete instance $\mathbf{x}^c \in X$ where k is the number of nearest complete neighbours that are utilised. Upon evaluating k -NCN, we produce a local surrogate model by taking the union over all surrogate sets with respect to k -NCN, we use this union of local surrogate sets to predict the missing feature value as a form of feature imputation that is inherently interpretable. Formally, we introduce our *Surrogate Set Imputer* (SSI) method as follows.

Given a dataset X , we let $X^c = \{\mathbf{x} \in X | \mathbf{x} \text{ is complete}\}$. To do feature imputation on a defected instance $\mathbf{x}^d \in X$, we first find a set of k nearest complete neighbours $N_s = \langle \mathbf{x}_s^1, \dots, \mathbf{x}_s^k \rangle$ such that

- $N_s \subseteq X^c$, and
- for any $\mathbf{x}' \in X^c \setminus N_s$ and $\mathbf{x}_s^i \in N_s$, it holds that

$$\delta(\mathbf{x}^d, \mathbf{x}') \geq \delta(\mathbf{x}^d, \mathbf{x}_s^i).^2$$

From each $\mathbf{x}_s^i \in N_s$, we further construct a surrogate set $\mathcal{Z}_{\mathbf{x}_s^i} = \langle \mathbf{z}_1, \dots, \mathbf{z}_N \rangle$ with a multivariate Gaussian distribution

$$\mathcal{Z}_{\mathbf{x}_s^i} \sim \mathcal{N}(\mathbf{x}_s^i, \Sigma),$$

where Σ is the covariance matrix for X . Subsequently, we produce a combined surrogate set $\mathcal{K}_{\mathbf{x}^d}$ with:

$$\mathcal{K}_{\mathbf{x}^d} = \bigcup_{i=1}^k \mathcal{Z}_{\mathbf{x}_s^i}. \quad (8.5)$$

Then, let j be the feature that is missing in \mathbf{x}^d . We train a predictor g to predict the missing feature using $\mathcal{K}_{\mathbf{x}^d}$. In other words, let \mathbf{h}^j be the values in $\mathcal{K}_{\mathbf{x}^d}$ for feature j and R be the remaining values, for each $\mathbf{r} \in R$, we construct g such that $g(\mathbf{r}) = h$.

To conform to the locality of feature imputation, we further employ a set of weights $\tau_{\mathbf{x}^d}$. We ensure that data points closer to \mathbf{x}^d having a greater influence as the weights decay when data points moving away from \mathbf{x}^d . This is defined as

$$\tau_{\mathbf{x}^d} = e^{-\delta(\mathbf{x}^d, \mathbf{r})^2 / \sigma^2} \quad (8.6)$$

²The function δ calculates the distance between instances as given in Algorithm 2.

with $\sigma = 0.75\sqrt{N}$ the kernel width. Details of this weighting can be found in [RSG16].

To fit a linear model over our data R and labels \mathbf{h}^j , we minimise the sum of squares error for the coefficients $\beta = \langle \beta^1, \dots, \beta^J \rangle$ with:

$$\mathcal{L}(\beta) = (\mathbf{h}^j - R\beta)^T \tau_{\mathbf{x}^d} (\mathbf{h}^j - R\beta) \quad (8.7)$$

and solve for coefficients β to approximate the weighted sum of squares error set to zero, we have:

$$\beta = (R^T \tau_{\mathbf{x}^d} R)^{-1} R^T \tau_{\mathbf{x}^d} \mathbf{h}^j, \quad (8.8)$$

which is used to produce a set of predicted values for the known values \mathbf{h}^j . Finally, the prediction is given by:

$$\hat{\mathbf{h}} = R\beta. \quad (8.9)$$

We note that this formulation implies we focus on the imputation of continuous features. We summarise the entire process in Algorithm 2.

Algorithm 2 Surrogate Set Imputer (SSI)

\mathbf{x}^d is a defected instance, k is the number of nearest complete neighbours

$i = 1$

$\mathcal{K}_{\mathbf{x}^d} = []$

procedure SSI(\mathbf{x}^d)

 set each missing feature value to the feature mean

for each missing feature index in \mathbf{x}^d **do**

while $i \leq k$ neighbours **do**

$\mathbf{x}_i^s \leftarrow i^{th}$ nearest complete neighbour of \mathbf{x}^d

 generate \mathcal{Z}_i from $\mathcal{N}(\mathbf{x}_i^s, \Sigma)$

 append \mathcal{Z}_i to $\mathcal{K}_{\mathbf{x}^d}$

 increment i

end while

 fit weighted linear regression over $\mathcal{K}_{\mathbf{x}^d}$

 predict the missing feature

end for

return recovered instance \mathbf{x}^r

end procedure

8.4.1 Multiple Value Imputation

In the case of multiple missing values, we introduce a multiple value imputation variant of SSI. For this, imputation is carried out in an iterative process such that one missing value is imputed at a time using Algorithm 2. We present experimental results for datasets with multiple missing values in Section 8.6 and the Appendix.

8.4.2 Interpreting Imputed Values

One chief advantage of predicting the missing value with a linear model as described with Equations 8.7 to 8.9 is that they allow a direct computation of feature attributions for the imputed values. Namely, one can calculate the linear SHAP values. In general, SHAP values take the following additive form:

$$f(\mathbf{x}) = \phi^0 + \sum_{j=1}^J \phi^j x^j \quad (8.10)$$

Here ϕ^0 is the attribution such that no features are present and ϕ^j is the attribution calculated for feature j of an instance \mathbf{x} . In the linear formulation shown in [ŠK14], we see that linear SHAP values are directly computed from β as:

$$\phi^j = \beta^j (x^j - \mathbb{E}[X^j]), \quad (8.11)$$

where β^j is the linear coefficients for the j^{th} feature of β . Thus in the context of our surrogate set, one can write the attribution towards an imputed feature to be given as

$$\phi^j = \beta^j (x^{d,j} - \mathbb{E}[R^j]). \quad (8.12)$$

Here $\mathbb{E}[R^j]$ corresponds to the expected value of the j^{th} feature for the set R , corresponding to the defected instance feature $x^{d,j}$ that we are explaining when imputing.

Note that due to the SHAP value formulation, the SSI imputation method and associated attribution values adhere to the set of key game theoretic properties as given by the SHAP method.

8.5 Evaluation

In evaluating the proposed SSI imputation method, we first show how SSI satisfies the proposed properties 5-8 as follows.

Proposition 10. *Any instance with a missing value can successfully be imputed by SSI.*

Proof. As a direct consequence from the SSI algorithm, there will always exist a surrogate set R and labels \mathbf{h}^j , since:

$$\beta = (R^T \tau_{\mathbf{x}^d} R)^{-1} R^T \tau_{\mathbf{x}^d} \mathbf{h}^j,$$

we can access $\beta \cdot \mathbf{x}^d$ thus an imputation is always accessible. It is clear that an imputed value can be provided for any instance, under the assumption that there exists at least one instance that is complete $\mathbf{x}^c \in X$. \square

Proposition 11. *The SSI imputation method satisfies properties 5 and 6.*

Proof. Consider an oracle instance $\mathbf{x} \in \mathbb{R}^J$, where $\|\mathbf{x}\|_0 = J$ and $J \in \mathbb{N}$ and a defected instance where $\|\mathbf{x}^d\|_0 = (J - 1)$ then the instance defect given by Definition 8.1 is

$$ID_{\mathbf{x}} = J - (J - 1) = 1.$$

Given an imputation method λ that provides a value for any missing instance, we have $\|\mathbf{x}^r\|_0 = J$. Thus $\|\mathbf{x}\|_0 - \|\mathbf{x}^r\|_0 = (J - J) = 0 \leq ID_{\mathbf{x}}$. Extrapolating on this, we show this holds given any integer $\kappa (1 \leq \kappa < J)$, such that a defected instance $\|\mathbf{x}^d\|_0 = (J - \kappa)$ then the instance defect is given as $ID_{\mathbf{x}^d} = J - (J - \kappa)$. Since SSI can provide imputations for any missing values we have $\|\mathbf{x}^r\|_0 = J$, and therefore it is clear that $\|\mathbf{x}\|_0 - \|\mathbf{x}^r\|_0 = (J - J)$ where $(J - J) \leq J - (J - \kappa) \implies 0 \leq ID_{\mathbf{x}}$. Both Properties 5 and 6 hold. \square

Proposition 12. *The SSI method satisfies Properties 7 and 8.*

Proof. Direct from Proposition 10 and 11, it is trivial that properties 5 and 6 are directly satisfied. \square

8.6 Experiment Results

To evaluate the introduced imputation method empirically, we consider a collection of EHRs that are publicly available and evaluate SSI on them, in comparison with existing methods in the literature.

8.6.1 Datasets

The datasets used for the experimental study are summarised in Table 8.3. The rest of this section introduces each of the datasets with running examples of the imputation method and associated explanations.

8.6.1.1 The Simulacrum

The Simulacrum, a synthetic dataset developed by Health Data Insight CiC derived from anonymous cancer data provided by the National Cancer Registration and Analysis Service, which is part of Public Health England³, provides data used in this study.

We extract five datasets from the Simulacrum. We isolate a classification problem of survival time and determine the patient cohorts using the lung cancer ICD-10 code “C34” *malignant neoplasm of bronchia and lung*, breast cancer ICD-10 code “C50” *Malignant neoplasm of breast.*, lymphoma cancer ICD-10 code

³<https://simulacrum.healthdatainsight.org.uk/>

Table 8.3: An overview of each dataset.

Dataset	Instances	Features	Type
Simulacrum Breast Cancer	1750	23	Classification
Simulacrum Lung Cancer	1750	23	Classification
Simulacrum Rectal Cancer	1750	23	Classification
Simulacrum Lymphoma Cancer	1750	23	Classification
Simulacrum Skin Cancer	1750	23	Classification
Wisconsin Breast Cancer	569	30	Classification
SEER Breast Cancer	4024	11	Classification
Diabetes	422	10	Regression

“C83” *Non-follicular lymphoma.*, skin cancer “C44 and C90” *malignant neoplasm of the sebaceous glands / sweat glands and multiple myeloma and malignant plasma cell neoplasms*, and rectal cancer “C20” *Malignant neoplasm of rectum.*, from within the Simulacrum dataset. Each dataset contains a subsample patient cohort, containing 1750 patients. Table 8.4 summarises the cancer code used in identifying patients in Simulacrum.

Table 8.4: ICD-10 codes that form the associated datasets of the Simulacrum.

ICD-10	Dataset
C34	SLC
C50	SBC
C84	SLyC
C44 & C90	SSC
C20	SRC

For each dataset, we use the eXtreme Gradient Boosting (XGBoost) [CG16], in predicting a patient’s six-month mortality. Consider a single patient for whom we have access to the true values denoted as \mathbf{x} . We randomly remove a feature value, resulting in a defective instance \mathbf{x}^d . We then evaluate several imputation methods: SSI, MICE, GAIN, SoftImpute, MissForest and kNN [BS16] with such defective instances. Each of these methods generates a recovered sample denoted as \mathbf{x}^r . In other words, for each method $\lambda \in \{\text{SSI, MICE, GAIN, SoftImpute, MissForest, kNN}\}$, $\lambda(\mathbf{x}^d)$ produces the imputed instance. Note that, for MICE, we use the *iterative imputer* (Iterative) implementation introduced in scikit-learn.⁴ and for the DICE, SoftImpute and MissForest methods we make use of the HyperImpute library

⁴<https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html>

[JCL⁺22]. We start by comparing single value imputations against the ground truth, with some results shown in Table 8.5.

Table 8.5: Example of the imputed values when compared to the ground truth for each imputation method applied to a single instance from each of the datasets. Values closer to the ground truth are better.

Imputation	Ground Truth	SSI	Iterative	kNN	GAIN	SoftImpute	MissForest
Patient (SLC)	68	67.78	73.75	67.70	70.21	70.59	73.27
Patient (SBC)	88.70	75.10	70.53	73.83	69.88	68.43	71.07
Patient (SLyC)	73	74.58	69.43	77	70.32	50.27	71.32
Patient (SSC)	73	73.37	71.73	69.33	70.58	74	70.30
Patient (SRC)	59.10	63.81	69.45	69.14	54.20	71.90	68.09
Patient (W-BC)	101.70	101.71	104.58	103.46	98.34	102.60	100.55
Patient (Diabetes)	-0.06	-0.06	-0.01	-0.05	-0.05	-0.06	-0.03
Patient (SEER-BC)	12	10.78	34.21	21	23.29	25.48	18.30

Empirically, we observe close imputations to the ground truth for selected instances in Table 8.5. Additionally, we notice the variance in imputed values, which arises from the random distribution within local neighborhoods for k -NCN. As a result, the local observations are dependent on the random seed. Similarly, we can also examine the SHAP values for local instances and their proximity to the ground truth in Table 8.6. We observe that closer value imputation mostly results in a closer explanation to the ground truth.

Table 8.6: Example of SHAP values returned by the SHAP method comparing each imputation method against the ground truth on a patient instance from each of the datasets. Values closer to the ground truth are better.

Imputation	Ground Truth	SSI	Iterative	kNN	GAIN	SoftImpute	MissForest
Patient (SLC)	-0.56	-0.52	-0.16	-0.52	-0.09	-0.08	0.22
Patient (SBC)	-0.29	-0.31	-0.09	-0.45	0.05	≈ 0	-0.21
Patient (SLyC)	0.37	0.56	-1.73	-0.99	-2.08	0.43	-1.89
Patient (SSC)	0.09	0.09	1.98	0.95	1.03	-1.13	1.07
Patient (SRC)	-1.26	-0.54	0.55	0.61	0.61	0.47	0.84
Patient (W-BC)	1.35	1.35	0.71	0.71	1.38	0.71	1.38
Patient (Diabetes)	-33.01	-36.20	-17.84	-36.22	-36.22	-36.20	-29.13
Patient (SEER-BC)	11.76	9.22	-5.87	0.80	0.12	3.43	5.14

8.6.1.2 Scikit-learn Datasets

In addition to the Simulacrum, we also conducted experiments on two scikit-learn datasets: the Wisconsin Breast Cancer dataset⁵, which contains 569 instances,

⁵[sklearn.datasets.load_breast_cancer.html](https://sklearn.org/datasets/load_breast_cancer.html)

and the Diabetes dataset⁶, which contains 422 instances. These datasets represent a classification problem and a regression problem, respectively. The breast cancer dataset includes a dependent feature used to predict whether a tumor is malignant or benign based on a set of independent tumor measurements. The diabetes dataset contains a dependent regression target feature used to determine the progression of diabetes after one year. We provide two imputed examples along with their associated explanations in Table 8.5 and Table 8.6, respectively. We see that the results are similar to the ones found with the Simulacrum dataset in that our approach (SSI) gives the best results than baseline methods for imputation and nearly the best for explanations.

8.6.1.3 SEER Breast Cancer

Lastly, we also adopt the SEER Breast Cancer dataset from the IEEE dataport made publically available in [TEN19]. The dataset contains a cohort of 4024 female breast cancer patients. The dependent feature of this dataset is the survival months of a patient, where the independent feature are a patient and tumour characteristics. We provide an imputed patient example and associated explanations given in Tables 8.5 and 8.6, respectively. Again, we see that the proposed SSI method works better than all other approaches for both imputation and explanation.

8.6.2 Performance Evaluation

To perform a systematic evaluation, we propose two metrics to evaluate the satisfiability of Properties 5 and 6, respectively. Consider a complete dataset X , which is a dataset without any missing value. To evaluate we iterate over each instance and drop each value for a single feature, then we replace the missing feature with the recovered values from imputation methods, thus producing the recovered data set X^r .

We determine the average difference in imputation ($\Delta\mathcal{I}$), over the imputed feature j for an imputation method of a single instance with

$$\Delta\mathcal{I}(X, X^r) = \sqrt{\frac{1}{|X^r|} \sum_{\mathbf{x} \in X} (x^j - x^{r,j})^2}. \quad (8.13)$$

Intuitively, $\Delta\mathcal{I}$ is the Root-Mean-Square Error (RMSE) defined with respect to the recovered dataset X^r . It assesses imputation error over partially recovered instances in X^r , with a specific focus on the imputed features. Similar to RMSE, a smaller error indicates better imputation performance.

⁶`sklearn.datasets.load_diabetes.html`

Following this, we introduce the concept of explanations being *faithful*. Let us consider an explanation vector for a feature j cross the entire dataset X . We denote this as $\mathbf{e}_j = \langle \phi_1^j, \dots, \phi_i^j, \dots, \phi_N^j \rangle$ for N instances. In other words, \mathbf{e}_j denotes the explanation for a feature j in X . Note that X is the “oracle” dataset containing complete instances. Then, we consider the recovered dataset X^r . We denote the explanation vector for a feature j on X^r with \mathbf{e}'_j .

From \mathbf{e}_j and \mathbf{e}'_j , we can then evaluate the “*Explanation Faithfulness (EF)*” of the explanations for any feature j against the explanations computed on the “oracle” as:

$$\text{EF}(\mathbf{e}_j, \mathbf{e}'_j) = \frac{\mathbf{e}_j \cdot \mathbf{e}'_j}{\|\mathbf{e}_j\| \|\mathbf{e}'_j\|} \quad (8.14)$$

Intuitively, EF represents the cosine similarity calculated across all explanations generated from imputed features. This metric serves as a means to gauge the similarity between explanations computed on the complete (oracle) instance and those generated for the recovered instance. When assessing explanations, it is arguably more significant to evaluate the alignment of each feature’s contribution in a feature attribution explanation, i.e., whether it supports or undermines the prediction, rather than merely calculating the absolute distance between explanations. The introduction of such metrics enables us to evaluate the quality of imputation methods from various perspectives.

To evaluate the imputation strategy, we explore each oracle \mathbf{x} and the associated recovered imputed instance \mathbf{x}^r , by masking the true feature value giving \mathbf{x}^d and comparing the imputation against the ground truth using the proposed metrics. Here, we can empirically observe closer satisfiability of Properties 5 and 6 through our proposed methods when compared against kNN, mean and iterative (MICE) imputation.

Table 8.7: Performance of imputation methods returned for the defected instances \mathbf{x}^d , for each instance averaged over in their respective datasets. These compared using oracle instances \mathbf{x} and the recovered instance \mathbf{x}^r . Here we observe the top 3 methods in the datasets ordered from best performing (1) to third best (3). The lower the value the better.

$\Delta\mathcal{I}$	SSI	Iterative	kNN	GAIN	SoftImpute	MissForest
SSC	113.30 (1)	115.38 (3)	138.63	127.37	126.88	113.68 (2)
SLC	156.63 (1)	172.37	198.90	170.26 (3)	170.99	162.40 (2)
SBC	107.70 (3)	108.18	150.66	117.06	107.15 (2)	105.79 (1)
SRC	105.17 (2)	116.41	103.74 (1)	122.49	115.22	108.82 (3)
SLyC	7.15 (1)	9.52	8.39 (2)	15.23	11	8.81 (3)
W-BC	113.69 (1)	438.31	115.85 (2)	144.60	511.10	143.55 (3)
Diabetes	0.31 (1)	0.79	0.83	0.75	0.68 (3)	0.64 (2)
SEER-BC	3.60 (1)	140.78 (3)	166.77	179.71	241.02	112.08 (2)

Table 8.8: Explanation faithfulness of the imputation methods across the complete instances and recovered instances, for every instance averaged over in their respective datasets. Here we observe the top 3 methods in the datasets ordered from best performing (1) to third best (3). The higher the value the better.

EF(e, e')	SSI	Iterative	kNN	GAIN	SoftImpute	MissForest
SSC	0.77 (3)	0.72	0.81 (2)	0.73	0.82 (1)	0.72
SLC	0.83 (2)	0.74	0.85 (1)	0.79	0.82 (3)	0.79
SBC	0.88 (2)	0.90 (1)	0.85	0.86	0.84	0.87 (3)
SRC	0.89 (2)	0.87	0.92 (1)	0.87	0.88 (3)	0.88 (3)
SLyC	0.87 (2)	0.82	0.89 (1)	0.82	0.84 (3)	0.82
W-BC	0.99 (1)	0.93	0.98 (2)	0.98 (2)	0.96 (3)	0.98 (2)
Diabetes	0.95 (1)	0.77	0.73	0.80	0.82 (3)	0.83 (2)
SEER-BC	0.99 (1)	0.54	0.60 (3)	0.64 (2)	0.58	0.60 (3)

In this set of experiments, we observe on each dataset how close the imputed feature value is to the ground truth, and similarly how close the explanation is to the ground truth. The performance metrics in this section suggest that our proposed SSI method produces the lowest error on average, with closer property satisfiability on both Properties 9 and 10.

8.6.3 Multiple Value Imputation

To represent the performance over multiple value imputation, we conduct experiments on instances with different number of missing values. Namely, we evaluate imputation quality as the defect $\|\mathbf{x}^d\|_0$ goes to $\|\mathbf{x}^d\|_0 = |\mathbf{x}| - 1$ in discrete steps from 1. We determine the accuracy of the imputation with respect to the ground truth by evaluating the ΔI and EF metrics for multiple value imputation. For the benefit of computation, we evaluate over 100 randomly selected instances when imputing all continuous features in the respective datasets, from this we can easily compare the imputed values against the ground truth. To achieve this, we increase the number of missing features for each instance randomly and evaluate the ΔI and EF metrics. Generally speaking we observe that the proposed SSI approach tackles the multiple value imputation problem better than both kNN and MICE imputation methods. For illustration we provide an example on the W-BC dataset as every feature is continuous, this can be seen in Figures 8.2 and 8.3, where other results are omitted to Figure F.1 and F.1 in the Appendix.

8.6.4 Imputation Runtime

The runtime experiments are evaluated with the following hardware: Intel(R) Core(TM) i7-8565U CPU at 1.80GHz, 16 Gigabyte RAM and an NVIDIA GeForce

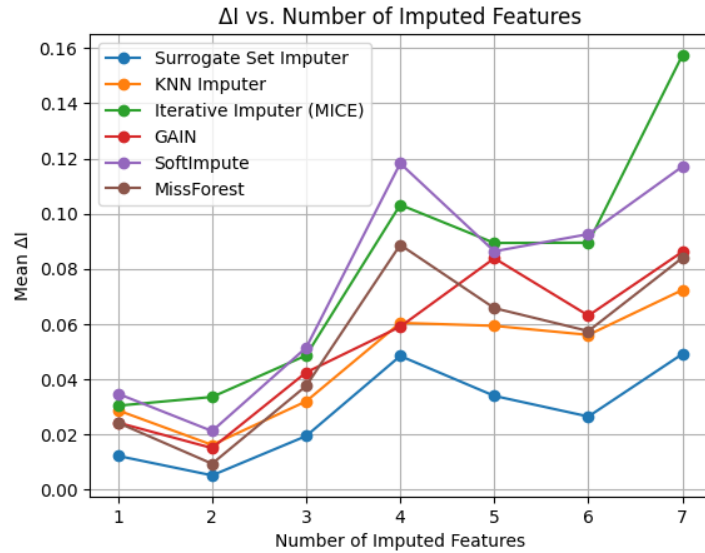


Figure 8.2: The mean ΔI for the SSI, kNN, MICE, GAIN, SoftImpute and MissForest imputation methods on the W-BC dataset as the number of missing features is increased randomly for each instance. Here we observe a lower average error for the SSI method.

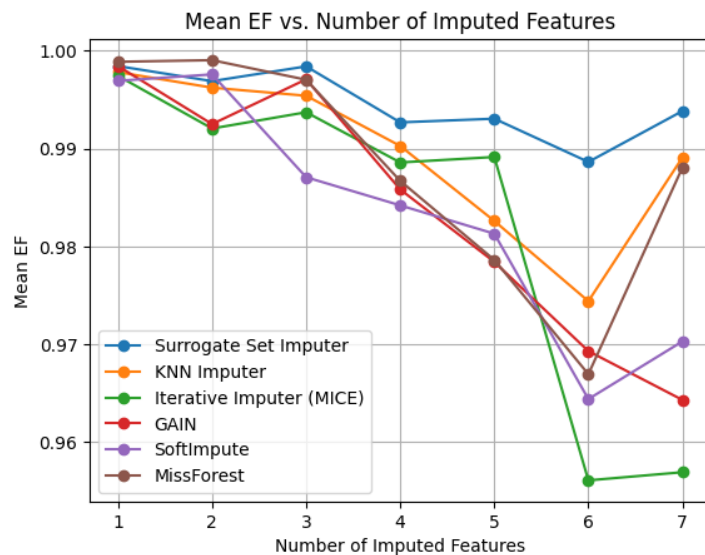


Figure 8.3: The mean EF for the SSI, kNN, MICE, GAIN, SoftImpute and MissForest imputation methods on the W-BC as the number of missing features is increased randomly for each instance. Here we observe more similar explanations produced using SSI when compared to the ground truth.

MX250 GPU. Here we take the average runtime of the imputation methods in seconds, where we take the average runtime for imputing each number of missing features over 100 instances. In Table 8.9 we observe the best runtime across the SSI, kNN and MICE imputation methods.

Table 8.9: We evaluate the runtime of the imputation algorithms evaluated in this work. Here we observe that SSI, MiCE and kNN imputation methods have the quickest run times when compared to GAIN, SoftImpute and MissForest.

Mean Runtime (seconds)	SSI	Iterative	kNN	GAIN	SoftImpute	MissForest
SSC	0.31±0.02	0.31±0.02	0.25±0.01	12.76±0.29	23.11±18.03	13.22±0.98
SLC	0.24±0.02	0.26±0.008	0.20±0.01	13.29±0.65	13.43±0.20	15.10±2.78
SBC	0.23±0.03	0.24±0.02	0.18±0.01	13.21±0.62	17.37±12.48	12.43±1.73
SRC	0.32±0.03	0.31±0.02	0.25±0.01	13.67±0.22	13.91±0.19	14.37±1.36
SLyC	0.32±0.03	0.22±0.02	0.17±0.008	13.41±0.16	14.58±0.69	14.64±2.59
W-BC	0.03±0.006	0.09±0.03	0.01±0.003	14.13±0.37	10.99±8.31	13.12±2.00
Diabetes	0.01±0.004	0.57±0.09	0.44±0.04	9.00±0.27	6.55±4.96	12.41±3.11
SEER-BC	1.57±0.11	1.11±0.13	1.03±0.06	9.61±0.17	12.15±0.48	16.35±2.93

8.6.5 Attribution for Imputed Values

For imputation interpretability, one can utilise the SSI method due to the inherent interpretability of the linear model that is used for imputation. Thus we can observe how each feature attributed towards the imputed prediction. In this section, we provide an example of imputation attribution for single instance (see Figure 8.4). Here we generate an interactive graph where one can view the feature values, and the associated attribution with respect to a target feature (feature to impute) and the imputed value.

8.7 Conclusion

In this chapter, we tackled the challenge of enhancing the explainability of data with missing values, particularly in the context of EHRs. This is motivated by addressing the “missingness property” of existing feature attribution algorithms in XAI, as which assign 0 attribution to the feature with a missing value, thus rendering feature attribution algorithm less effective when dealing with incomplete data containing missing values.

To address this issue, we introduced a novel feature imputation technique, *Surrogate Set Imputer (SSI)*, inspired by local feature attribution methods. Our approach utilizes local surrogate models and synthetic samples in the vicinity of missing values to impute them more accurately. This technique is shown to be effective on eight different datasets we have experimented.

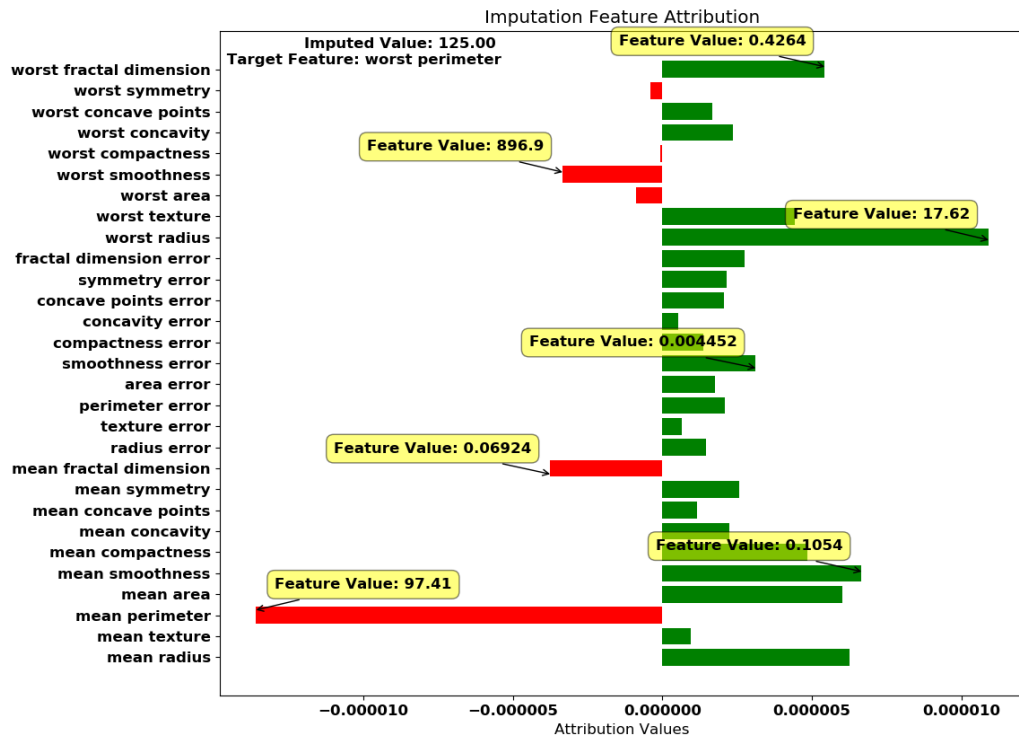


Figure 8.4: An example explanation for imputation given for an instance taken from the the Wisconsin Breast Cancer dataset, providing the imputed feature values, attribution values and magnitude of importance.

We also introduced a set of properties that connect explanations and imputations, providing a comprehensive framework for explaining predictions with incomplete data. Namely, for both feature value recovery and explanation attribution, (1) imputation should recover information and not causing damage; (2) information recovery with imputation should be monotonic; and (3) the ultimate goal of imputation is to recover the complete instance. These properties serve as guidelines for achieving improved explainability in the presence of missing data. Moreover, we have proposed two metrics for evaluating imputations and explanations on incomplete data, enabling a thorough assessment of their performance.

We believe that our work paves the way for more accurate and informative explanations in AI models, especially in domains like healthcare, where data incompleteness is a common issue. As AI models continue to play a critical role in decision-making, it is essential to ensure that their predictions are not only accurate but also transparent and interpretable. Our research contributes to achieving this goal by improving the explainability of AI models in the presence of missing data.

Part VI

Enhancing Explainability - a User Perspective

Chapter 9

ExMed: An AI Tool for Experimenting Explainable AI Techniques on Medical Data Analytics

Contents

9.1	Introduction	145
9.2	Related Work	146
9.3	ExMed Workflow	147
9.4	Case Study I: COVID-19 Control Measures	148
9.5	Case Study II: Lung Cancer Life Expectancy	152
9.6	Conclusion	154

9.1 Introduction

Explainable AI (XAI) has drawn tremendous attention in the recent years [Mil19]. XAI systems not only aim to make intelligent decisions or accurate predictions, but also provide an insight into the process of AI decision making [AB18]. A goal of enabling explainability in AI systems “is to ensure algorithmic predictions and any input data triggering those predictions can be explained” [CPC19]. In the context of Machine Learning (ML), XAI focuses on developing human-understandable prediction models producing explanations, along with predictions and model agnostic techniques that generate explanations to existing ML models. However, current XAI software implementations are scattered across multiple libraries written in different programming languages and predominately intended for data science developers rather than domain experts.

In this paper, we present , a self-contained XAI toolkit for domain experts. ExMed performs XAI analysis for prediction models. With its simple user interface, it supports both *global* explanations presenting patterns of the entire dataset and *instance* explanations that are local to individual predictions, for both classification and regression tasks. Although various XAI techniques have been proposed in recent years – e.g., a good overview of these techniques is presented in [Mol19] – we focus on feature attribution explanation techniques [LL17] due to the transparency of their explanations, their computational effectiveness and general popularity.

To better illustrate the work, we present two real world case studies that demonstrates ExMeds functionalities. In case study I, a COVID-19 transmission study reveals how different COVID-19 control measures were used and impacted the virus transmission rates. In case study II, we examine lung cancer patient life expectancy using the Simulacrum dataset.¹ Through the two case studies, we illustrate how ExMed can be used for making predictions and generating explanations.

9.2 Related Work

The use of Machine Learning (ML) has become more prominent in several areas of healthcare, such as diabetes, arthritis, cancer [LHZL20, BPSK17, AMMN16], with varying input formats ranging from tabular data in stored in relational databases to large scale image datasets [SZX⁺20]. Stemming from the involvement of data sensitivity in the medical domain is the necessity of gaining human trust towards ML application [TJMG19]. Thus, we see a recent surge in the production of interpretable results using state-of-the-art models such as Local Interpretable Model-Agnostic (LIME) [RSG16] and SHapley Additive exPlanations (SHAP) [LL17] to supplement the outputs provided by black-box algorithms, with much work showing the intent of XAI expansion through new prediction model architectures [dPM⁺21, MQS⁺20].

A few open-source applications have been created to ease the application of AI to datasets, e.g., [SACF⁺12, HFH⁺09, HEH21, Mei12]. Much data in biology is stored as images and Fiji [SACF⁺12] is an example of an open-source tool designed from biological-image analyses that aims to prototype algorithms for image-processing.

MOA [HEH21] is a free software that focuses on data streams and makes predictions on the fly. This platform offers a variety of AI algorithms for data stream analysis, as well as the ability to develop test models and apply them to input data. MOA may also visualise clusters and highlight outliers. The WEKA [HEH21] workbench contains multiple machine learning frameworks to support

¹<https://simulacrum.healthdatainsight.org.uk/>

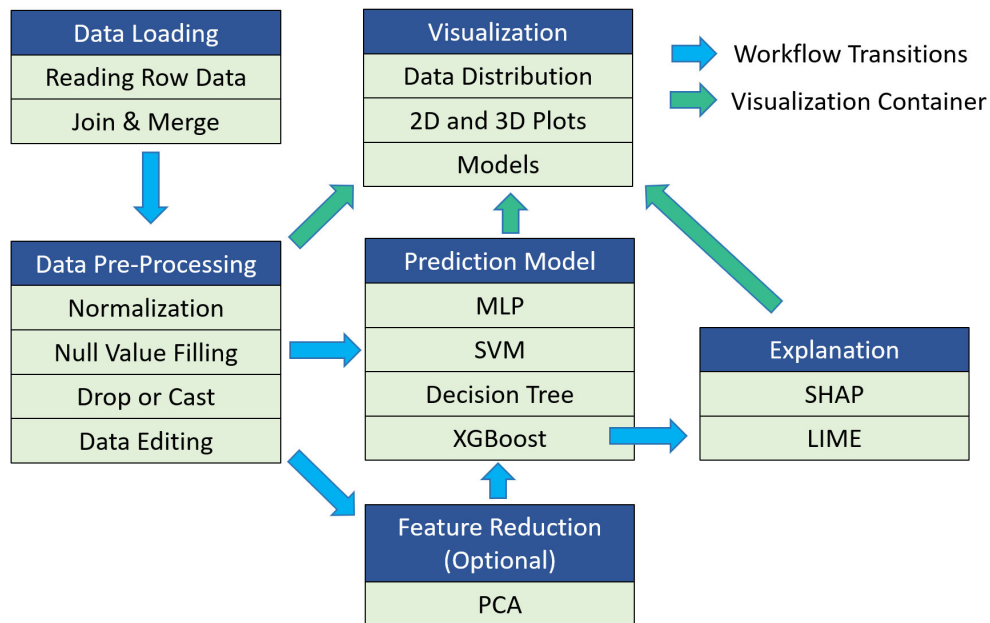


Figure 9.1: ExMed Workflow. ExMed provides the user with a sequence of simple actions, including loading, merging and editing data, and creating prediction as well as explanation models. Various visualisation techniques are supported in several stages of this pipeline.

various analyses via a gui. WEKA allows for quick access to the information included in the datasets, as well as the selection of specific areas of interest. WEKA, on the other hand, does not support both dataset merging and concatenation, which is frequently necessary when introducing new medical data. Lightside Researcher’s Benchmark [unk20] is another freely available tool, which combines machine learning and feature extraction within a graphical user interface. This software includes tools for preparing data, building and training a model, and doing data analytics. However, there is no visualisation component.

ExMed contrasts with these applications by including several data preparation tools for the majority of medical datasets. Another unique feature is the inclusion of XAI techniques combined with a wide variety of Visualization techniques.

9.3 ExMed Workflow

When sending medical data to an ML model, human errors and noise can reduce the quality of the results. Moreover, as one of the leading challenges in medical data analysis is to aggregate data from multiple data sources for performing joint analysis [DRA⁺20], it is crucial for medical data analytic platforms to support the pre-processing stage. Our new application ExMed addresses both challenges

and makes the integration of data pre-processing tools easier in order to minimise error and increase the baseline performance of the ML model.

ExMeds main functionalities, architecture and selected interface illustrations are shown in Fig. 9.1. ExMed implements a wide set of tools to load, process, predict, interpret and explain data. Its backend design is modular and designed to accept future extensions. Most common data files are accepted (e.g. Excel, CSV, or SAS, and XPT files). Input data can be combined through classic database join operators, whether or not a common key exists, to give users the potential to create larger datasets rapidly. Cells, rows, columns and data types can be edited by the user directly within ExMed, allowing greater freedom for data manipulation and quality checks. Data validation is supported by various visualisation tools included with the interface.

The ML models used include SVM, Random Forests, MLP Regression and XGBoost. Optional dimensionality reduction is done using Principal Component Analysis (PCA). Results of the PCA can be visualised in 2D or 3D from the two or three largest eigen vectors respectively. To interpret data, individual models have their own functions to offer specific explanations. SHAP dot plots, SHAP bar plots, SHAP dependence plots and LIME plots can be used for this purpose. LIME and SHAP adhere to ML local interpretability requirements for patient instances; expressed as a necessity from clinicians [TJMG19], whilst also producing global explanations. To invoke trust, we provide explanations from both LIME and SHAP as both models see a lack of ubiquity in feature priority, but may still provide valuable insight into the data as these methods still often see the same trend in feature attribution [DFB⁺21].

9.4 Case Study I: COVID-19 Control Measures

Here, we show how ExMed can be used to investigate relative effectiveness of COVID control measures used in the UK. From Public Health England ², we collect daily infection numbers reported across the nine regions of England as well as and the other three nations in the UK. Non-pharmaceutical control measure data were collected based on UK's COVID policies as summarised in Table 9.1. Data is collected from various sources including Wikipedia and major news agencies such as BBC. Control Measures are coded based on the level of severity ("High", "Moderate" or "Low") for all control measures excluding Non-essential shops and School closures, which are coded as binary choices ("Open" and "Closed"). Temperature and humidity data are obtained from the weather website Rospisaniye Pogodi Ltd³ were also included. A total of 4,257 data points that were collected between Feb. 2020 and Feb. 2021.

²<https://www.gov.uk/government/organisations/public-health-england>

³https://rp5.ru/Weather_in_the_world

Table 9.1: Non-pharmaceutical COVID Control Measures.

Control Measures	Type
Meeting Friends / Family (Indoor)	Categorical
Meeting Friends / Family (Outdoor)	Categorical
Domestic Travel Control	Categorical
International Travel Control	Categorical
Cafes and Restaurants Control	Categorical
Pubs and Bars Control	Categorical
Sports and Leisure Closure	Categorical
Hospitals / Care and Nursing Home Visits	Categorical
Non-Essential Shops Closure	Binary
School Closure	Binary

We study the effectiveness of control measures by observing their impacts to the virus transmission rate R_t . From daily infection numbers, we estimate R_t using the method reported in [FMG⁺20, WLB⁺20]. R_t is one of the most important quantities used to measure the epidemic spread. If $R_t > 1$, then the epidemic is expanding at time t , whereas if $R_t < 1$, then it is shrinking at time t . A *serial interval distribution*, which is a Gamma distribution $g(\tau)$ with mean 7 and standard deviation 4.5, is used to model the time between a person getting infected and he/she subsequently infecting another person on day τ . The number of new infections c_t on a day t is computed as:

$$c_t = R_t \sum_{\tau=0}^{t-1} c_\tau g_{t-\tau} \implies R_t = \frac{c_t}{\sum_{\tau=0}^{t-1} c_\tau g_{t-\tau}} \quad (9.1)$$

where c_τ is the number of new infections on day τ and g_s defined as:

$$g_1 = \int_{\tau=0}^{1.5} g(\tau) d\tau, \quad g_s = \int_{\tau=s-0.5}^{s+0.5} g(\tau) d\tau \quad \text{for } s \geq 2, \quad (9.2)$$

For $x = t$ and τ , c_x is the difference between the confirmed case on day x and the confirmed case on day $x - 1$, which is available from the dataset directly.

Using this data, we pose a simple classification question:

Given the infection number and control measures implemented on a day t , can we predict $R_t \geq 1$?

As control measures take time to affect the infection rate, we expand the dataset to include the duration of control measure implementations for all control measures. For example, “*Meeting Indoors (High) = 2*” means that “*it is the second week that meeting indoors has been banned completely*”. Similarly, “*International Travel (Low) = 0*” means that “*there is no restriction implemented on international*

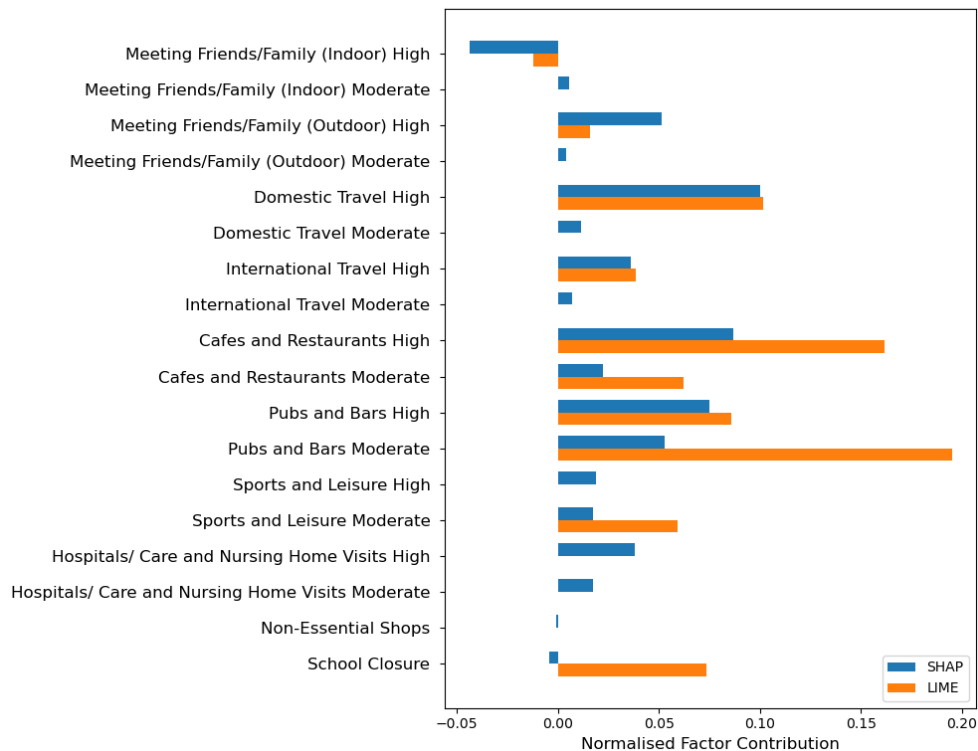


Figure 9.2: Example of an Explanation computed with SHAP and LIME. For this instance, both explainers consider top measures contributing to this prediction being *Domestic Travel*, *Cafes and Restaurants Closure* and *Pubs and Bars Closure*.

travel". We also drop instances before March 15, 2020 across all 12 regions and nations in our dataset due to the low number of infections.⁴ In this way, we form a data file with 18 features and 3,937 instances with 1,550 positive ones.

Table 9.2: Prediction performance on the COVID dataset with four different classifiers.

Classifier	MLP	Random Forest	SVM	XGBoost
Precision	0.87	0.90	0.87	0.87
Recall	0.79	0.84	0.78	0.79
F1-score	0.83	0.87	0.83	0.84

The classification results are summarised in Table 9.4. We observe that all four classifiers are able to achieve good performance on this dataset with a 70/30 training/testing split. As an illustration, for a prediction query instance such that:

⁴As can be seen from Equation 9.1, when c_x is small, R_t can flatten in a unrealistically large range and generate noises in the dataset.

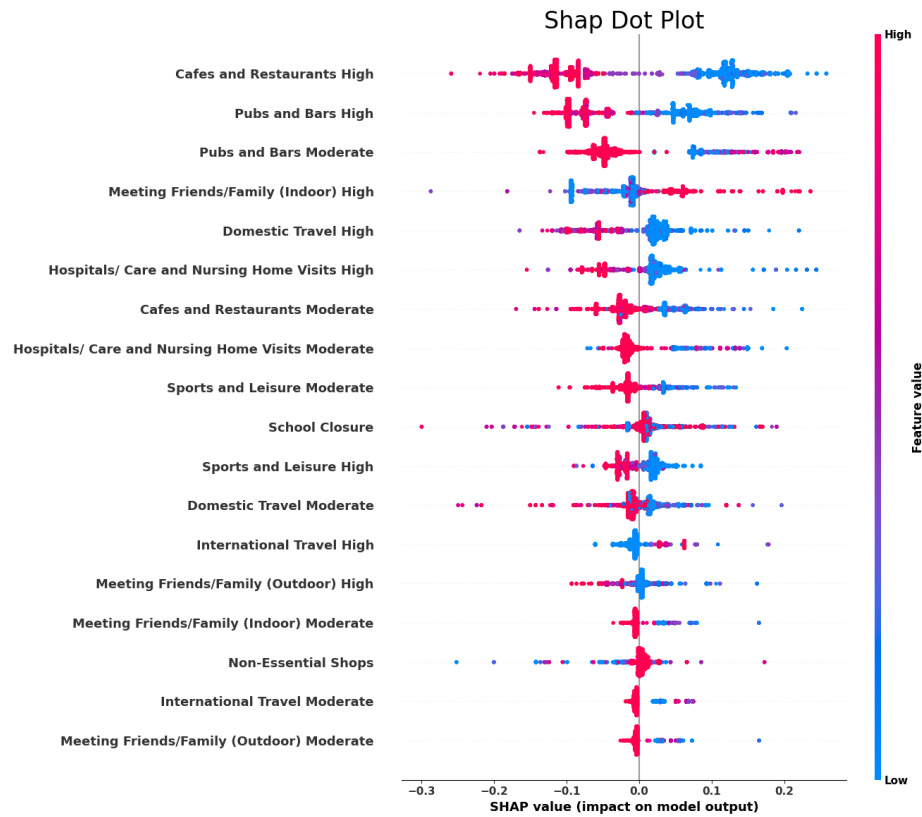


Figure 9.3: Global explanations generated using SHAP on our COVID dataset for the prediction whether $R_t \geq 1$. We see that closing down cafes and restaurants as well as pubs and bars are the most effective control measures. When their feature values are high (red), they have a strong negative impact to the prediction; whereas when their feature values are low (blue), they have strong positive impact to the prediction.

- All control measures (Table 9.1) except *International Travel (IT)* and *Hospital / Care and Nursing Home Visits (HCNHV)* are used for more than 35 days at the level *High*;
- *IT* has been implemented for more than 35 days at the level *Moderate*; and
- *HCNHV* implemented for 20-25 days at the level *High*.

Our Random Forest prediction model predicts correctly that $R_t < 1$, with SHAP and LIME (Fig. 9.2) producing similar explanations for the instance. In addition to local explanations, ExMed can also use SHAP to compute global explanations for the entire dataset - describing the “trend” of all instances - as illustrated in Fig. 9.3, where the most influential control measures for the predictions are *Cafes and Restaurants Control* and *Pubs and Bars Control*.

Table 9.3: Each patient is described with 20 features.

Feature	Value	Feature	Value
ACE	2.0	T Best	0.0
Sex	M	M Best	3.0
CNS	9.0	N Best	4.0
Age	68	Cycle Number	0.0
Grade	0.0	Ethnicity	1.0
Height	1.6	Cancer Plan	1.0
Weight	75.6	CReg Code	4.0
Morph	8041.0	Chemo Radiation	N
Laterality	901.0	Regimen Time Delay	N
Performance	1.0	Regimen Stopped Early	N

9.5 Case Study II: Lung Cancer Life Expectancy

Our second case study investigates the application of XAI to electronic patient records for cancer research instead of using public health epidemiology data in order to emphasise the transferability provided by ExMed. We use artificial data from the synthetic Simulacrum⁵ dataset that was developed by Health Data Insight CiC and derived from anonymous cancer data provided by the National Cancer Registration and Analysis Service⁶ (part of Public Health England). This dataset contains 1,322,100 cancer patient instances.

We first isolate a cohort of interest, opting for lung cancer patients as they represent a large portion of cancer-based deaths [SFS⁺21]. Therefore, we pose the following multi-class classification medical question:

Given a set of features for a patient, what will be the predicted survival time for the patient? Under six months, six to twelve months, or more than twelve months?

To study this, we first identified the subset of lung cancer patients in the dataset from the ICD-10 code “C34” *Malignant neoplasm of bronchus and lung* and a deceased status, which includes 108,282 patients in total. We removed records from the original dataset with obvious errors and included only patients with a vital status date posterior to the diagnosis date.

A major challenge in medical data analytics, as exemplified in the Simulacrum one, is missing or incomplete patient records. This results in a large number of “null” entries in the dataset. To address this, we identify a smaller cohort of patients such that each patient contains 20 features, with each patient instance

⁵<https://simulacrum.healthdatainsight.org.uk/>

⁶<http://www.ncin.org.uk/>

only able to contain a maximum of one “null” value. This explicit filtering isolates a balanced cohort of 2,260 patients.

Table 9.4: Predictions for the Lung Cancer dataset.

Classifier	MLP	Random Forest	SVM	XGBoost
Precision	0.86	0.90	0.77	0.69
Recall	0.76	0.90	0.98	0.66
F1-score	0.81	0.90	0.86	0.67

We first provide a local explanation example using both SHAP and LIME for a patient instance as shown in Table 9.3. We observe that both explainers give similar explanations as shown in Fig 9.4. Using the entire dataset, we produce a global explanation determining feature importance towards each output class in Fig 9.5. We then provide granularity to feature value importance towards a target class with Fig 9.6. We interpret these results as:

Cancer grades, BMI, age, patient performance and the absence of distant metastatic spread are key indicators for estimating patients survival time.

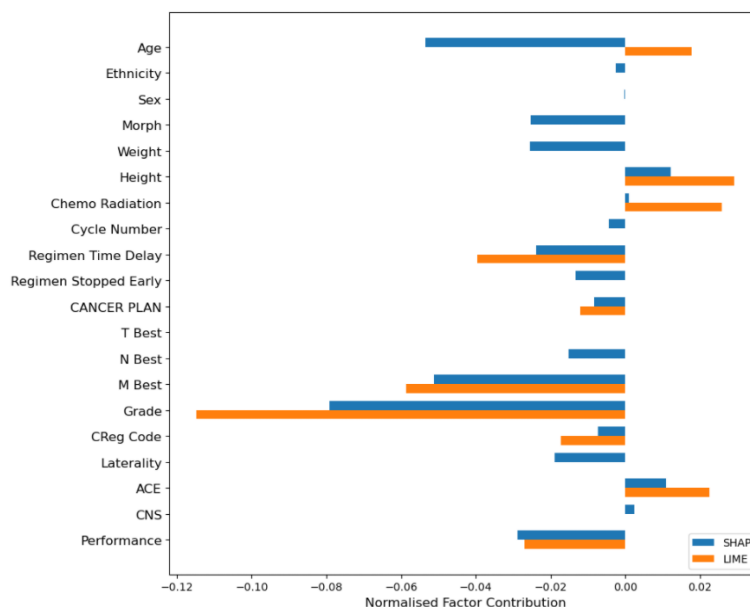


Figure 9.4: Local explanation on the Lung Cancer life expectancy data set for a patient instance. We see that the most impactful features amongst SHAP and LIME are ubiquitous: “Grade” *How the cancer cells act; the higher the grade the less normality the cell resembles and it may act more aggressive* and “M Best” *Presence or Absence of Distant Metastatic Spread*, followed by a disagreement on age attribution.

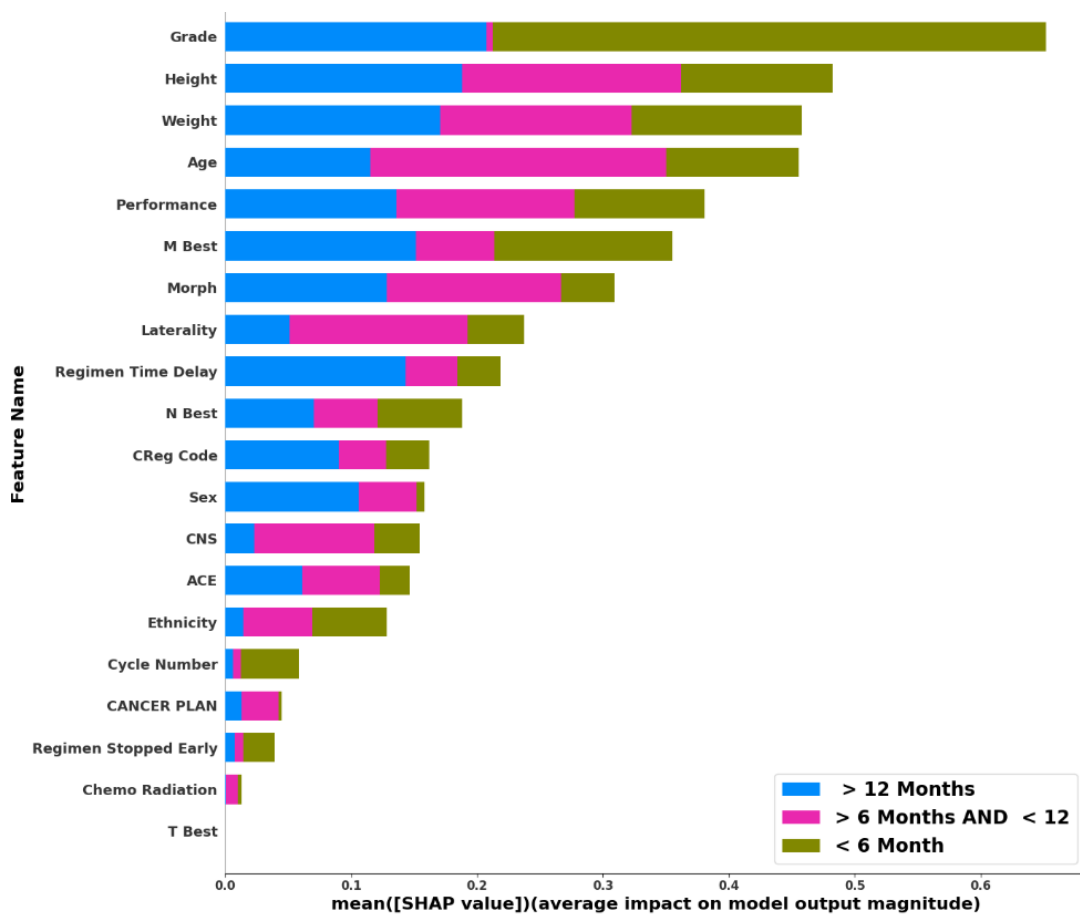


Figure 9.5: The largest impact towards the survival boundaries *greater than* 1 year and *less than* 6 months is the cancer grade. It has a direct impact on the longest and least time survival. Height, weight and patient age are also significant factors.

9.6 Conclusion

We have presented ExMed, a self-contained software package that enables Explainable AI data analysis for medical domain experts without the need for explicit programming. It can both concatenates the flexibility of medical sub-domain transferability and obtain an essence of trust through explainability using XAI methods. ExMed accepts multiple data input types and supports several standard pre-processing operations. It employs a number of different prediction models and visualisation techniques, while implementing two popular feature attribution XAI algorithms.

Through ExMed, we studied the effectiveness of COVID control measures in

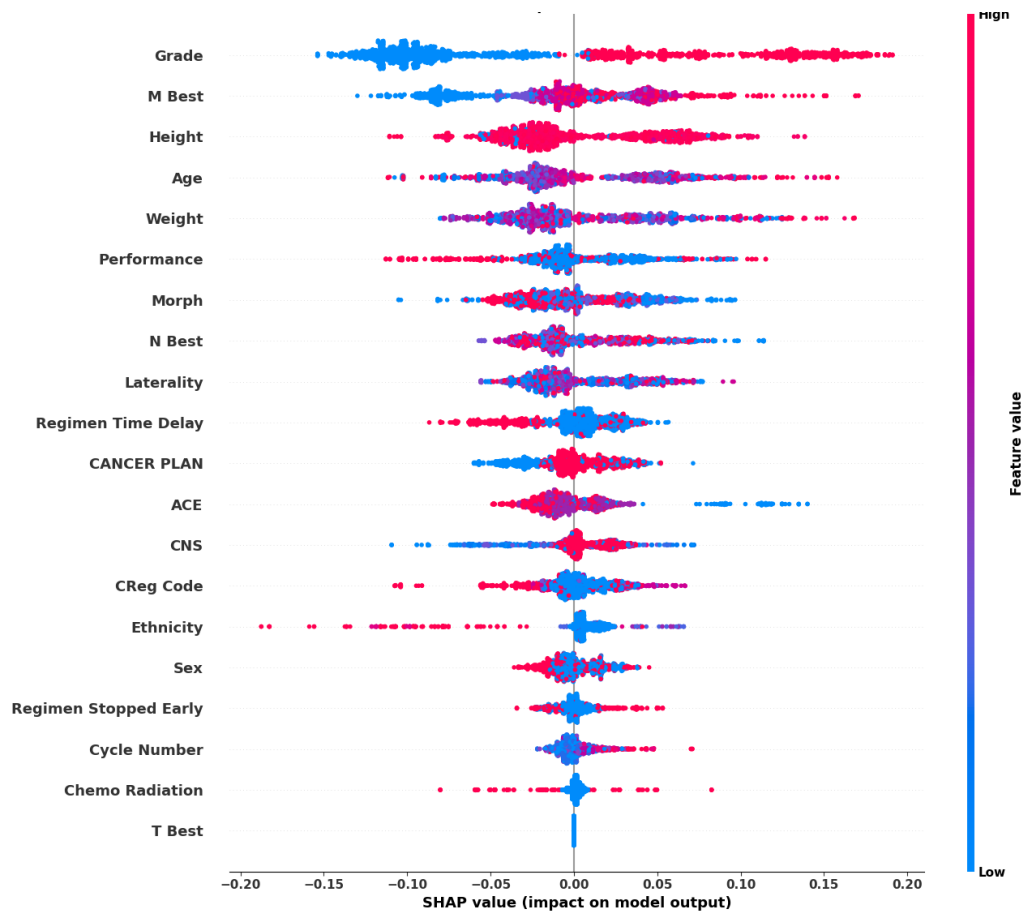


Figure 9.6: Global explanation measuring feature attribution against the class *Survival time of less than 6 months*, where we see the cancer grade of higher value - indicative of cell abnormality and aggressiveness, followed by “M Best” , “weight” and “height” determinants of body mass index (BMI) and “age”.

the UK using data from March 2020 to January 2021 and the life expectancy of lung cancer patients using the Simulacrum dataset. We observed that closing down cafes and restaurants as well as pubs and bars had the most impact in reducing the virus transmission rate. From the cancer case study, we saw that cancer grades, BMI, age and M Best variables are amongst the most influential factors for survival. In the future, we plan to (1) experiment ExMed with healthcare professionals and conduct user studies to evaluate effectiveness of various XAI approaches; (2) further expand the functionality of ExMed and explore features such as parameter tuning; (3) incorporate additional missing value imputation techniques such as MICE [BBGOR06] and SICE [KH20]; and (4) introducing additional XAI techniques such as Anchors[RSG18] in ExMed.

Part VII
Conclusion

Chapter 10

Summary

Contents

10.1 Conclusion	159
10.2 Future Work	161

10.1 Conclusion

This thesis provided an increase in transparency for black-box models, and sufficiently provided an increase in explainability from three perspectives: *Model*, *Data* and *User*. Thus emphasising that explainability can be approached from different directions. Here the:

- *Model* perspective provided many solutions to increase interpretability by either approximating the black-box model or utilise the infrastructure of the black-box. Here, adaptive local explanations, counterfactual explanations and irregular temporal explanations are explored and new methods PALE, CF-IG, Batch-IG and QUCE were introduced.
- *Data* perspective focused on building a clear relationship between imputation and explainability, drawing upon theoretical relationships and providing a new imputation method SSI that better adhere to the univariate distribution of feature values.
- *User* perspective introduces a tool: ExMed, that allows users to have an ease of access experimental tool to explore data, apply machine learning techniques and apply XAI techniques (LIME and SHAP specifically).

To briefly summarise the contributions of this thesis, I provided an early evaluation and identification of the disagreement problem for XAI methods in healthcare. Then I followed this with the introduction of a model-agnostic method

aiming to improve the accuracy of the LIME method and subsequently introduce an “adaptive” functionality to the explanations, where each patient has an optimised model to the best performing degree polynomial. The increase in accuracy in the model-agnostic LIME saw an increase in agreement with SHAP (TreeSHAP, thus not a model-agnostic method) on the patient sample, but the key contributions of this paper were instead the idea of local adaptive explanations and the increase in model-agnostic XAI accuracy.

Following this, I identified that whilst models such as LIME, SHAP and PALE provide explanations of attribution to a single instance, this does not conform to commonly used forms of evaluating effects. Thus it seems natural to utilise the causal effects of intervention, this often takes the form of counterfactual explanations in XAI (as opposed to A/B testing and evaluating the ATE). Here, I identified a clear gap where feature-attribution and a variety of generative counterfactual methods could be unified, and similarly evaluated and determined a set of desirable properties that hold with respect to the introduced method, namely Counterfactual Integrated-Gradients (CF-IG) which is a clear modification of the original IG method as to utilise the theoretic guarantees and expand the methods capabilities to an alternative XAI domain of counterfactuals. Similarly, from CF-IG there is a linear path constraint that is relaxed with the introduction of Quantified Uncertainty Counterfactual Explanations (QUCE), this approach enables for both the minimisation and the quantification of uncertainty along generated paths to provide generative counterfactual examples, where both the gradients and the generated instance are more reliable.

Extending upon this, with the knowledge of theoretic guarantees of IG, this method seemed natural to utilise and adapt further, and thus lead to another modification where I introduced “Batch Integrated-Gradients”, this method along with its formal introduction, aims to provide a new form of feature attribution for Electronic Health Records that are often presented temporally. Thus this chapter brings to light how certain changes of feature dimensions between time points, influence the change in prediction probability.

Exploring further, I identify a link between data imputation and explainability, this is heavily influenced by the *missingness* property identified in the SHAP properties. The implication of missing data would imply zero-value feature attribution and an altered prediction, this is consequential in domains such as healthcare where data is vital. Thus, it seemed necessary to bridge a gap between the fields of XAI and data imputation and to provide more accurate imputation methods is vital to the success of explainers. To this end I propose the Surrogate Set Imputer (SSI) method, that not only provides more accurate imputation, but also is inherently interpretable as it takes the form of a linear model with Shapley values that can be extracted directly, therefore promoting transparency of imputation and bridging an important relationship.

The accessibility of ML and XAI tends towards those with experience and exposure in the respective fields. To facilitate the usability for domain experts, we provide a simple tool (ExMed) for data exploration, visualisation and more critically the application ML and XAI for wider audiences. The implication of presenting such tool, is the greater ease of access with simple drop down menus and selection tools to apply complex methods in both ML and XAI. For example, such tool can aid in quicker patient predictions and population health analytics without the need to outsourcing to ML experts.

Therefore, to conclude; this thesis provides a novel comparative approach at the time of publication in the medical domain, followed by a set of newly introduced methods to improve explainability of medical records. These methods show how one can consider many aspects of explainability, ranging from model-agnostic (PALE), to the first (to my knowledge) exploration of path based methods for temporal explainability (Batch-IG), to counterfactuals instances and explanations generated through paths (Counterfactual-IG), and the minimisation of uncertainty over generated counterfactual paths (QUCE), also how one can indirectly increase explainability through the likes of imputation (SSI) and finally and how one can improve user-centric explainability through accessible tools (ExMed) for explainability, that do not rely on expertise in XAI, thus facilitating domain experts with a readily available tool to analyse data.

10.2 Future Work

There are many avenues to explore posterior to the completion of this thesis. To reflect on the body of this thesis, this section contains suggestions for future work, this is sequential to the occurrence of each part and respective chapter within the thesis.

10.2.1 Comparative Methods

The comparative methods introduced in this work provide little knowledge on the quality of explanations given by state-of-the-art XAI methods. Instead, this chapter focused on comparing the explanations returned by each method, thereby expanding the pool of XAI methods used in the comparison would provide further insight. For future work, considerations of independent XAI method performance will also be considered, evaluating the sensitivity and effects of hyper-parameters on methods, as well as other considerations such as the robustness of the explanations produced.

Thus, one could consider an evaluating of ϵ -Lipschitz continuity of explanations with respect to the input and a minor perturbation. Formally, given an instance \mathbf{x} , and a small perturbation $\Delta\mathbf{x} = \alpha \cdot \mathbf{x}$, for some $\alpha \in [0, 1]$. Then, given a

small perturbation to the input, an explainer Φ can be considered robust, if the explanation provided similarly only changes by a small amount. Therefore one could compute:

$$\epsilon = \frac{\|\Phi(\mathbf{x}) - \Phi(\mathbf{x} + \Delta\mathbf{x})\|}{\|\mathbf{x} - (\mathbf{x} + \Delta\mathbf{x})\|} \quad (10.1)$$

Following this, the explanations provided by different XAI methods can be further matched against domain knowledge through user-studies, or through the access to the ground truth explanations of real world data.

10.2.2 Enhancing Explainability a Model Perspective

Deeper insights of the PALE method can be extrapolated, this by exploring the property satisfiability of the method through theoretical analysis. Similarly, further research into adapting the neighbourhoods, that are generated with respect to the instance can be explored, to better define the perturbation space and improve the quality of explanation. The PALE framework could enhance its effectiveness by incorporating a technique to determine the optimal m degree polynomial fit, ensuring it balances between avoiding overfitting and achieving high accuracy within the local neighborhood. As the polynomial degrees increased, the PALE method demonstrated a substantial improvement in performance compared to the LIME method, suggesting a promising avenue for further experimentation and research.

To extrapolation further on the CF-IG method, a crucial aspect for future work involves considering counterfactual explanations as feasible tools for understanding "what-if" questions. It is imperative that we focus on modifiable features that are reasonable for a given patient. For instance, certain aspects, such as the dose of a drug, may not be modifiable at certain stages of a patient's treatment. In such cases, we need to carefully control the alteration of such features, similar to the idea of controllable and uncontrollable factors proposed in [KLS⁺22] to ensure the realism and practicality of counterfactual explanations. Similarly, the method only excels in continuous settings, further exploration is needed to adhere to categorical feature values, one could consider the step size being discrete for categorical values.

The Batch-IG method introduced in the body of this thesis encapsulates a line-integral formulation of explainability. For expansion of the Batch-IG approach, one can consider the path integral formulation that considers an larger number of paths, to instead consider a bounded number of paths that are *probable* for instance transitions between time steps. Here one approach would be to follow a similar notion to the QUCE extension for counterfactual explanations, such that explanations can be produced along the paths that aim to minimise

uncertainty. For QUCE we will aim to relax the assumption that all feature values are continuous to provide more realistic and reliable generated examples. Exploration of optimal parameters in the QUCE framework is currently a manual process. Automating this approach would provide greater flexibility and ease of application upon distributing the method to end users.

Following this, one can refer to the Expected Gradients and Integrated Hessian formulations introduced in section 2, these formulations can easily be translated to the CF-IG, Batch-IG and QUCE methods.

10.2.3 Enhancing Explainability a Data Perspective

The SSI method introduced in the body of this thesis are both limited to uni-variate feature imputation, such that only the uni-variate distribution is considered and thus all features are assumed to be independent. Naturally, this is the ideal case for imputation, but in many cases linear and non-linear correlations must be considered when providing imputation. Such considerations can also be utilised when approaching multiple value imputation. Further properties that unify the idea of imputation and explanations can also be proposed, to help aid the design of new methods. In fact, it will be a likely approach to use the learning mechanisms of the proposed QUCE method to achieve this.

One possible approach for data imputation involves evaluating 'what-if' scenarios, where different values are imputed for missing instances. This allows for the analysis of hypothetical scenarios, facilitating the drawing of inferences. For instance, considering an instance with a missing feature, evaluating it in various states enables us to understand the effects of introducing that feature for that specific instance. This exploration naturally leads to a set of counterfactuals, and the calculation of permutation feature attribution scores. An actionable set of plausible counterfactuals could be valuable for domain experts in feature imputation tasks, advocating for a more human-in-the-loop approach. Looking ahead, future research could delve deeper into the connection between imputation, counterfactuals, and feature attribution to bridge existing gaps and advance knowledge in this area.

For the extension of SSI in future work, we aim to enhance the usability of our approach by investigating two key aspects. First, we plan to explore methods that optimize the selection of k , the number of neighbors used to form the surrogate set, with the goal of improving accuracy of the imputation. Secondly, we intend to address the challenge of determining the best order for imputing features in instances with multiple missing values, with the aim of further refining our approach and making it more effective in handling complex data scenarios. These future developments are expected to contribute significantly to the practical applicability and robustness of our approach in real-world applications.

For the developed SSI library, greater accessibility is needed, which can be facilitated by creating working tutorials, installation guides, and functionality guidelines. These resources are planned for development upon acceptance of the initial manuscript. This will inevitably lead to another publication in a software-oriented journal and/or conference.

10.2.4 Enhancing Explainability a User Perspective

We introduced a tool called ExMed. Although this marks a promising start for an accessible tool that includes XAI methods, it is inherently limited by the selection of XAI methods that were selected during its development. At the time of development, LIME and SHAP were the most widely applied methods. Therefore, enhancing the tool’s utility and relevance by incorporating a wider range of introduced XAI methods would be advantageous. An initial step would be to integrate the proposed XAI methods from this thesis (PALE, CF-IG, Batch-IG, QUCE) for explanations. Additionally, including the proposed SSI method for data imputation would further augment the accessibility of the developed methods.

Extending upon this, another area to explore would be the involvement of the user in the explanation process. Consider explanations that, when evidently incorrect, the user could reinforce the explainer with a “more correct” explanation. To achieve this, one would need an explainer that learns continuously from new information.

10.2.5 Future Areas

Diversifying from the works explored in the chapters of this thesis, here further ideas are briefly explained to introduce a further body of focus. Diversifying from the previously introduced model of Batch-IG, allows for further opportunities in explanations catered towards temporal and time-series data. Hereinafter, the exploration of Neural Ordinary Different Equations [CRBD18] and Liquid Time-Constant Neural Networks [HLA⁺21] seem ideal avenues to explore, this primarily due to the continuous architecture. Given such architectures, the thought of how to extract explanations from such architecture (possibly real-time) would be a promising direction.

Throughout the body of this thesis there is an underlying assumption than one should be able to access a single ground truth. But it suffices to say that, this may not be optimal or even correct. Often when explaining different perspectives in real life, people may disagree with one another or shed light by providing different perspectives. It is not clear to me yet what an explanation should fully encapsulate, which angle should one approach an explanation and how should something be explained.

Should an explanation be given in such way a human would explain something?, or should a machine explanation be formed in a different way?. To me, the latter seems more reasonable if the trend of application specific AI continues, where the questions posed by a user can simply be “explained” through visualisation, quantified or by some other means e.g. given a prediction task, we wish to know what the model deems to be important, we can use feature attribution and see the result.

Aside from the focal point of this thesis given in the form of feature attribution, there are various works in other sub-fields that can be used to produce explanations such as argumentation theory [FT15, YPT23, PYT23] that follow in the prospects of human reasoning, thus it is clear there is not a conclusive way to produce explanations yet, nor any clear definition of what an explanation should entail, but indeed all are approaches are interesting.

Appendix A

Machine Learning

This chapter gives a brief overview of the Machine Learning methods used throughout the parts of this thesis. This does not provide a deep understanding of each method for the reader, instead provides a brief overview to supplement the rest of the thesis. The notation in this chapter does not follow convention and is independently defined here, as the background information and supporting notation is not reused throughout. Hereby, this chapter introduces: Linear Regression, Logistic Regression, k-Nearest Neighbour, eXtreme Gradient Boosting and Artificial Neural Networks. Following this, this chapter provides introductory formal explanations to further elements of Machine Learning (e.g. metrics, regularization techniques and loss functions) that are used throughout this thesis.

A.1 Linear Regression

First and foremost, we consider the linear regression method, linear regression can be represented with the simple equation,

$$y = mx + b$$

where, m is the slope, (x, y) are the variables and b is the y-intercept. Often, by evaluating m one can determine $\frac{dy}{dx}$, such that we obtain a interpretable solution for x .

Unfortunately, manually solving for β in the case of high-dimensional data becomes a task that cannot easily be computed by hand, as such we consider a Machine Learning approach for solving for $\langle \beta^1, \dots \beta^J \rangle$. For ease of representation, for the multivariate case, there exists a vector of coefficients $\hat{\beta}$ and matrix of instances X for a vector of outputs \mathbf{y} . Therefore, this is represented as

$$\mathbf{y} = \hat{\beta}X + \epsilon$$

where:

$$y = \beta^1 x^1 + \dots + \beta^J x^J + \epsilon_i$$

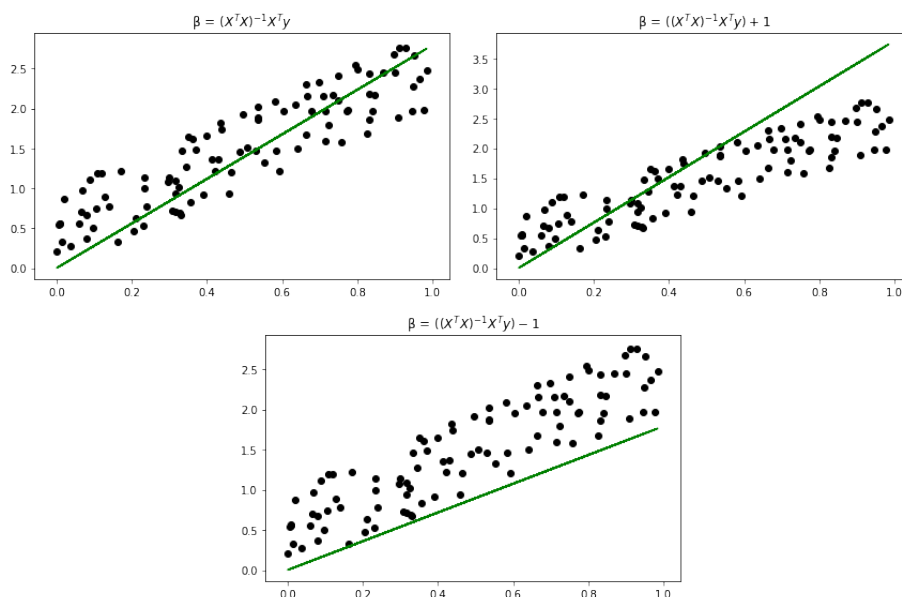


Figure A.4: Solving for β through the normal equation, with $\beta = (X^T X)^{-1} X^T y$ and $\beta \pm 1$. Here, it is shown that the best fit is given by the normal equation β .

One approach of solving for coefficients $\hat{\beta}$ is through the Ordinary Least Squares (OLS) and obtaining coefficient estimates from the normal equation.

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

Consequently, the linear approach assumes a linear relationship between independent features and the dependent feature. Therefore, although highly interpretable, the approach is good for simple problems with a continuous dependent variable, that is linearly associated with the independent variable(s). The effects of solving for $\hat{\beta} = \beta$ in the 1-Dimensions case, are shown in Figure A.4.

A.2 Logistic Regression

Extending upon linear regression towards a discrete setting, where a dependent variable $y \in \{0, 1\}$ follows a Bernoulli distribution is *logistic regression*. One can consider the linear equation:

$$y = \beta^1 x^1, \dots, \beta^J x^J.$$

To transform the linear equation to a discrete setting, one can utilise the sigmoid function $\rho : \mathbb{R} \rightarrow (0, 1)$, where:

$$\rho(y) = \frac{1}{1 + e^{-(\beta^1 x^1 + \dots + \beta^J x^J)}}.$$

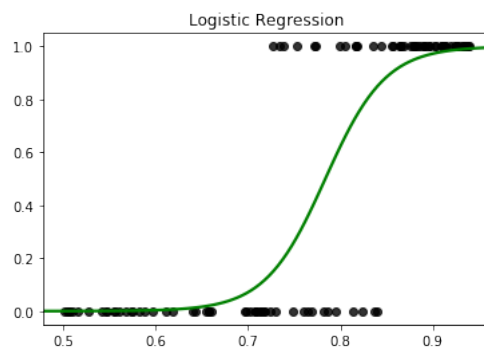


Figure A.5: Extrapolated from the linear regression example in Figure A.1, by applying the Sigmoid function to the solution of βX . The previous values of y in linear regression have been transformed such that, a value 1 is assigned if $y \geq \frac{1}{N} \sum_{y \in \mathbf{y}} y$ and 0 otherwise. Here, N is the number of samples in both X and y respectively.

Extrapolating from this, the log-odds can be obtained through applying the logit link function, which is the inverse of the sigmoid function ρ^{-1} . The respective odds ratios can be extracted by applying the exponential function to ρ^{-1} , yielding the equation:

$$\exp(\rho^{-1}(y)) = e^{(\beta^1 x^1 + \dots + \beta^J x^J)}$$

Therefore, although highly interpretable, the approach is good for simple problems with a discrete dependent variable, that is linearly associated with the independent variable(s).

A.3 k-Nearest Neighbour

The k-Nearest Neighbour (kNN) algorithm is a simple concept. Consider the Euclidean distance metrics (see section A.6.2). Here, the distance metric will be simply defined as the function $\delta(\cdot, \cdot)$ between two arbitrary vectors. Consider a labelled set of pairs of N pairs, when y is the label and \mathbf{x}' is the instance, such that: $\{\mathbf{x}', y\}^N$. Then, given a new instance \mathbf{x} with unknown label y , the closest k neighbours are found, by finding an instance that satisfies the following:

$$\arg \min_{\mathbf{x}'} \delta(\mathbf{x}, \mathbf{x}')$$

Here δ is an arbitrary distance metric, this is solved for k instances of \mathbf{x}' . Then given the set of k instances of \mathbf{x}' , the associated labels of the majority class are assigned to the new instance \mathbf{x} . This is illustrated in Figure A.9 for different values of k .

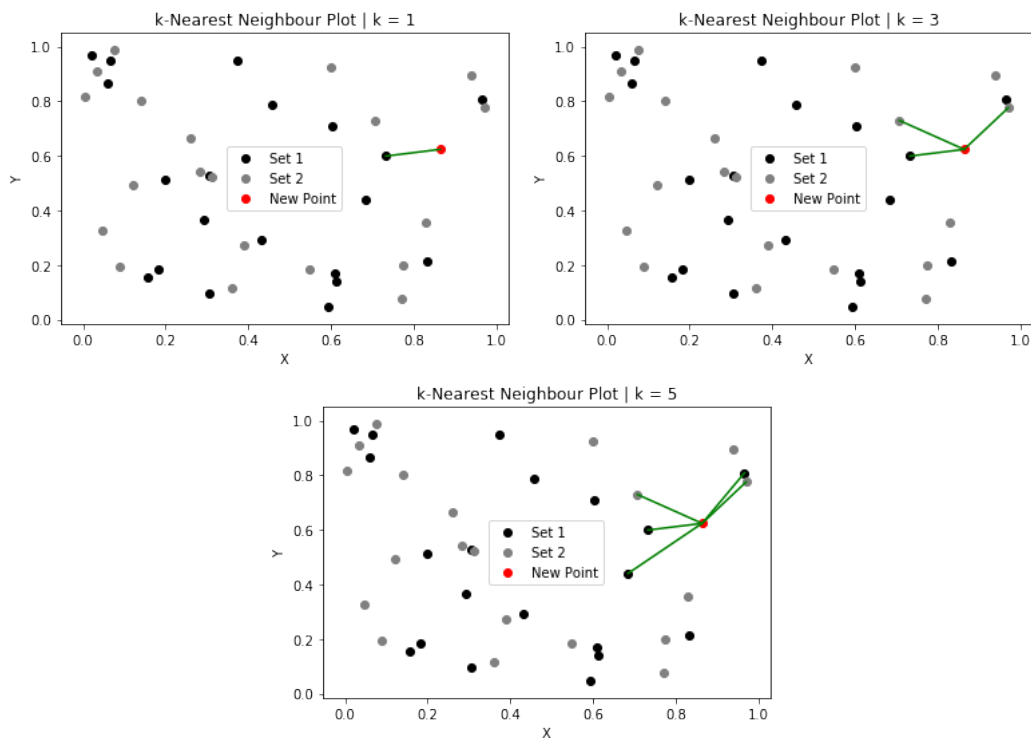


Figure A.9: Examples of the kNN algorithm applied to $k = \{1, 3, 5\}$. Here, consider two classes, one containing black points, the other containing grey. A new data point in red is added, and then assigned to the appropriate class. Here, when $k = 1$ the red point is assigned to the black class. When $k = 3$, the red point is assigned to the grey class. When $k = 5$, the red point is assigned to the black class.

A.4 eXtreme Gradient Boosting

The eXtreme Gradient Boosting (XGBoost) [CG16] method is a tree boosting method, that is an ensemble of trees. Gradient boosting tree ensemble methods aim to optimise the following equation:

$$\mathcal{L}^{(t)} = l(\mathbf{y}, \hat{\mathbf{y}}^{(t-1)} + f_t(\mathbf{x})) + \Omega(f_t)$$

Here, l is a convex differentiable loss function, \mathbf{y} is the true value and $\hat{\mathbf{y}}$ is the predicted value. The value of t represents the t^{th} iteration. The best performing model of f_t is added greedily, where Ω penalises the complexity of the model.

A.5 Artificial Neural Networks

The ANN architecture can take many forms in ML research: Recurrent Neural Networks [RHW86], Convolutional Neural Networks [ON15], Liquid Neural Networks

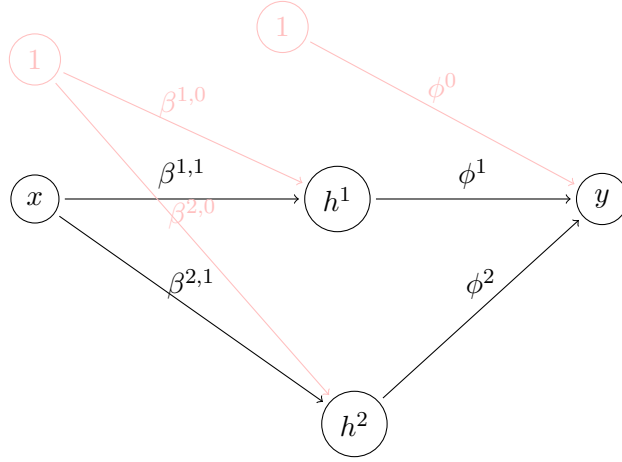


Figure A.10: Demonstrative diagram for the example Neural Network given in section A.5.

[HLA⁺21], Autoencoder [Bal12], to standard Artificial Neural Networks [Ros58]. For the scope of this thesis, the standard neural network architecture is given. The construct of a simple neural network follows a collection of linear functions with an activation function λ . For an example of activations, one can refer to the Sigmoid activation of the linear equation given in section A.2. Defining a simple neural network with 7 parameters $\hat{\beta} = \langle \phi^0, \phi^1, \phi^2, \beta^{1,0}, \beta^{1,1}, \beta^{2,0}, \beta^{2,1} \rangle$, then the network can be represented as:

$$f(x, \hat{\beta}) = \phi^0 + \phi^1 \lambda[\beta^{1,0} + \beta^{1,1}x] + \phi^2 \lambda[\beta^{2,0} + \beta^{2,1}x]$$

To simplify the equation, one can combine activation of the linear equations, into a single representation of the hidden units, such that:

$$\begin{aligned} h^1 &= \lambda[\beta^{1,0} + \beta^{1,1}x] \\ h^2 &= \lambda[\beta^{2,0} + \beta^{2,1}x] \end{aligned}$$

Therefore, a simple neural network can be represented as:

$$y = f(x, \hat{\beta}) = \phi^0 + \phi^1 h^1 + \phi^2 h^2$$

A diagrammatic representation is given figure A.10.

It is clear one can produce a piece-wise linear function, this associative to the *universal approximation theory*. This informally states; giving the correct set of weights to a feed-forward neural network can approximate any function [Pin99]. To produce a good fit for the neural network, the parameters of the model need to be tuned. Therefore, back-propagation is fundamental to learning

within neural networks. Whilst, there is the introduction of more recent learning algorithms such as the Forward-Forward Algorithm [Hin22], back-propagation has been fundamental for all forms of neural networks. The parametric update process in back-propagation is known as gradient descent (and corresponding variants). Details of gradient descent are seen in section A.6.3.

A.6 Supplementary Mathematics for ML

This section serves as a further formal introduction to foundational concepts that are used within ML. Therefore, this section is organised as follows: Section A.6.1 provides a formal intuition into the structure of loss functions for ML methods. Section A.6.2 provides a formal background of measurements that are used in ML with different application purposes. Section A.6.3 provides simple mathematical intuition into the gradient descent algorithm used for training ML models.

A.6.1 Loss Functions

By convention, in Machine Learning it is common to setup an optimization problem, as to find the minimum of a function. The natural formulations for Machine Learning problem often fall under one of the three categories in supervised learning: *Regression*, *Binary Classification* and *Multi-class Classification*.

Machine Learning problems, can be framed as finding the conditional probability of an output y , given an input \mathbf{x} . As to obtain an accurate prediction of y , one can formulate the idea of **maximising** the **likelihood** of finding the output $y \in \mathbf{y}$, given an input \mathbf{x} . The conditional probability can be formulated as:

$$\Pr(y|\mathbf{x}).$$

Constructing a loss function \mathcal{L} aims to maximise the probability that y does in fact belong to \mathbf{x} . Whilst, posing this question, considerations as to how to maximise the probability given a set of inputs $\mathbf{x} \in X$ and outputs \mathbf{y} arise.

Machine Learning of takes an approach where a set of parameters θ are to be tuned, for this we refer to the formulation of linear regression and logistic regression in section A.1 and A.2 respectively. Each each case, we have a set of coefficients $\hat{\beta}$. Therefore, given a function that takes an instance and set of parameters $f(\mathbf{x}, \hat{\beta})$, one can set up a parametric distribution in the form:

$$\Pr(y|\theta),$$

where the network $f(\mathbf{x}, \hat{\beta})$ aims to find θ , where θ shapes the distribution of \mathbf{x} .

Therefore maximum likelihood criterion is defined as:

$$\begin{aligned}\hat{\beta} &= \arg \max_{\hat{\beta}} \left[\prod_{\mathbf{x} \in X, y \in \mathbf{y}} \Pr(y|\mathbf{x}) \right] \\ &= \arg \max_{\hat{\beta}} \left[\prod_{\mathbf{x} \in X, y \in \mathbf{y}} \Pr(y|\theta) \right] \\ &= \arg \max_{\hat{\beta}} \left[\prod_{\mathbf{x} \in X, y \in \mathbf{y}} \Pr(y|f(\mathbf{x}, \hat{\beta})) \right].\end{aligned}$$

This, indicates an assumption of independence, which is later given in Chapter 2. By convention, the products are better represented in log-form, thus the maximum log likelihood is often considered, recalling that:

$$\log_a(cd) = \log_a(c) + \log_a(d)$$

one can arrive at the following summation to compute the maximum log-likelihood:

$$\hat{\beta} = \arg \max_{\hat{\beta}} \left[\sum_{\mathbf{x} \in X, y \in \mathbf{y}} \log[\Pr(y|f(\mathbf{x}, \hat{\beta}))] \right].$$

As previously stated, Machine Learning algorithms often construct a loss \mathcal{L} , as an minimisation problem. Therefore, one can instead consider minimising the negative log-likelihood:

$$\hat{\beta} = \arg \min_{\hat{\beta}} \left[- \sum_{\mathbf{x} \in X, y \in \mathbf{y}} \log[\Pr(y|f(\mathbf{x}, \hat{\beta}))] \right],$$

formulating our loss function \mathcal{L} , parameterised by $\hat{\beta}$, which can be rewritten as:

$$\hat{\beta} = \arg \min_{\hat{\beta}} \left[\mathcal{L}[\hat{\beta}] \right].$$

As to satisfy the criterion of the loss function \mathcal{L} , the formulation of the loss function as to satisfy the problem type are given as follows:

- **Regression:** takes the form, to solve the least squares loss function:

$$\mathcal{L} = \sum_{\mathbf{x} \in X, y \in \mathbf{y}} (y - f(\mathbf{x}, \hat{\beta}))^2$$

this is an extrapolation of solving for $\hat{\beta}$. For example, in the case of the univariate Gaussian distribution assumption, all terms that do not depend on $\hat{\beta}$ are mitigated:

$$\hat{\beta} = \arg \min_{\hat{\beta}} \left[- \sum_{\mathbf{x} \in X, y \in \mathbf{y}} \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} - \frac{(y - f(\mathbf{x}, \hat{\beta}))^2}{2\sigma^2} \right] \right]$$

here, σ^2 represents the variance of the Gaussian distribution (see Appendix B for the parametric effects on the Gaussian Distribution).

- **Binary Classification:** aims to solve the binary cross-entropy function for the training data, given by:

$$\mathcal{L} = \sum_{\mathbf{x} \in X, y \in \mathcal{Y}} -(1 - y) \log \left[1 - \text{sig}[f(\mathbf{x}, \hat{\beta})] \right] - y \log \left[\text{sig}[f(\mathbf{x}, \hat{\beta})] \right]$$

intuitively, the sigmoid function is used to predict the Bernoulli distribution parameter κ for a discrete $y \in \{0, 1\}$. Thus, when given a probability threshold to assign a prediction to a class τ (often $\tau = 0.5$), class assignment is given, such that:

$$y = \begin{cases} 1, & \text{if } \kappa > \tau \\ 0, & \text{otherwise.} \end{cases}$$

- **Multi-class Classification:** takes the form, to solve the cross-entropy loss function is defined as:

$$\mathcal{L} = - \sum_{\mathbf{x} \in X, y \in \mathcal{Y}} \log \left[\text{softmax}_y \left[f(\mathbf{x}, \hat{\beta}) \right] \right]$$

where:

$$\text{softmax}_c [\mathbf{z}] = \frac{\exp[\mathbf{z}_c]}{\sum_{c \in \mathbf{c}} \exp[\mathbf{z}_c]}$$

here, \mathbf{z} represents an arbitrary input vector and \mathbf{c} contains each possible output, and $c \in \mathbf{c}$ represents a single output.

Note that, the term cross-entropy is used in this section. Cross-entropy is functionally equivalent to minimising the negative log-likelihood. Informally, the cross-entropy loss can be thought of as, the difference in the distribution between the empirical data and observed data. Thus, to minimise the loss (maximise the likelihood), one would need to better map the observed data to follow the distribution of the empirical.

Kullback-Leibler (KL) divergence is a measurement of divergence between two probability distributions. Consider a true distribution $p(\cdot)$, and approximation of

the true distribution $q(\cdot)$. KL divergence is defined as:

$$\begin{aligned} KL(p||q) &= \int_{-\infty}^{\infty} p(y) \log \left[\frac{p(y)}{q(y)} \right] dy - \int_{-\infty}^{\infty} q(y) \log \left[\frac{q(y)}{p(y)} \right] dy \\ &= \int_{-\infty}^{\infty} p(y) \log \left[\frac{p(y)}{q(y)} \right] dy \\ &= \mathbb{E}_{y \sim p(y)} \left[\log \left[\frac{p(y)}{q(y)} \right] \right] \end{aligned}$$

Therefore, if we consider an estimation of the distribution in the ML setting parameterised by $\hat{\beta}$, and given the observed θ , this can be rewritten as:

$$\begin{aligned} KL(\Pr(y|\theta) || \Pr(y|\hat{\beta})) &= \mathbb{E}_{y \sim \Pr(y|\theta)} \left[\log \left[\frac{\Pr(y|\theta)}{\Pr(y|\hat{\beta})} \right] \right] \\ &= \mathbb{E}_{y \sim \Pr(y|\theta)} \left[\log \left[\Pr(y|\theta) \right] - \log \left[\Pr(y|\hat{\beta}) \right] \right] \end{aligned}$$

as to minimise the divergence, with respect to $\hat{\beta}$, consider the ML function f and data \mathbf{x} , the equation can be reduced to terms only including $\hat{\beta}$. Giving:

$$\begin{aligned} \hat{\beta} &= \arg \min_{\hat{\beta}} \mathbb{E}_{y \sim \Pr(y|\theta)} \left[-\log \left[\Pr(y|f(\mathbf{x}, \hat{\beta})) \right] \right] \\ &= \arg \min_{\hat{\beta}} \left[-\frac{1}{N} \sum_{\mathbf{x} \in X, y \in \mathbf{y}} \log \left[\Pr(y|f(\mathbf{x}, \hat{\beta})) \right] \right] \end{aligned}$$

The minima can be found regardless of the scaling term $\frac{1}{N}$, the equation can be reduced to the form:

$$\hat{\beta} = \arg \min_{\hat{\beta}} \left[- \sum_{\mathbf{x} \in X, y \in \mathbf{y}} \log \left[\Pr(y|f(\mathbf{x}, \hat{\beta})) \right] \right]$$

Thus, functionally equivalent to minimising the negative log-likelihood. Further details on the background of ML that supplement the understanding presented in this thesis can be found in [GBC16, Pri23].

A.6.2 Distance Metrics, Norms & Regularization

Consider a vector $\mathbf{x} = \langle x^1, \dots, x^J \rangle$, $0 < j \leq J$, we provide descriptions of each norm:

- l_0 norm = $||\mathbf{x}||_0$. The l_0 norm represents the cardinality of non-zero elements of a vector. More formally, let:

$$||\mathbf{x}||_0 = \sum_{j=1}^J \mathbb{1}_{[x^j \neq 0]}.$$

- l_1 norm = $\|\mathbf{x}\|_1$. The l_1 norm is given by the sum of absolute values of a vector, thus represented as:

$$\|\mathbf{x}\|_1 = \sum_{j \in J} |x^j|.$$

- l_2 norm = $\|\mathbf{x}\|_2$. The l_2 norm of a vector, is also known as the Euclidean norm, here:

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{j \in J} |x^j|^2}.$$

This section briefly covers commonly used regularization techniques for Machine Learning. The common forms of regularization are l_1 and l_2 regularization. Therefore, consider a regularization function applied to the coefficients $\hat{\beta}$, represented $\Omega(\hat{\beta})$. The regularization parameter extends a loss function \mathcal{L} by constraining weights. For example, consider the least squares equation:

$$\mathcal{L} = \sum_{\mathbf{x} \in X, y \in Y} (y - f(\mathbf{x}, \hat{\beta}))^2$$

The regularization function penalises the weights. Therefore, the loss function can be rewritten as:

$$\mathcal{L} = \sum_{\mathbf{x} \in X, y \in Y} (y - f(\mathbf{x}, \hat{\beta}))^2 + \lambda \Omega(\hat{\beta})$$

Where λ is a weighting parameter. Introducing the l_1 and l_2 regularization methods, they are given as:

- **l_1 Regularization:** is given as the sum of absolute values of $\beta \in \hat{\beta}$. Thereby, shrinking obsolete coefficients to zero. Therefore:

$$\Omega(\hat{\beta}) = \|\hat{\beta}\|_1 = \sum_{\beta \in \hat{\beta}} |\beta|$$

- **l_2 Regularization** is given as a the sum of squared coefficients. Thereby, shrinking coefficients evenly. Therefore:

$$\Omega(\hat{\beta}) = \|\hat{\beta}\|_2 = \sum_{\beta \in \hat{\beta}} \beta^2$$

A.6.3 Gradient Descent

Given a formal introduction to Machine Learning methods in this chapter, one way of solving for the optimal parameters $\hat{\beta}$ is through the gradient descent method. As the scope of this thesis resides less around optimisation techniques for learners, this section is limited to only introducing the gradient descent algorithm.

Gradient descent is an algorithm used in ML, aiming to minimise a (often) loss function \mathcal{L} . Informally, gradient descent aims to find the optimal value for parametric terms in a neural network (weights and biases). Thus, consider a set of weights $\hat{\beta} = \langle \beta^1, \dots, \beta^J \rangle$. Using the effect of the weights on the loss, which can be calculated via the partial derivatives of \mathcal{L} with respect to $\hat{\beta}$:

$$\nabla \mathcal{L} = \left\langle \frac{\partial \mathcal{L}}{\partial \beta^1}, \dots, \frac{\partial \mathcal{L}}{\partial \beta^J} \right\rangle$$

Ideally, through the gradient descent process, one will find $\nabla \mathcal{L}(\hat{\beta}) \approx 0$. Intuitively, if the loss is non-zero, the function \mathcal{L} decreases in the direction of the negative gradient. Thus, $\nabla \mathcal{L}(\hat{\beta}) \geq \nabla \mathcal{L}(\hat{\beta}_{new})$. Here, $\hat{\beta}_{new}$ is given by:

$$\hat{\beta}_{new} = \hat{\beta} - \alpha(\nabla \mathcal{L}(\hat{\beta}))$$

where α is a learning rate.

Example A.1 Consider a simple function to minimise f , such that:

$$f(x) = 3x^2 + 2x + 1.$$

Here, for practicality only terms including x are considered, as constants not including x will be mitigated in the differentiation (with respect to x) process, thus ignoring the term, it is easy to see that:

$$\begin{aligned} f(x + \Delta x) &= 3(x + \Delta x)^2 + 2(x + \Delta x) \\ &= 3(x + \Delta x)(x + \Delta x) + 2(x + \Delta x) \\ &= 3(x^2 + 2x\Delta x + \Delta x^2) + 2x + 2\Delta x \\ &= 3x^2 + 6x\Delta x + 3\Delta x^2 + 2x + 2\Delta x \end{aligned}$$

Thus, recall derivations from first principles, where:

$$\frac{df}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

then:

$$\begin{aligned}\frac{df}{dx} &= \lim_{\Delta x \rightarrow 0} \frac{(3x^2 + 6x\Delta x + 3\Delta x^2 + 2x + 2\Delta x) - (3x^2 + 2x)}{\Delta x} \\ &\implies \lim_{\Delta x \rightarrow 0} \frac{6x\Delta x + 3\Delta x^2 + 2\Delta x}{\Delta x} \\ &\implies \lim_{\Delta x \rightarrow 0} \frac{\Delta x(6x + 3\Delta x + 2)}{\Delta x} \\ &\implies \lim_{\Delta x \rightarrow 0} 6x + 3\Delta x + 2 \\ &= 6x + 2\end{aligned}$$

Therefore,

$$\frac{df}{dx} = 6x + 2.$$

Initialising x with the value 5 gives:

$$f(5) = 3 * 5^2 + 2 * 5 + 1 = 86$$

Therefore, it is necessary to update the value of 5. Thus, to find a value that satisfies the function minima, where the gradient tends towards 0, x is updated via:

$$x_{new} = x - \alpha(6x + 2).$$

Let the learning rate $\alpha = 0.1$, over 10 iterations as illustrated by the example in Figure A.11. Thus, numerically working through each iteration in the given example, we have:

$$\begin{aligned}x_{new} &= 5 \\ &= 5 - 0.1(6 * 5 + 2) = 1.8 \\ &= 1.8 - 0.1(6 * 1.8 + 2) = 0.52 \\ &= 0.52 - 0.1(6 * 0.52 + 2) = 0.008 \\ &= 0.008 - 0.1(6 * 0.008 + 2) = -0.1968 \\ &= -0.1968 - 0.1(6 * -0.1968 + 2) = -0.27872 \\ &= -0.27872 - 0.1(6 * -0.27872 + 2) = -0.311488 \\ &= -0.311488 - 0.1(6 * -0.311488 + 2) = -0.3245952 \\ &= -0.3245952 - 0.1(6 * -0.3245952 + 2) = -0.32983808 \\ &= -0.32983808 - 0.1(6 * -0.32983808 + 2) = -0.331935232 \\ &= -0.331935232 - 0.1(6 * -0.331935232 + 2) = -0.3327740928 \\ x_{new} &= -0.3327740928\end{aligned}$$

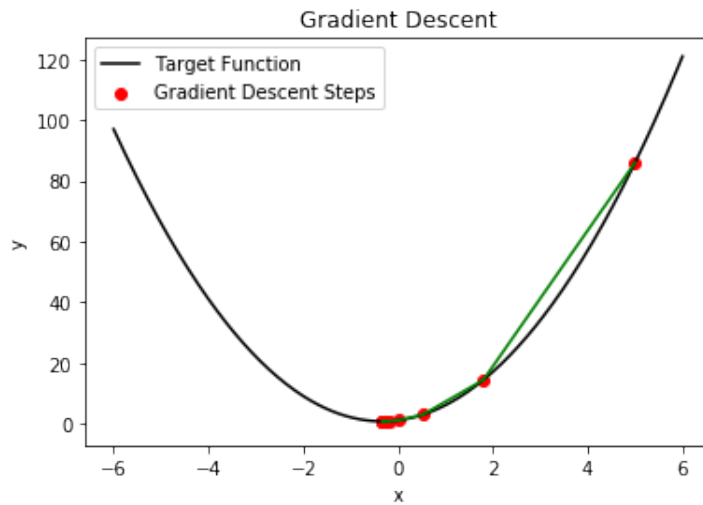


Figure A.11: Simple example of gradient descent on a one dimensional value. Here, our value is initialised at 5. x is then updated through the gradient descent process, with a learning rate of $\alpha = 0.1$ and 10 step iteration process. Here, each red point illustrates a step location of the function with respect to each updated x . Here, we see convergence from example A.1 to the value $f(x) \approx 0.0022$

Then, plugging the new value x_{new} into the original function f :

$$f(-0.3327740928) = 3 * -0.3327740928^2 + 2 * -0.3327740928 + 1 \approx 0.0022$$

Thus, concluding a simple arithmetic example of gradient descent. ■

Of course, this idea can fundamentally be extrapolated to J dimensions. In figure A.12 another simple example is found in 3 dimensions, with a 2 dimensional input vector.

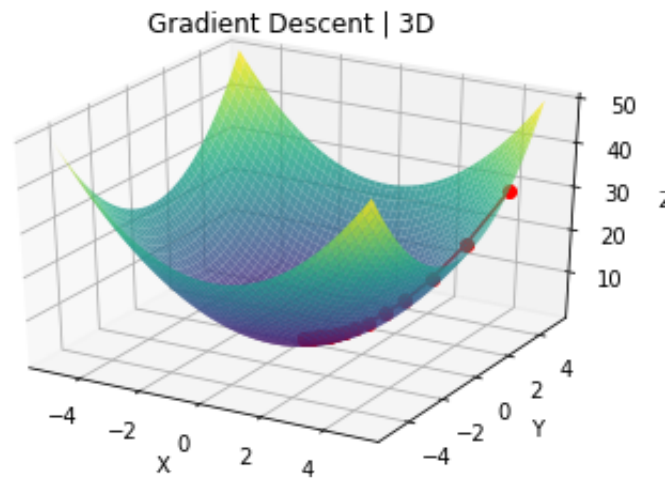


Figure A.12: Simple example of gradient descent on a two dimensional input. Here, our inputs are initialised at $x = 5$ and $y = 3$, where $z = f(x, y)$. x and y are then updated through the gradient descent process, with a learning rate of $\alpha = 0.1$ and 20 step iteration process. Here, each red point illustrates a step location of the function with respect to each update.

Appendix B

Parametric effects on the Gaussian Distribution

Reconsider the following solution for $\hat{\beta}$:

$$\hat{\beta} = \arg \min_{\hat{\beta}} \left[- \sum_{\mathbf{x} \in X, y \in Y} \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} - \frac{(y - f(\mathbf{x}, \hat{\beta}))^2}{2\sigma^2} \right] \right] \quad (\text{B.1})$$

This is derived as a solution given the probability density function (PDF), for a Gaussian distribution, which is given by:

$$\Pr(y|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[- \frac{(y - \mu)^2}{2\sigma^2} \right]$$

here, σ^2 is the variance and μ is the mean of the distribution. Thus, by plugging this into the maximum negative log-likelihood equation, we arrive at equation B.1. In this section, we provide a set of plots in Figure B.3, showing the effects of μ and σ^2 on the distribution:

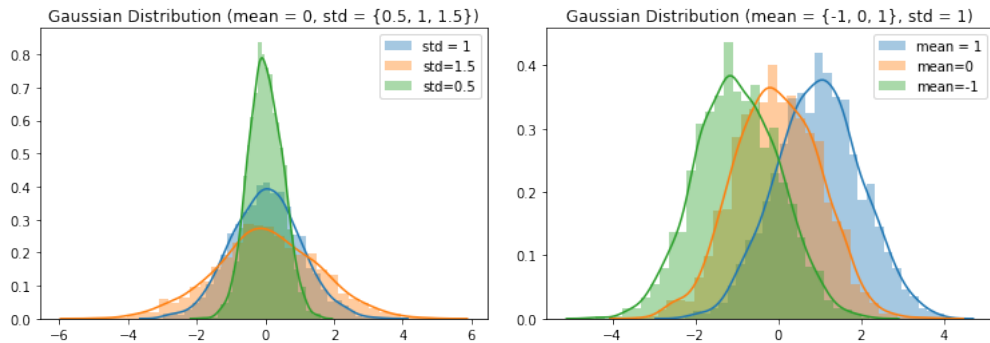


Figure B.3: Collection of plots illustrating the effects of μ and σ^2 on the distribution.

Appendix C

Simulacrum Data

Table C.1: The simulacrum feature names with given description

Feature Name Transformed	Feature Description
Age	Age of the patient at the time of instance
Sex	Sex of the patient
Weight	Weight in kg. at the start of the SACT regimen
Height	Height in metres at the start of the SACT regimen
Morph	Morphology ICD-10 code
Grade	Indicative of Tumor grade
CANCER PLAN	The cancer plan
Dose Administration	Dose in mg for each administration in the SACT cycle
Chemo Radiation	Whether or not a patient is undergoing radiotherapy
Behaviour	Behaviour description
Outcome	A description of regimen outcome
STATUS	Whether the patient is Dead or Alive
T Best	The size and extent of the tumor
N Best	No. of nearby lymph nodes that have cancer
M Best	Whether the cancer has spread from the primary tumor
Drug Group	The drug group name
Administration route	Method of delivery for each cycle administration
Time Delay	Time delay between administration
Stopped Early	Regimen stopped early
Regimen	The mapped regimen name
Clinical Trial	Patient is in an active Systemic Anti Cancer therapy trial
Cycle	The current cycle
CNS	Clinical Nurse Specialist
Reduction	Identifies the reduction of the drug dose during regimen
ACE	Adult Co-morbidity Evaluation 27

Appendix D

Conceptualising Batch-Integrated Gradients for Temporal EHR Explanations

D.1 Introduction

Due to modern computation capabilities, black-box models in recent Artificial Intelligence (AI) literature often take the form of Deep Neural Networks (DNNs), the complexity of such methods enable an increase in accuracy. At the same time, the increase in accuracy is usually associated with a decrease in model interpretability. Explainable Artificial Intelligence (XAI) is a common approach for increasing the transparency of black-box AI methods, XAI encompasses an increase in importance as desire to use accurate predictive models is inherited within high-risk domains. Feature-attribution is a commonly used method for XAI, where the aim is to determine how each feature influences the prediction of an instance, the landmark papers for this are introduced by the authors in [LL17] and [RSG16].

Extrapolating from Chapters 4 and 5, where the focus was on both counterfactual model-specific explanations and model-agnostic explanations for approximating the decision boundary, where the intent is to explain a single instance prediction, instead the temporal dynamic is explored, as despite recent success in XAI methods in developing feature attribution explainers such as SHapley Additive exPlanations (SHAP) [LL17], Local Interpretable Model-Agnostic Explanations (LIME) [RSG16] and Integrated Gradients [STY17], the temporal nature of data is often neglected when developing XAI methods for tabular data [SSV21]. Whilst there does exist the application of existing XAI methods to temporal data (e.g., [VABH22, SSV21, DP21]), to our knowledge, there is no local explanation method is designed to focus on the temporal nature of the data and the associated

change in prediction across instances. Hereinafter, we propose the adaptation of the Integrated Gradients (IG) method [STY17] to adhere to temporal data. Following this, we propose properties inherent from IG that conform to ideal properties for temporal data. Comparing with existing approaches to XAI, our method deviates away from instance based attribution and instead determines attribution with respect to the change in prediction value (probability) for regression (classification).

The occurrence of temporal data can be seen often in healthcare [SSV21]. Healthcare data is often stored in the form of Electronic Health Records (EHR) and an explanation is necessary when providing black-box predictions. Therefore, we consider an EHR case to demonstrate the proposed approach to produce explanations and provide a comparison against current state-of-the-art XAI methods when explaining temporal data.

Consider a breast cancer patient in the Simulacrum dataset¹:

Age: 78, Sex: Female, Site: 0, Morph: 8500, Weight: 84, Height: 1.67, **Dose Administration: 600 → 300 → 450 → 50**, Chemo Radiation: No, Regimen Outcome Description: N/A, **Admin Route: Subcutaneous → Subcutaneous → Intravenous → Oral**, Regimen Time Delay: No, Regimen Stopped Early: No, **Cycle Number: 1 → 3 → 3 → 5**, Cancer Plan: 2, Ethnicity: J, Behaviour: Malignant, Grade: G3, CReg Code: L1201, T Best: 2, N Best: 0, M Best: 0, Laterality: Left, CNS: Y1, ACE: 9, Performance: 1, Clinical Trial: Yes

There are four records of this patient, representing the sequences of treatments the patient has received. From these records, we observe that three features of this patient has gone through the following changes: drug dose administration (from **600** to **50**), cycle number (from **1** to **5**) and drug administration route (from **subcutaneous** to **oral**), while other features have remain unchanged. In the context of XAI, we pose the question:

How does each of these changing features affect the patient’s survival?

Answering such a question is critical for medical decision making [AB⁺20]. Yet, existing feature attribution algorithms in XAI [LOS⁺22] cannot directly answer this question, as they treat each record as an independent instance and do not consider temporal changes. In other words, state-of-the-art XAI explainers such as the SHAP and LIME would consider the above as four separate patients and

¹<https://simulacrum.healthdatainsight.org.uk/> - The Simulacrum is a synthetic dataset developed by Health Data Insight CiC derived from anonymous cancer data provided by the National Cancer Registration and Analysis Service, which is part of Public Health England.

provide attribution values to all features, instead of only analyzing the changing ones.

The rest of this chapter is organised as follows, we provide a background of the method which our model adopts. We then introduce a representation of temporal data and the details of the constructed method. In doing so, we identify properties that should be satisfied with regards to XAI methods in a temporal domain and compare this to current state-of-the-art explainer DeepSHAP [LL17]. Finally, we provide a controlled experiment with a working example providing empirical evidence of property satisfiability and a comparison of performance when recovering feature-attribution against a known ground truth for temporal data for both LIME and SHAP.

D.2 Method

We introduce Batch-IG as an extension to IG for temporal explainability over batches of time-based data, such that we explain a collection of sequential instances and determine the attribution with respect to each point within the time batch. We represent a time batch as a matrix $\chi \in \mathbb{R}^{T \times J}$. Given a time batch contains T time points, where each time point is a vector, then $\chi = \langle \mathbf{x}_1, \dots, \mathbf{x}_t, \dots, \mathbf{x}_T \rangle$.

Analysing the behaviour of the black-box model between time points helps to determine the behaviour of the model with respect to data of a temporal nature, so we analyse where between time points that a feature had the greatest change in importance. The insight provided into the change in partial derivatives with respect to time intervals could lead to deeper insight to which point the partial derivative had greater importance in changing in prediction. We introduce Batch-IG, by overriding the baseline with an iterative function using the prior time-step t as the baseline, and following step $t + 1$ as the target, therefore we have

$$\text{Batch-IG}(\chi) := \sum_{t=1}^{T-1} \sum_{j=1}^J \left((x_{t+1}^j - x_t^j) \times \int_{\alpha=0}^1 \frac{\partial F(\mathbf{x}_t + \alpha \times (\mathbf{x}_{t+1} - \mathbf{x}_t))}{\partial x^j} d\alpha \right).$$

This can then be computed through the Riemann approximation method, namely:

$$\text{Batch-IG}^{\mathcal{R}}(\chi) := \sum_{t=1}^{T-1} \sum_{j=1}^J \left((x_{t+1}^j - x_t^j) \times \sum_{k=1}^M \frac{\partial F(\mathbf{x}_t + \frac{k}{K} \times (\mathbf{x}_{t+1} - \mathbf{x}_t))}{\partial x^j} \times \frac{1}{M} \right)$$

where $(\mathbf{x}_t + \frac{k}{K} \times (\mathbf{x}_{t+1} - \mathbf{x}_t))$ takes an initial point in time for an instance \mathbf{x}_t , and integrates with respect to $x_t^j \in \mathbf{x}_t$ for all j over $\frac{k}{K}$ steps, where $\frac{k}{K} \in [0, 1]$ and $0 < k \leq K$ to approximate the path integral between \mathbf{x}_{t+1} and \mathbf{x}_t such that, sub-intervals have equal lengths between both points. Therefore, a larger value of

K will allow for a more accurate approximation of Batch-IG between points \mathbf{x}_{t+1} and \mathbf{x}_t .

D.3 Results

We present three sets of experiments in this section. Starting from an in-depth discussion on the Example shown in the Introduction, we first illustrate the steps involved in computing Batch-IG and outcomes of this case study. Then, we compare Batch-IG results with explanations found by DeepSHAP and demonstrate that both approaches produce distinctive explanations. Lastly, after witnessing differences between Batch-IG and DeepSHAP, we construct some controlled experiments with known “ground truth” to explanations and evaluate Batch-IG against several other state-of-the-art explainers.

D.3.0.1 Example Explanation

From the Simulacrum dataset, we isolate a cohort of patients with the ICD-10 code “C50” *Malignant neoplasm of breast*. We group patients by their patient unique identifiers, and within these groups we order the patients by cycle number to maintain temporally organised patient data as a means for generating explanation examples. We want to obtain explanations of the form, “*given features that change during the course of patient treatment, how do the changes effect the survival prediction probability?*”. Standard feature-attribution methods such as SHapley Additive exPlanations (SHAP) [LL17] and Local Interpretable Model-Agnostic Explanations (LIME) [RSG16] gives explanations of the form, “*given an instance, which features attributed towards the prediction probability?*”, yet, this is potentially problematic, as when we observe temporal data groups of data belonging to the same patient should not be viewed independently. For example, let us consider a temporal batch of instances of the same patient where the alterations at $\{t_0, t_1, t_2, t_3\}$ then we have the patient instance state transitions:

Dose Administration: 600 \rightarrow 300 \rightarrow 450 \rightarrow 50,

Admin Route: Subcutaneous \rightarrow Subcutaneous \rightarrow Intravenous \rightarrow Oral,

Cycle Number: 1 \rightarrow 3 \rightarrow 3 \rightarrow 5.

The following predictions given at each time interval:

$t_0 = 94.49\%$, $t_1 = 95.87\%$, $t_2 = 95.92\%$, $t_3=93.82\%$ towards the class ≥ 6 Months survival.

Therefore, upon generating explanation with respect to the introduced patient cycle, we see that the dose administration feature is the only attributed feature

under all 3 transitions, such that $t_0 : 600 \rightarrow t_1 : 300 \rightarrow t_2 : 450 \rightarrow t_3 : 50$, whereas the cycle number between the time intervals $t_1 \rightarrow t_2$ does not change, the attribution given for the cycle number feature is only evident in figures D.1 and D.3.

In Figures D.1, D.2 and D.3 we see the attribution of the controllable features F^d for a breast cancer patient through a set of 3 recorded cycles, with two instances under the same cycle, such that Cycle Number = $\{1,3,3,5\}$, such that the set of controllable features are given by

$$F^d = \{ \text{“Dose Administration”}, \text{“Cycle Number”}, \text{“Admin Route”} \}.$$

Observing each sub-figure, we determine that the most influential feature in altering predictions between time points is given by the adjustment to the drug dose administration for the patient.

The explanations highlights that the cycle number in the earlier cycles attribute towards a probability of longer survival, whereas in the final recorded cycle, the later cycle numbers attribute towards a shorter survival. Similarly, drug administration has positive influence over longer survival earlier in the cycle and negative influence in the final recorded cycle.

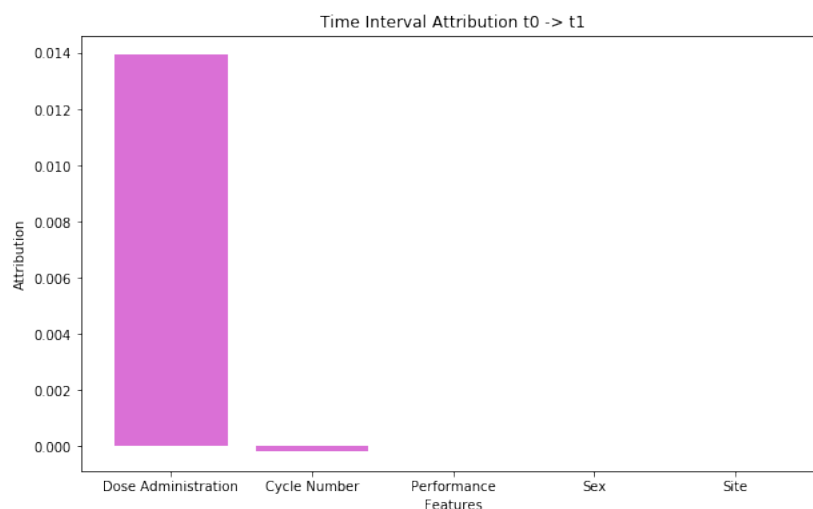


Figure D.1: Feature attribution for the features from time interval $t_0 \rightarrow t_1$. We observe the dose administration had **positive** attribution towards the class ≥ 6 Months and the cycle number transition from $1 \rightarrow 3$ had **negative** attribution towards the ≥ 6 Months class.

Furthermore, by evaluating attributions for time-intervals, we can then further explore the influential sub-intervals between two time points. From this, we can gauge between two time points, at which point the attribution becomes influential.

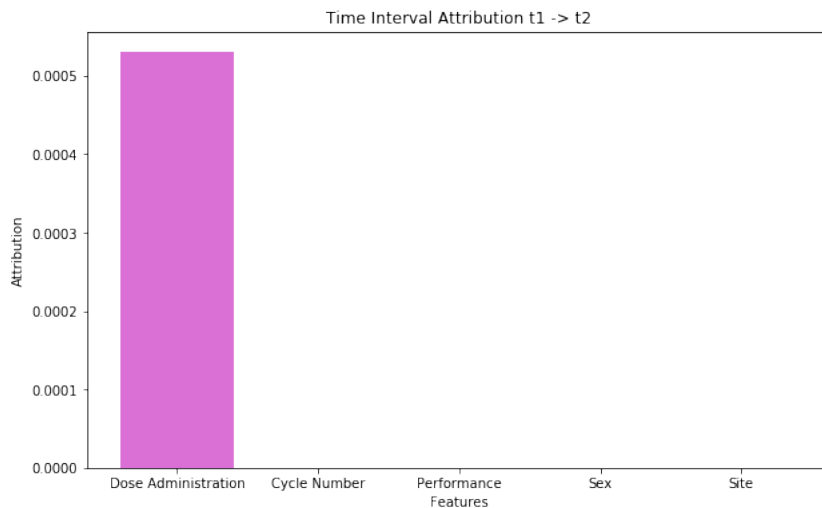


Figure D.2: Feature attribution for the features from time interval $t1 \rightarrow t2$. We observe the dose administration had **positive** attribution towards the class ≥ 6 Months. This time interval was observed during the drug cycle 3, where there exists only change to the drug administration.

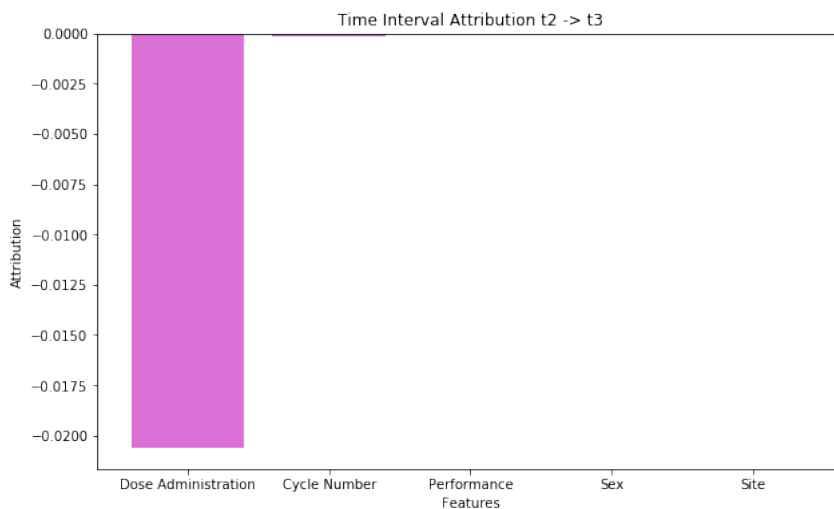


Figure D.3: Feature attribution for the features from time interval $t2 \rightarrow t3$. We observe the dose administration had **negative** attribution towards the class ≥ 6 Months and the cycle number transition from 3 \rightarrow 5 also had **negative** attribution towards the ≥ 6 Months class.

For example, let us consider an evaluation of dose administration (see Figures D.4, D.5 and D.6).

By observing the respective gradients within time intervals, we can see that the gradual decrease in dose administration began to have effects in the earlier part of the reduction in the first time interval (Figure D.4). Then the modification towards the latter part of $t1 \rightarrow t2$ had great attribution as the dose administration increased back to 450. Finally, the decrease to 50 towards the end of $t2 \rightarrow t3$ had the greatest negative influence w.r.t the prediction ≥ 6 Months this when paired with the total attribution given for time intervals.

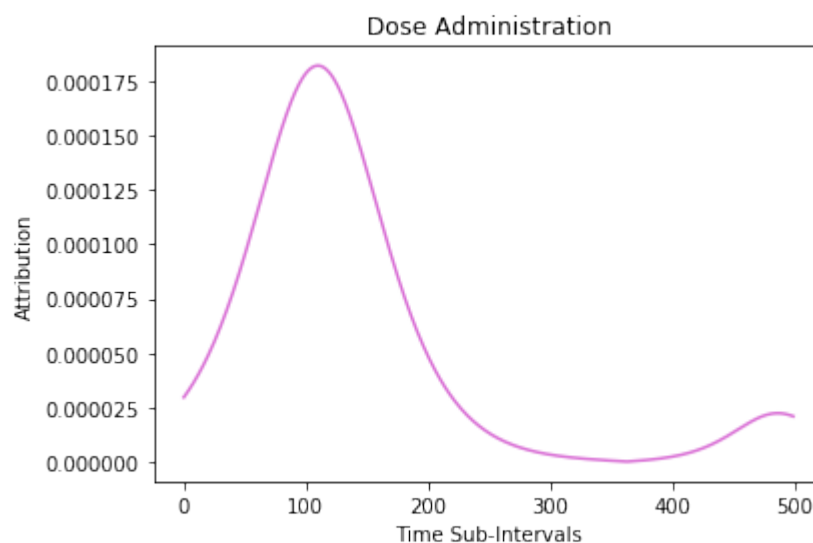


Figure D.4: Evaluation of the partial derivative of the prediction w.r.t the change in drug dose administration between time intervals $t0 \rightarrow t1$.

D.3.0.2 XAI Comparison

We compare explanations returned by Batch-IG ^{\mathcal{R}} and DeepSHAP, by evaluating the average Pearson r correlation coefficient of returned explanation vectors over the first 100 instances over 5 datasets, where the DeepSHAP approximation background set uses the whole training dataset and Batch-IG ^{\mathcal{R}} uses $M=5000$ steps for the Riemann approximation.

As DeepSHAP only produces feature attribution explanations for individual instances without considering the temporal aspect within our EHR data, we manually take the difference between two instances at each adjacent time intervals and consider that as the impact of changes. We observe that there's a moderate to high average correlation between explanations returned by Batch-IG ^{\mathcal{R}} and

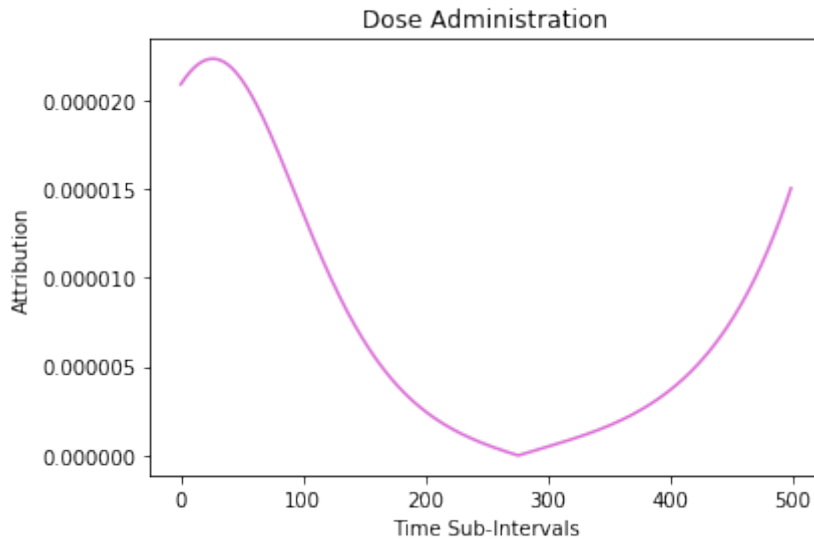


Figure D.5: Evaluation of the partial derivative of the prediction w.r.t the change in drug dose administration between time intervals $t_1 \rightarrow t_2$.

DeepSHAP across the majority of datasets with a relatively large standard deviation.

Table D.1: Comparison of explanations returned by SHAP and Batch-IG^R

Dataset	Average Correlation	Correlation Std.	Correlation Var.
Breast Cancer	0.52	0.22	0.05
Lung Cancer	0.71	0.30	0.09
Rectal Cancer	0.44	0.24	0.06
Lymphoma Cancer	0.67	0.27	0.07
Skin Cancer	0.46	0.27	0.07

D.3.0.3 Controlled Experiment

We generate a simple synthetic data set where the importance of features are known. Therefore, we define the input data set to be a $\mathbb{R}^{D \times 2}$ matrix of instances, where $D = 50,000$ and a label for an instance at time point t is given by:

$$p_t = 2 \sin(x_t^1) + 4 \sin(x_t^2).$$

Considering an instance given at time-point t and $t + 1$ respectively, keeping the subscript notation for time points, we have the time batch χ_i containing $\mathbf{x}_t =$

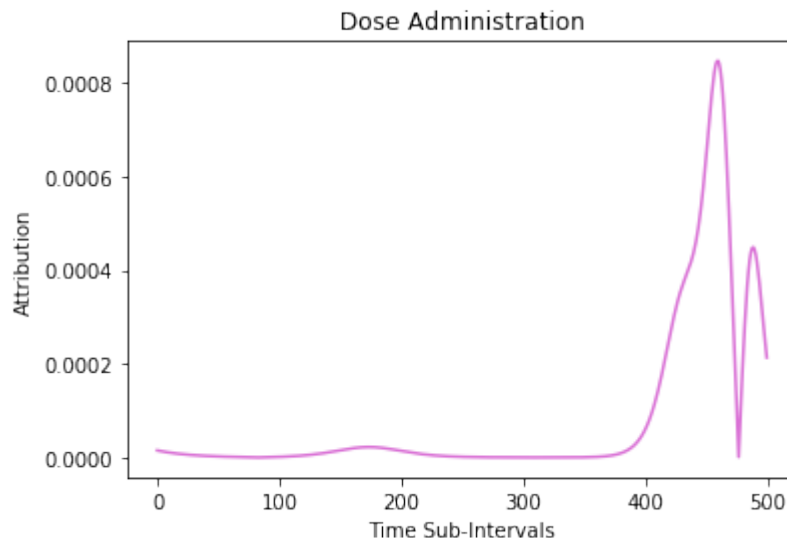


Figure D.6: Evaluation of the partial derivative of the prediction w.r.t the change in drug dose administration between time intervals $t_2 \rightarrow t_3$.

$\langle 2_t, 8_t \rangle$ and $\mathbf{x}_{t+1} = \langle 2_{t+1}, 7_{t+1} \rangle$. Generating the true labels we have $p_t \approx 5.776028$ and $p_{t+1} \approx 4.446541$, the change in value equates to $\Delta p_t \approx -1.329487$, therefore the difference in attribution is given by the difference in the second term of the equation at both time points, namely $\Delta p_t = 4 \sin(7) - 4 \sin(8)$, as the first term is the same. The predicted values are $p_t = 5.7793$ and $p_{t+1} = 4.4459$ where $\Delta p_t = -1.3334$. where Δp_t is given by a change in x^2 .

We compare Batch-IG to DeepSHAP, SHAP and LIME in this setting to determine if the returned attribution recovers the difference in prediction whilst correctly assigning attribution from our example (see Table D.2). We observe that Batch-IG indeed identifies feature change impact most accurately, exceeding all other methods.

D.4 Conclusion

In this chapter, we identify a gap in current literature surrounding XAI for temporal data. To combat this, we introduce Batch-IG as a modification to the IG framework to consider time and both dynamic and static features. We provide an empirical comparison between Batch-IG, SHAP, DeepSHAP and LIME. Similarly, we provide a quantitative comparison on a controlled example comparing the same methods. From this, we determine that Batch-IG preserves the true attribution from the controlled example.

Limitations of the proposed approach are the requirement of knowledge (e.g. the temporal data needs to be linked via an identifier) with respect to temporal

D. Conceptualising Batch-Integrated Gradients for Temporal EHR Explanations

Table D.2: We demonstrate attribution recovery for an instance, such that we know the ground truth. Therefore, the difference in predictions should be fully recovered by the attribution given to x^2 . As in the previous example, for calculation of attribution for all methods besides Batch-IG, we take the difference in attribution between $t + 1$ and t for example $\Phi(x^2) = \Phi(x_{t+1}^2) - \Phi(x_t^2)$.

XAI Method (Model)	$f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t)$	$\Phi(x^1)$	$\Phi(x^2)$
DeepSHAP (ANN)	-1.3334	-0.00016	-1.3332
Batch-IG (ANN)	-1.3334	0	-1.3334
LIME (XGBoost)	-1.3298	-0.005	-1.632
SHAP(XGBoost)	-1.3298	-0.002	-1.3278

sequences within the data. The model specificity of such approach limits the ML models that can be applied in order to use the Batch-IG framework. Similarly, as with other current methods, Batch-IG also assumes independence between features.

Appendix E

QUCE Supplementary Material

E.1 Computing QUCE Explanations

Proposition 1. *The QUCE explainer has a computable Riemann approximation solution for each feature.*

Proof. Given an instance \mathbf{x} that is the origin of our instance for our counterfactual explanation, we consider the \mathbf{x}^Δ steps produced by our learning process. In order to compute QUCE for feature explanations, we deconstruct the definition to focus on a single step of QUCE. This can be directly extracted from the additive property of integrals. Recall the definition:

$$\Phi_{\text{QUCE}}(\mathbf{x}^\Delta) := (\mathbf{x}_{\Delta_n} - \mathbf{x}_{\Delta_0}) \times \left(\int_{\mathbf{x}_{\Delta_0}}^{\mathbf{x}_{\Delta_n}} \nabla F(\psi(\alpha)) \cdot \psi'(\alpha) d\alpha \right).$$

Expanding this, we obtain

$$\begin{aligned} \Phi_{\text{QUCE}}(\mathbf{x}^\Delta) := & \left((\mathbf{x}_{\Delta_1} - \mathbf{x}_{\Delta_0}) \times \left(\int_{\mathbf{x}_{\Delta_0}}^{\mathbf{x}_{\Delta_1}} \nabla F(\psi(\alpha)) \cdot \psi'(\alpha) d\alpha \right) \right) \\ & + \dots + \left((\mathbf{x}_{\Delta_n} - \mathbf{x}_{\Delta_{n-1}}) \times \left(\int_{\mathbf{x}_{\Delta_{n-1}}}^{\mathbf{x}_{\Delta_n}} \nabla F(\psi(\alpha)) \cdot \psi'(\alpha) d\alpha \right) \right). \end{aligned}$$

Rewriting in terms of partial derivatives we get,

$$\begin{aligned} \Phi_{\text{QUCE}}(\mathbf{x}^\Delta) := & \sum_{j=1}^J \left(\int_{\mathbf{x}_{\Delta_0}}^{\mathbf{x}_{\Delta_1}} \frac{\partial F(\psi(\alpha))}{\partial \psi^j(\alpha)} \frac{\partial \psi^j(\alpha)}{\partial \alpha} d\alpha \right) \\ & + \dots + \sum_{j=1}^J \left(\int_{\mathbf{x}_{\Delta_{n-1}}}^{\mathbf{x}_{\Delta_n}} \frac{\partial F(\psi(\alpha))}{\partial \psi^j(\alpha)} \frac{\partial \psi^j(\alpha)}{\partial \alpha} d\alpha \right) \end{aligned}$$

where ψ^j is along a single feature dimension of a path ψ . With this decomposition, we can further extrapolate by defining each step in \mathbf{x}^Δ as a piecewise linear path, such that

$$\begin{aligned} \Phi_{\text{QUCE}}(\mathbf{x}^\Delta) &:= \\ &\sum_{j=1}^J \left((x_{\Delta_1}^j - x_{\Delta_0}^j) \times \int_{\alpha=0}^1 \frac{\partial F(\mathbf{x}_{\Delta_0} + \alpha(\mathbf{x}_{\Delta_1} - \mathbf{x}_{\Delta_0}))}{\partial x^j} d\alpha \right) \\ &+ \dots + \\ &\left((x_{\Delta_n}^j - x_{\Delta_{n-1}}^j) \times \int_{\alpha=0}^1 \frac{\partial F(\mathbf{x}_{\Delta_{n-1}} + \alpha(\mathbf{x}_{\Delta_n} - \mathbf{x}_{\Delta_{n-1}}))}{\partial x^j} d\alpha \right) \end{aligned}$$

and rewriting for a single feature j , we simply remove the summation and define the explanation over $x^{\Delta,j}$:

$$\begin{aligned} \Phi_{\text{QUCE}}(x^{\Delta,j}) &:= \\ &\left((x_{\Delta_1}^j - x_{\Delta_0}^j) \times \int_{\alpha=0}^1 \frac{\partial F(\mathbf{x}_{\Delta_0} + \alpha(\mathbf{x}_{\Delta_1} - \mathbf{x}_{\Delta_0}))}{\partial x^j} d\alpha \right) \\ &+ \dots + \\ &\left((x_{\Delta_n}^j - x_{\Delta_{n-1}}^j) \times \int_{\alpha=0}^1 \frac{\partial F(\mathbf{x}_{\Delta_{n-1}} + \alpha(\mathbf{x}_{\Delta_n} - \mathbf{x}_{\Delta_{n-1}}))}{\partial x^j} d\alpha \right). \end{aligned}$$

We can then rewrite this as a computable Riemann approximation for K steps, as

$$\begin{aligned} \Phi_{\text{QUCE}^\mathcal{R}}(x^{\Delta,j}) &:= \\ &\left((x_{\Delta_1}^j - x_{\Delta_0}^j) \times \frac{1}{K} \sum_{k=1}^K \frac{\partial F(\mathbf{x}_{\Delta_0} + \frac{k}{K}(\mathbf{x}_{\Delta_1} - \mathbf{x}_{\Delta_0}))}{\partial x^j} \right) \\ &+ \dots + \\ &\left((x_{\Delta_n}^j - x_{\Delta_{n-1}}^j) \times \frac{1}{K} \sum_{k=1}^K \frac{\partial F(\mathbf{x}_{\Delta_{n-1}} + \frac{k}{K}(\mathbf{x}_{\Delta_n} - \mathbf{x}_{\Delta_{n-1}}))}{\partial x^j} \right) \end{aligned}$$

which yields a computable explanation over the j^{th} dimension of an instance \mathbf{x} along a piecewise linear path. This can simply be executed across all features j . \square

The Riemann approximation is the algorithm for computing explanations that we use in our implementation carried out on each dimension j , which returns a vector containing the overall attribution over piecewise linear path integral formulation. It follows that the expected gradients variant can be easily computed

by averaging explanations over each counterfactual example in the set C .

Corollary 1. *The expected QUCE variant has a Riemann approximation solution for each feature.*

Proof. Given a set of k counterfactual examples in the set C , we can simply compute the expected attribution over the j^{th} feature of each generative counterfactual example as

$$\Phi_{\text{exQUCE}^{\mathcal{R}}}(x^{\Delta,j}) = \frac{1}{k} \sum_{\mathbf{x}_c \in C} \Phi_{\text{QUCE}^{\mathcal{R}}}(x^{\Delta,j}).$$

□

E.2 Proof of Proposition 2

Proposition 2. *Increasing the λ -tolerance of uncertainty provides a more flexible search space for possible paths to a generative counterfactual example.*

Proof. Recall the objective function in equation 7.1. For simplicity we let $\lambda_1 = 0$, $0 < \lambda_2 \leq 1$ and $0 \leq \lambda_3 \leq 1$, such that

$$\begin{aligned} \mathcal{G}(\mathbf{x}) &= \lambda_2 \mathcal{L}_\delta + \lambda_3 \mathcal{L}_\epsilon \\ &= \frac{\lambda_2}{2} \|\mathbf{x}_c - \mathbf{x}\|^2 + \\ &\lambda_3 (\mathbb{E}_{q_{\theta^*}} [\log q_{q_{\theta^*}}(\mathbf{z}|\mathbf{x}_c) - \log p_{\psi^*}(\mathbf{z})] - \mathbb{E}_{q_{\theta^*}} \log p_{\psi^*}(\mathbf{x}_c|\mathbf{z})). \end{aligned}$$

Trivially, as λ_3 decreases toward zero (we accept more uncertainty), the freedom in the distance function increases, as it not constrained by uncertainty, since

$$\begin{aligned} &\lim_{\lambda_3 \rightarrow 0^+} \left(\frac{\lambda_2}{2} \|\mathbf{x}_c - \mathbf{x}\|^2 + \right. \\ &\left. \lambda_3 (\mathbb{E}_{q_{\theta^*}} [\log q_{q_{\theta^*}}(\mathbf{z}|\mathbf{x}_c) - \log p_{\psi^*}(\mathbf{z})] - \mathbb{E}_{q_{\theta^*}} \log p_{\psi^*}(\mathbf{x}_c|\mathbf{z})) \right) \\ &= \frac{\lambda_2}{2} \|\mathbf{x}_c - \mathbf{x}\|^2. \end{aligned}$$

Here the eigenvalues of the Hessian are given by λ_2 and since $\lambda_2 > 0$, the Hessian is positive definite and thus the search space is convex, implying the global minima are conditioned on \mathcal{L}_δ when uncertainty \mathcal{L}_ϵ is relaxed. □

E.3 Proof of Proposition 3

Proposition 3. *Given the function $\Phi_{exQUCE}(\mathbf{x})$, the expected difference in prediction probabilities between generated counterfactuals in the set C with respect to the prediction probability given by $F(\mathbf{x})$, the following equality holds:*

$$\mathbb{E}_{\mathbf{x}_c \sim C, \alpha \sim \mathcal{U}(0,1)} \left[\Phi_{QUCE}(\mathbf{x}^\Delta) \right] \quad (\text{E.1})$$

$$= \mathbb{E}_{\mathbf{x}_c \sim C} \left[F(\mathbf{x}_c) - F(\mathbf{x}) \right]. \quad (\text{E.2})$$

Proof. Due to the *completeness* axiom the following holds true:

$$F(\mathbf{x}_{\Delta_n}) - F(\mathbf{x}_{\Delta_0}) = \quad (\text{E.3})$$

$$(\mathbf{x}_{\Delta_n} - \mathbf{x}_{\Delta_0}) \times \left(\int_{\mathbf{x}_{\Delta_0}}^{\mathbf{x}_{\Delta_n}} \nabla F(\psi(\alpha)) \cdot \psi'(\alpha) d\alpha \right) \quad (\text{E.4})$$

and extending this via $\mathbf{x}_{\Delta_n} = \mathbf{x}_c$ and $\mathbf{x}_{\Delta_0} = \mathbf{x}$ respectively, and thus we have

$$F(\mathbf{x}_c) - F(\mathbf{x})$$

while rearranging the RHS of equation E.4 we have

$$\underbrace{\mathbf{x}_c \times \left(\int_{\mathbf{x}}^{\mathbf{x}_c} \nabla F(\psi(\alpha)) \cdot \psi'(\alpha) d\alpha \right)}_{F(\mathbf{x}_c)} \quad (\text{E.5})$$

$$- \underbrace{\mathbf{x} \times \left(\int_{\mathbf{x}}^{\mathbf{x}_c} \nabla F(\psi(\alpha)) \cdot \psi'(\alpha) d\alpha \right)}_{F(\mathbf{x})} \quad (\text{E.6})$$

which follows by relaxing a strict definition of \mathbf{x}_c , where we instead use a set of generated counterfactuals in the set C . Our proposed approach takes the integral over \mathbf{x}_c with respect to a change in the set C , yielding

$$\int_{\mathbf{x}_c} \left(\mathbf{x}_c \times \left(\int_{\mathbf{x}}^{\mathbf{x}_c} \nabla F(\psi(\alpha)) \cdot \psi'(\alpha) d\alpha \right) \right) p_C(\mathbf{x}_c) d\mathbf{x}_c \quad (\text{E.7})$$

$$= \mathbb{E}_{\mathbf{x}_c \sim C, \alpha \sim \mathcal{U}(0,1)} \left[\mathbf{x}_c \times \nabla F(\psi(\alpha)) \cdot \psi'(\alpha) \right] \quad (\text{E.8})$$

$$= \mathbb{E}_{\mathbf{x}_c \sim C} \left[F(\mathbf{x}_c) \right] \quad (\text{E.9})$$

for equation E.5 and $F(\mathbf{x})$ for equation E.6. and since $F(\mathbf{x})$ is a constant, namely

$$\mathbb{E}_{\mathbf{x}_c \sim C} \left[F(\mathbf{x}_c) \right] - F(\mathbf{x}) = \mathbb{E}_{\mathbf{x}_c \sim C} \left[F(\mathbf{x}_c) - F(\mathbf{x}) \right], \quad (\text{E.10})$$

equations E.1 and E.2 are equivalent. \square

E.4 Counterfactual Reconstruction Error

In addition to evaluating the VAE loss, we also analyze the average reconstruction error per instance across all 100 instances on both the training and test datasets. This highlights the closeness of the reconstructed sample against the ground truth counterfactual generated by different methods. In Table E.1 we observe that our proposed QUCE method provides better reproduced counterfactuals through the VAE than either the DiCE or AGI methods.

Table E.1: Comparison of the average sum of feature-wise reconstruction error between original instances and their generated counterfactual examples. This is experimented on 100 instances for each dataset. Here we observe that the QUCE method performs best in generating counterfactuals with minimal uncertainty across all datasets.

CF Proximity	Lung	Breast	Skin	Lymph	Rectal	COVID	W-BC
Train							
QUCE	0.95	0.73	0.90	0.66	0.78	1.16	0.73
DiCE	1.80	0.95	1.33	1.01	1.18	1.38	1.04
AGI	2.55	1.89	2.25	0.91	2.91	2.19	0.80
Test							
QUCE	0.88	0.85	0.80	0.63	0.78	0.57	0.76
DiCE	1.81	0.95	1.32	0.99	1.18	1.38	1.04
AGI	1.53	2.41	1.66	0.92	2.08	0.79	0.81

E.5 QUCE Evaluated against Further Properties of Explainability

In the work of [ABN22] the authors present a desirable set of axiomatic foundations for XAI methods. As a brief informal overview, we consider the following proposed axioms:

- **Success:** The explainer method should be able to produce explanations for any instance.
- **Explainability:** An explanation method should provide informative explanations. An empty explanation here is not recommended.
- **Irreducibility:** An explanation should not contain irrelevant information.
- **Representativity:** An explanation should be possible on unseen instances.
- **Relevance:** Information should only be included if it impacts the prediction.

We evaluate QUCE against these properties as interpreted with respect to our model design. We show that it is inherently straightforward to prove that our proposed QUCE method satisfies these desirable axioms.

Proposition 13. *The QUCE method can always provide an explanation satisfying the success axiom.*

Proof. Whilst a counterfactual may not always be *valid*, as a direct implication of the generative learning process QUCE will achieve an explanation. \square

Corollary E.1 The QUCE method satisfies the Explainability axiom.

Proof. Assuming that different instances generated by QUCE do not have the same prediction probability with respect to the target class, it follows immediately from proposition 3 that explainability holds. \square

Corollary E.2 The QUCE method satisfies the Irreducibility axiom.

Proof. Here, we characterize irrelevance under our own interpretation: since a feature that does not change does not affect the predicted outcome, it should be assigned zero attribution. Then directly from the definition of QUCE it is clear that irreducibility holds, as the gradients are multiplied by a zero-value scalar for the same valued features. \square

Proposition 14. *The QUCE method satisfies the Representativity axiom.*

Proof. It is easy to see that any instance with the same dimensionality of the instances from a training dataset can utilize the QUCE approach. \square

Corollary E.3 The QUCE method satisfies the relevance axiom.

Proof. Relevance holds as a direct implication of irreducibility as seen in corollary E.2 and the fact that gradients are traced over the change in the predictions along paths, thereby guaranteeing model-specific relevance. \square

E.6 Experimental Setup

For the experiments presented in this paper, the details of hyper-parameters and experimental setup are found in the notebook file at: <https://github.com/jamie-duell/QUCE>. For the comparative experiments in this paper we set up QUCE with single-path solutions using the Adam optimizer. The empirical intuition for using Adam for a single-path approach is as follows: as the loss is minimized and points along the path become more reliable there will be an increase in the frequency of points as we approach the solution; thus averaging the gradient along such paths become more reliable. Similarly the approach

is deterministic, therefore does not require multiple paths as with alternative optimizers.

E.7 Deletion Experiments

To compare counterfactual feature attribution methods we evaluate the deletion score, a common metric used for evaluating feature attribution methods for identifying important features. The deletion score is used in various studies [YAWM23, AJ23]; here, a lower value indicates better performance. In Table E.2 we observe that the QUCE method performs better on average than both DiCE and AGI for counterfactual feature attribution performance.

Table E.2: Comparison of the deletion scores for counterfactual generative methods that provide feature attribution values. This is experimented over 100 instances on each dataset. Here the lower the value the better. We observe that the proposed QUCE method performs best across a larger fraction of datasets.

Deletion	Lung	Breast	Skin	Lymph	Rectal	COVID	W-BC
QUCE	0.556	0.689	0.656	0.611	0.669	0.699	0.632
DiCE	0.561	0.688	0.649	0.619	0.669	0.710	0.637
AGI	0.559	0.683	0.659	0.607	0.670	0.728	0.648

Appendix F

Multiple Value Imputation Experiments

F. Multiple Value Imputation Experiments

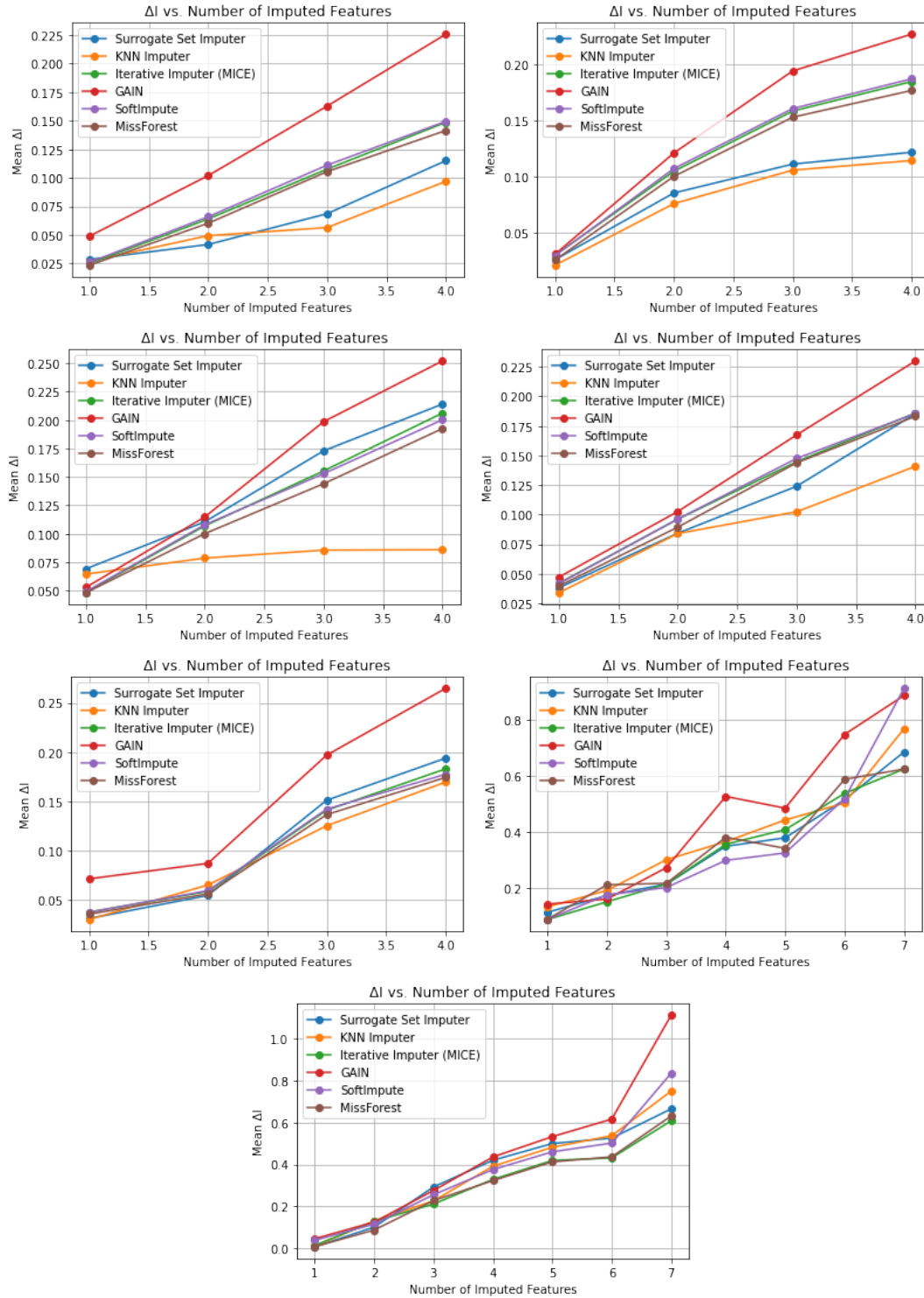


Figure F.1: Here we have the random imputation experiments on the ΔI metrics for (from left to right): SBC, SLC, SLyC, SRC, SSC, Diabetes and SEER datasets. Here we observe a competitive performance displayed with the proposed SSI method.

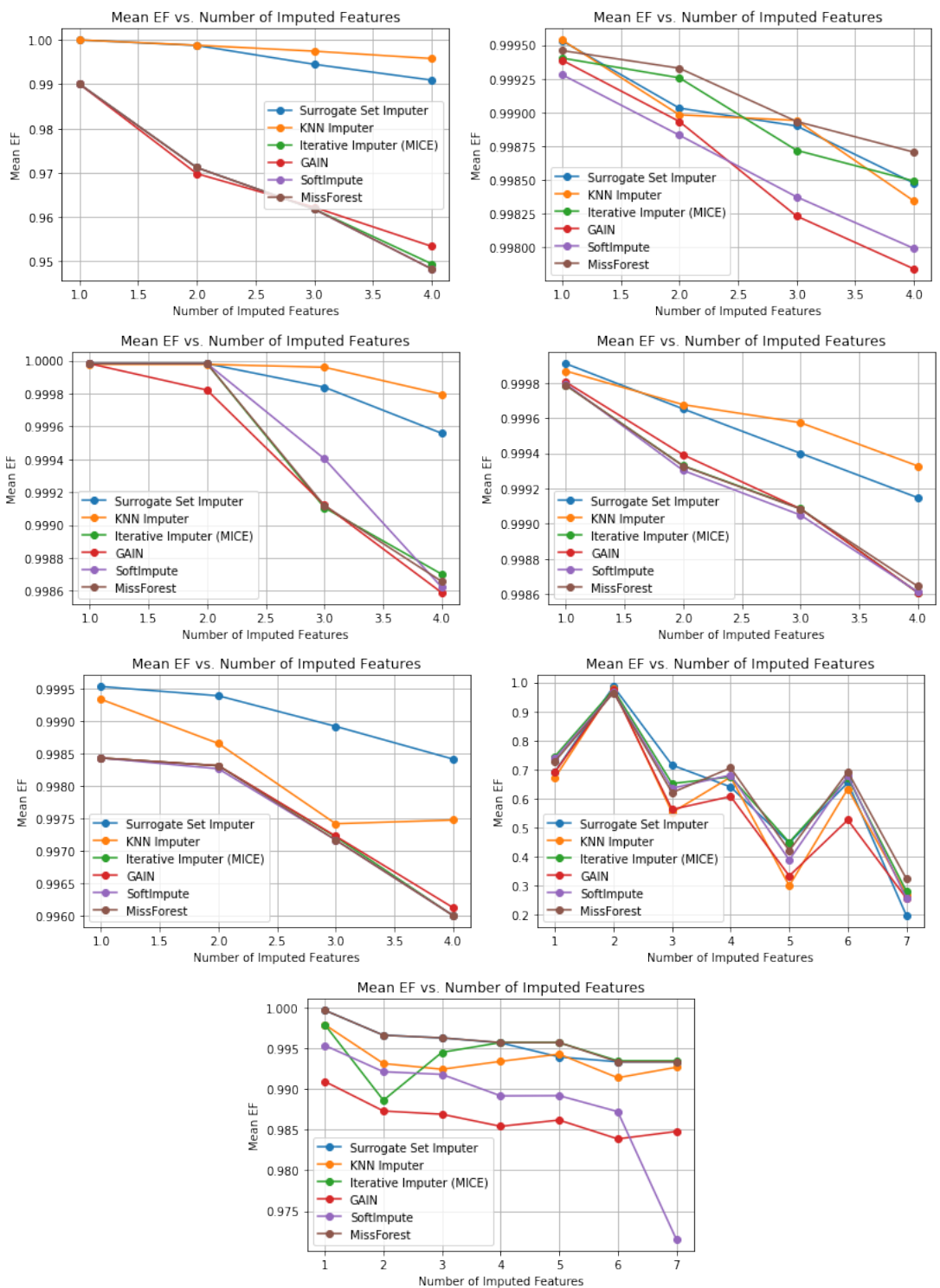


Figure F.2: Here we have the random imputation experiments for EF on the datasets (from left to right): SBC, SLC, SLyC, SRC, SSC, Diabetes and SEER. Here we observe aberration in performance with all methods, with a competitive performance displayed by the proposed SSI method.

Appendix G

Further Comparisons of Explanations on EHRs

G.1 Introduction

There is a significant demand for *Explainable Artificial Intelligence (XAI)* in medical machine learning, driven by the need for transparency, trust, and ethical deployment [ADG⁺21]. In the medical domain, where decisions can have life-or-death consequences, understanding the underlying factors behind machine learning predictions is crucial. Explainability enables healthcare professionals to comprehend the causal relationships between treatments and outcomes, facilitating informed decision-making.

Interpreting predictive models in the context of complex medical data, such as Electronic Health Records (EHRs), poses challenges. EHRs contain diverse patient information, making it difficult to extract actionable insights solely from black-box models. Feature attribution methods [LL17, RSG16] play a pivotal role in addressing this challenge by determining the importance of different features in the decision-making process of predictive models. By identifying influential features, clinicians gain valuable insights into the factors driving predictions, enabling them to make informed decisions about patient care.

Incorporating XAI techniques in medical machine learning not only enhances model transparency and trust but also upholds ethical standards. It empowers healthcare providers to explain treatment decisions to patients, promoting transparency and patient-centered care. XAI bridges the gap between complex machine learning algorithms and the need for comprehensible and justifiable decision-making in medicine, ultimately improving patient outcomes and ensuring responsible deployment of AI technologies in healthcare.

In recent years, there has been a surge in the development of feature attribution methods, leading to the need for comparative approaches to evaluate XAI models.

In their work, [DFB⁺21] presents a comparison of the most significant features identified by various XAI methods. Similarly, [NSBL21] conducts comparisons of explanations and employs the Kendall τ ranking to assess the consistency of these explanations. These studies highlight the lack of consensus or ubiquity in explanations provided by state-of-the-art XAI methods such as *Local Interpretable Model-Agnostic Explanations* (LIME) [RSG16] and *SHapley Additive exPlanations* (SHAP) [LL17].

The need for comparative analysis arises from the observation that different XAI methods may yield divergent explanations for the same model and data. These discrepancies raise questions about the reliability and robustness of XAI methods in providing consistent and reliable insights. The comparative evaluations conducted in [DFB⁺21], [NSBL21] and [YFL21] shed light on the variations in the identified important features and the consistency of explanations generated by different XAI techniques.

Counterfactual methods, such as *Diverse Counterfactual Explanations* (DiCE) [MST20], also incorporate feature attribution. In a comparative analysis conducted in [KMMTS21], correlations between LIME, SHAP, and DiCE were examined. The study revealed a weak correlation between SHAP and LIME across a large number of features, while the strongest correlation was observed between DiCE and LIME.

In terms of stability of important features, LIME has been shown to outperform SHAP [MC21]. Conversely, SHAP exhibits better performance under certain conditions due to its adherence to game theoretic principles [GG21]. In the work [KMMTS21], the authors acknowledge these differences and emphasize the importance of not relying solely on a single method. They highlight that as the dimensionality of a problem increases, inconsistencies become more apparent when applying these methods to 20 or more features, despite the presence of a high positive correlation in the explanations. These findings raise concerns regarding the plausibility and faithfulness of explanations.

The dissimilarity in XAI explanations identified by these studies underscores the need for cautious interpretation and consideration of multiple XAI methods. While each method offers unique insights and strengths, their discrepancies suggest the existence of limitations and challenges in achieving consistent and reliable explanations. Further research is essential to address these concerns and develop robust evaluation frameworks for XAI methods to enhance the interpretability and trustworthiness of AI systems.

Building upon this understanding, we propose an approach for comparison by leveraging the model-agnostic explanation methods, LIME and SHAP, to analyze the counterfactual data generated using DiCE. Our primary objective is to introduce additional metrics for comparison, considering that different explanation methods yield diverse representations of explanations. We also aim

to investigate the predictive capability of a model trained on factual data when applied to counterfactual data. This analysis is crucial in assessing the validity of the generated counterfactuals, as it plays a pivotal role in understanding causal effects.

Here, we have the following objectives:

- Determine the quality of counterfactual instances produced through predictive models;
- Propose metrics for the comparison of XAI methods;
- Compare the explanations given by XAI methods.

By addressing these objectives, we aim to enhance our understanding of XAI methods and their applicability in analyzing counterfactual data. This research contributes to the development of robust evaluation frameworks for XAI techniques, fostering greater transparency, interpretability, and trustworthiness in AI-driven decision-making processes.

G.2 Background

We focus on state-of-the-art deployable XAI methods LIME, SHAP and a generative counterfactual method DiCE. To ease representation, we introduce local and global explanations replicating the notation introduced in [AdF22].

XAI methods can be described such that given a black-box model f trained on a data set $X = \langle \mathbf{x}_1, \dots, \mathbf{x}_N \rangle$. A feature attribution algorithm produces a **local explanation** for an instance \mathbf{x}_i , where $\mathbf{x}_i \in \mathbb{R}^J$ has an associated explanation vector $\mathbf{e}_i \in \mathbb{R}^J$, such that $\mathbf{e}_i = (e^1, \dots, e^J)$, where e^j and $j(1 \leq j \leq J)$ represents the feature explanation for the i^{th} instance. A **global explanation** is an explanation over a dataset, where an associated global explanation vector is given by $\mathcal{E} \in \mathbb{R}^J$.

DiCE is a counterfactual data generative method. For each $\mathbf{x}_i \in X$, $\text{DiCE}(\mathbf{x}_i) = \mathbf{c}_i$ such that $\mathbf{c}_i \in \mathbb{R}^J$ and \mathbf{c}_i is “close” to \mathbf{x}_i , $f(\mathbf{c}_i) \neq f(\mathbf{x}_i)$. \mathbf{c}_i is referred to the counterfactual of \mathbf{z}_i ¹.

G.3 Method

Counterfactual explanations provide a valuable approach for conducting “what-if” analyses in the medical field. These explanations involve creating hypothetical

¹DiCE can create up to p explanations such that, a set of counterfactual instances $\{c_1, \dots, c_p\}$, can be generated that differs in prediction for an instance \mathbf{x}_i .

scenarios where closely related patients undergo different treatments or interventions, allowing us to observe the corresponding predictions for those patients. In many cases, the effects of treatments can vary significantly among individuals, making it challenging to establish clear causality by solely examining existing instances.

To address this challenge, it becomes crucial to shift our focus to the patients themselves and consider how altering specific aspects of an individual’s profile would impact the prediction. By generating counterfactual instances, we can explore personalized treatment effects and understand how changes to a patient’s characteristics or interventions would influence their predicted outcomes. This patient-centric approach enables a more fine-grained and specific analysis, accounting for the stochastic nature of treatment effects that can differ from one patient to another.

In essence, counterfactual explanations provide a powerful tool to investigate causal relationships and assess the potential effects of different interventions on individual patients. By generating and analyzing counterfactual instances, we can gain valuable insights into personalized treatment effects, enabling more informed and targeted decision-making in the medical field.

Formally, a **counterfactual generative method** over a data set X is $\Psi(X) : \mathbb{R}^{N \times J} \rightarrow \mathbb{R}^{M \times J}$. $\Psi(X)$ retains the features of X , but the number of instances may differ between $\Psi(X)$ and X . As such, we produce a counterfactual set C , where $C = \Psi(X)$. Specially, we define

$$C = \{\text{DiCE}(x_i) | x_i \text{ is an instance in } X\}.$$

The use of global explanations can help identify trends at a population level in EHR [HFL⁺21, KLS⁺22], given that EHRs containing a large population, the associated important features can be hard to determine when making predictions, as such, global explanations can help identify trends.

To produce a global explanations we take the mean absolute sum of explanations over all instances. Namely, given a feature attribution method $F \in \{\text{LIME}, \text{SHAP}\}$, we let the global explanation with respect to a dataset Z be:

$$\mathcal{E}^F(Z) = \frac{1}{b} \sum_{i=1}^b |F(\mathbf{z}_i)|. \tag{G.1}$$

Note that Z can be either factual X or counterfactual C . Also, we take the absolute value on F to generalise importance by rank.

In this work we compare explanations using the Pearson r correlation coefficient, Jaccard Similarity Index and Attribution Space, introduced as follows.

G.3.1 Correlation of Attribution

The Pearson correlation coefficient, denoted as r , measures the linear relationship between two variables. It is calculated as the covariance of the two variables divided by the product of their standard deviations. The formula for the Pearson r correlation is:

$$r = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2} \sqrt{\sum(Y_i - \bar{Y})^2}} \quad (\text{G.2})$$

where:

- X_i and Y_i are the individual data points in the two variables, and
- \bar{X} and \bar{Y} are the means of the two variables.

G.3.2 Jaccard Similarity Index

We introduce the Jaccard similarity index as a means of comparing the most important features returned by two feature attribution methods F_1 and F_2 . Let S_1 and S_2 be the top v features from global explanations $\mathcal{E}^{F_1}(Z)$ and $\mathcal{E}^{F_2}(Z)$ respectively, the Jaccard Similarity Index is

$$J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}. \quad (\text{G.3})$$

G.3.3 Attribution Space

We propose a metric of *attribution space* for comparison, which compares the positive or negative attribution assigned by each feature in a pairwise form, given two XAI methods. Therefore, we generate the global attribution without absolute values, such that,

$$\mathcal{E}'^F(Z) = \frac{1}{b} \sum_{i=1}^b F(\mathbf{z}_i). \quad (\text{G.4})$$

From this, we can identify the proportion of shared agreement in the attribution space towards a given class. We consider the attribution space as shared if the explanation is in the same positive, negative or null space, over the number of features b . We can determine if the respective sgn^2 is shared. In other words, we

²The sgn function represents the sign returned. In this work we consider the sgn function to be defined by

$$\text{sgn}(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x = 0 \\ -1, & \text{otherwise.} \end{cases}$$

can represent a comparison of attribution space as:

$$\text{AttributionSpace}(Z_1, Z_2) = \frac{1}{J} \sum_{j=1}^J \mathbb{1}_{[\text{sgn}(\mathcal{E}'^F(Z_1)) = \text{sgn}(\mathcal{E}'^F(Z_2))]}.$$

G.4 Dataset and Prediction Result

The objective of this chapter is twofold: to assess the performance of a counterfactual generator through predictive analysis and to provide feature attribution-based explanations for both counterfactual and original data, comparing them with counterfactual explanations. To conduct this study, we utilize the Simulacrum synthetic data set³, which serves as a representation of the National Cancer Registration and Analysis Service (NCRAS) data set⁴. The data set pertains to a binary medical classification problem focused on lung cancer. Specifically, we conduct our experiments based on answering a binary classification problem:

Given a set of features for a lung cancer patient, whether the patient will survive longer than 6 months?

The patients in the data set are divided into two classes: those who survive *less than 6 months* and those who survive *longer than 6 months*. Each instance is described by 26 features. To generate counterfactual instances, we employ the Diverse Counterfactual Explanations (DiCE) method. For the classification task, we utilize the eXtreme Gradient Boosting (XGBoost) algorithm. The optimal parameters for XGBoost are determined through 10-fold cross-validation.

In our study, we use the factual data set as the training data and the counterfactual data set as the test data. This combined data set consists of a total of 4,385 instances, with a split ratio of approximately 57% for training and 43% for testing. By examining the predictive performance of a model trained on true data when applied to new counterfactual instances, we can evaluate the generalization capability of the model to unseen scenarios.

To assess the performance of the model on the counterfactual test set, we employ various performance metrics. The results of these metrics are presented in Table G.1, which provides insights into the accuracy and effectiveness of the model in predicting outcomes for the counterfactual instances.

By analyzing the performance metrics on the counterfactual test set, we gain valuable insights into the model’s ability to handle new and altered scenarios. These findings shed light on the model’s robustness and its capacity to make accurate predictions in the context of counterfactual data, contributing to our understanding of the model’s reliability and applicability in real-world scenarios.

³<https://simulacrum.healthdatainsight.org.uk/>

⁴http://www.ncin.org.uk/about_ncin/

Table G.1: XGBoost performance metrics where we use NonCF for training and CF for testing

Survival Time	Precision (%)	Recall(%)	F1 Score(%)
< 6 Months	83	86	84
> 6 Months	82	78	80
Accuracy (%)	82.4		

G.5 Explanation Results

We generate global explanations over the factual and counterfactual data sets for LIME shown in Figure G.1 and SHAP shown in Figure G.2. From this, we can observe that for LIME, “M Best”, being the *presence or absence of distant metastatic spread* and “Weight” in the given order, are the most important features when predicting survival time across both the factual and counterfactual datasets. Conversely, for SHAP, “Weight” and “M Best” in the given order, are the most important features in predicting survival time.

We use $\mathcal{E}^{\text{LIME}}(X)$ for the factual LIME set (LIME-NonCF) and $\mathcal{E}^{\text{LIME}}(C)$ for the counterfactual set (LIME-CF). Similarly, we refer to SHAP as $\mathcal{E}^{\text{SHAP}}(Z)$, with explanations for sets SHAP-NonCF and SHAP-CF respectively.

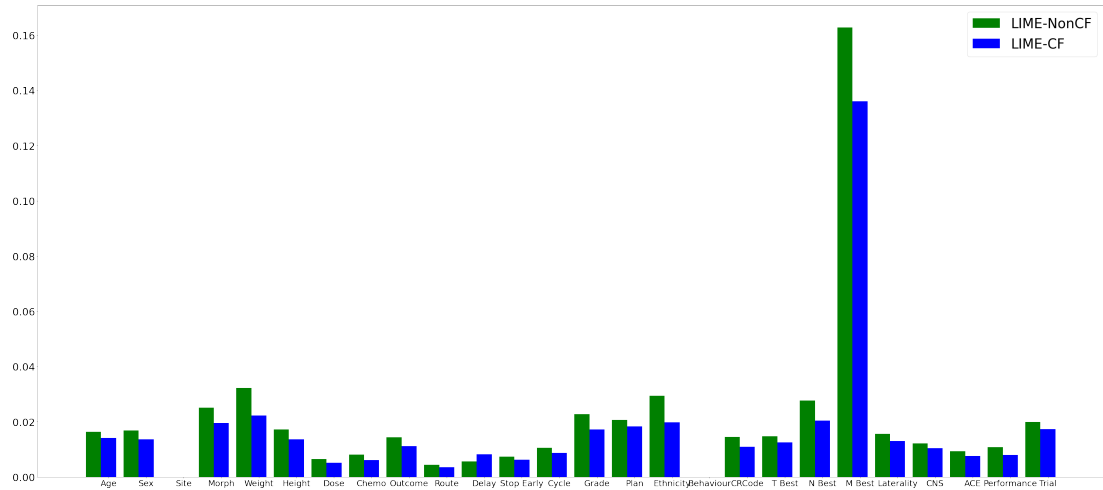


Figure G.1: Global explanation for LIME across the factual (NonCF) and counterfactual (CF) data set. From this, we can observe that across both factual and counterfactual datasets, “M Best” is the most important feature. We observe a strong similarity in feature attribution towards predictions.

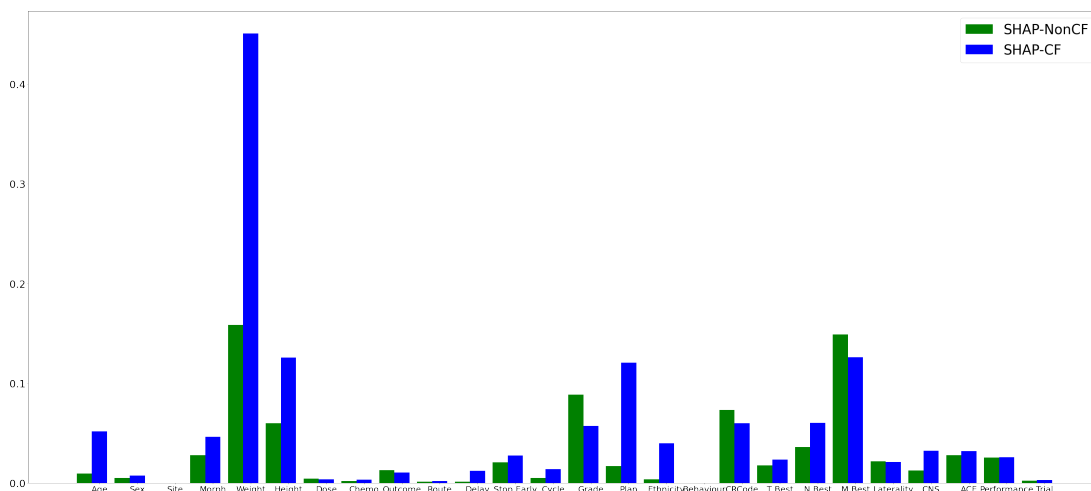


Figure G.2: Global explanation for SHAP across the factual (NonCF) and counterfactual (CF) data set. From this, we can observe that across both factual and counterfactual datasets, there’s a similarity in feature attribution towards predictions, with the most importance on the feature “Weight”.

G.5.1 Correlation

We compare the global explanations, to determine the overall r correlation for the attribution methods LIME and SHAP across both factual and counterfactual datasets, this illustrated in Figure G.3. We observe, the correlation range between

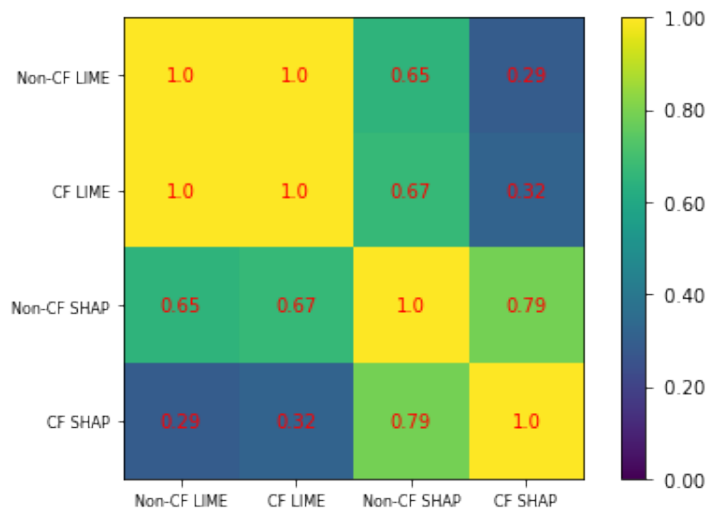


Figure G.3: Demonstrating the pearson correlation between the global explanations from SHAP and LIME.

explanations exists such that $0.29 \leq r \leq 1$, where SHAP across both sets ranges

from $0.79 \leq r \leq 1$ and LIME has a correlation of $r = 1$ when compared with itself across both sets, indicating SHAP has greater change in explanation when introduced to the counterfactual set. Conversely, SHAP and LIME explanations when compared have a correlation of $0.29 \leq r \leq 0.67$, where the greatest correlations exists between explanations of LIME and SHAP on the factual set, and between SHAP on the factual set, with LIME on the counterfactual set.

G.5.2 Jaccard Similarity

In this chapter we explore $v = 5$, which will determine the number of shared features within a set of 5 most important features from the feature attribution methods. From Table G.2, we can determine the Jaccard similarity between LIME and SHAP across both the factual and counterfactual datasets. We observe, LIME consistently determines the same features as important across both sets, whereas SHAP holds 67% similarity when compared across both sets. There exists little similarity between LIME and SHAP.

Table G.2: Jaccard Index $v = 5$

Dataset-Method	NonCF-SHAP	NonCF-LIME	CF-SHAP	CF-LIME
NonCF-SHAP	1	0.25	0.67	0.25
NonCF-LIME		1	0.25	1
CF-SHAP			1	0.25
CF-LIME				1

G.5.3 Shared Attribution

From Table G.3 we determine the shared attribution space of the LIME and SHAP methods, as they provide a sign in either positive and negative space with respect to the attribution towards a prediction. From this we determine that LIME across both factual and counterfactual sets holds the greatest similarity in attribution space, followed by SHAP across both sets. Between SHAP and LIME we see a greater agreement of 50% on the counterfactual generated set, as opposed to 46% on the factual sets.

Table G.3: Shared Attribution

Dataset-Method	NonCF-SHAP	NonCF-LIME	CF-SHAP	CF-LIME
NonCF-SHAP	1	0.46	0.77	0.5
NonCF-LIME		1	0.38	0.88
CF-SHAP			1	0.5
CF-LIME				1

G.6 Conclusion

In this study, we have conducted a comprehensive comparison of feature attribution methods, namely LIME and SHAP, in the context of counterfactual and factual data sets. We have introduced a novel pairwise comparative method called Attribution Space, which allows us to examine the direction of feature importance in positive or negative space.

Our findings emphasize the varying similarities and observations among different XAI methods when applied to different data conditions, specifically factual and counterfactual data. We have observed that SHAP exhibits a greater degree of change when presented with data sets that deviate from the factual data, while LIME demonstrates more consistency within itself. Moreover, we have noticed limited agreement between LIME and SHAP across all comparison metrics when analyzing Electronic Health Records (EHRs). Therefore, it is evident that employing a combination of XAI methods can provide more comprehensive insights.

The lack of consistency in explanations underscores the importance of using multiple XAI methods and carefully evaluating the explanations in the context of EHRs. While certain explanations may exhibit similarity in terms of top features or correlation, it is crucial to interpret these findings cautiously. These consistent explanations may indicate a higher likelihood of feature importance, providing valuable insights into the predictive models.

Moving forward, there are several avenues for future research. First, it would be beneficial to explore additional XAI methods and compare their performance and consistency in explaining EHRs. Furthermore, investigating the impact of different similarity metrics and evaluation techniques on explanation consistency could provide a deeper understanding of the interpretability of XAI methods. Additionally, considering real-world medical scenarios and evaluating the practical implications of XAI methods in decision-making processes would be valuable. By addressing these areas, we can further enhance the reliability, transparency, and trustworthiness of AI-driven systems in the medical field.

Bibliography

- [AB18] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.
- [AB⁺20] Julia Amann, , Alessandro Blasimme, Effy Vayena, Dietmar Frey, and Vince I. Madai. Explainability for artificial intelligence in healthcare: a multidisciplinary perspective. *BMC Medical Informatics and Decision Making*, 20(1), 2020.
- [ABA⁺21] Javier Antoran, Umang Bhatt, Tameem Adel, Adrian Weller, and José Miguel Hernández-Lobato. Getting a CLUE: A method for explaining uncertainty estimates. In *International Conference on Learning Representations*, 2021.
- [ABN22] Leila Amgoud and Jonathan Ben-Naim. Axiomatic foundations of explainability. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 636–642. International Joint Conferences on Artificial Intelligence Organization, 2022.
- [AdF22] Guilherme Seidyo Imai Aldeia and Fabrício Olivetti de França. Interpretability in symbolic regression: a benchmark of explanatory methods using the feynman data set. *Genetic Programming and Evolvable Machines*, 2022.
- [ADG⁺21] Anna Markella Antoniadis, Yuhan Du, Yasmine Guendouz, Lan Wei, Claudia Mazo, Brett A. Becker, and Catherine Mooney. Current challenges and future opportunities for XAI in machine learning-based clinical decision support systems: A systematic review. *Applied Sciences*, 11(11), 2021.
- [AFF23] Fabrizio Angiulli, Fabio Fassetti, and Luca Ferragina. Reconstruction error-based anomaly detection with few outlying examples. *arXiv*, 2023.

- [AJ23] Naveed Akhtar and Mohammad A. A. K. Jalwana. Towards credible visual model interpretation with path attribution. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 439–457. PMLR, 2023.
- [AKH12] Ilya Sutskever, A. Krizhevsky, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84–90, 2012.
- [AMMN16] Hiba Asri, Hajar Mousannif, Hassan Al Moatassime, and Thomas Noel. Using machine learning algorithms for breast cancer risk prediction and diagnosis. *Procedia Computer Science*, 83:1064–1069, 2016. The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / The 6th International Conference on Sustainable Energy Information Technology (SEIT-2016) / Affiliated Workshops.
- [ASFL11] Melissa J. Azur, Elizabeth A. Stuart, Constantine Frangakis, and Philip J. Leaf. Multiple imputation by chained equations: what is it and how does it work? *Int J Methods Psychiatr Res*, 20(1):40–49, 2011.
- [Bal12] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham Taylor, and Daniel Silver, editors, *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pages 37–49, Bellevue, Washington, USA, 2012. PMLR.
- [BBGOR06] Stef Van Buuren, Jaap P.L. Brand, Catharina G.M. Groothuis-Oudshoorn, and Donald B. Rubin. Fully conditional specification in multivariate imputation. *Journal of Statistical Computation and Simulation*, 76(12):1049–1064, 2006.
- [BBM⁺15] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS One*, 10(7):e0130140, 2015.
- [BCN20] Vaishnavi Bhargava, Miguel Couceiro, and Amedeo Napoli. Lime-out: An ensemble approach to improve process fairness. In *ECML PKDD 2020 Workshops*, pages 475–491, Cham, 2020. Springer International Publishing.

-
- [BCR97] José M. Benítez, Juan Castro, and Ignacio Requena. Are artificial neural networks black boxes? *IEEE transactions on neural networks*, 8 5:1156–64, 1997.
- [BDGP22] Joaquín Borrego-Díaz and Juan Galán-Páez. Explainable artificial intelligence in data science. *Minds Mach. (Dordr.)*, 2022.
- [BG20] Rickard Brüel Gabriëlsson. Universal function approximation on graphs. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19762–19772. Curran Associates, Inc., 2020.
- [BHTL20] Steven Bramhall, Hayley Horn, Michael Tieu, and Nibhrat Lohia. Qlime-a quadratic local interpretable model-agnostic explanation approach. In *SMU Data Science Review: No. 1 , Article 4.*, volume 3, 2020.
- [BL17] Michael Barras and Amy Legg. Drug dosing in obese adults. *Australian Prescriber*, 40:189–193, 2017.
- [BMP23] Subrato Bharati, M. Rubaiyat Hossain Mondal, and Prajoy Podder. A review on explainable artificial intelligence for healthcare: Why, how, and when? *IEEE Transactions on Artificial Intelligence*, pages 1–15, 2023.
- [BPSK17] Neeraj Bhargava, Renuka Purohit, Sakshi Sharma, and Abhishek Kumar. Prediction of arthritis using classification and regression tree algorithm. In *2017 2nd International Conference on Communication and Electronics Systems (ICCES)*, pages 606–610, 2017.
- [BS16] Lorenzo Beretta and Alessandro Santaniello. Nearest neighbor imputation algorithms: a critical evaluation. *BMC Medical Informatics and Decision Making*, 16(S3), 2016.
- [BSKM16] Larry E. Beutler, Kathleen Someah, Satoko Kimpara, and Kimberley Miller. Selecting the most appropriate treatment for each patient. *International Journal of Clinical and Health Psychology*, 16(1):99–108, 2016.
- [BSSG20] Joseph W. Bull, Niels Strange, Robert J. Smith, and Ascelin Gordon. Reconciling multiple counterfactuals when evaluating biodiversity conservation impact in social-ecological systems. *Conservation Biology*, 35(2):510–521, 2020.

- [CFB⁺21] José M. Clementino, Bruno S. Façal, Christian C. Bones, Caetano Traina, Marco A. Gutierrez, and Agma J. M. Traina. Multilevel clustering explainer: An explainable approach to electronic health records. In *2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS)*, pages 253–258, 2021.
- [CG16] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, pages 785–794, New York, NY, USA, 2016. Association for Computing Machinery.
- [CH67] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
- [Cla87] William Clancey. *Knowledge-Based Tutoring: The GUIDON Program*. MIT Press, 1987.
- [CLE⁺21] Hugh Chen, Scott M. Lundberg, Gabriel Erion, Jerry H. Kim, and Su-In Lee. Forecasting adverse surgical events using self-supervised transfer learning for physiological signals. *NPJ Digital Medicine*, 4(1), 2021.
- [CMR⁺23] Xun-Qi Chen, Chao-Qun Ma, Yi-Shuai Ren, Yu-Tian Lei, Ngoc Quang Anh Huynh, and Seema Narayan. Explainable artificial intelligence in finance: A bibliometric review. *Finance Research Letters*, 56:104145, 2023.
- [CPC19] Diogo Carvalho, Eduardo Pereira, and Jaime Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8:832, 2019.
- [CRBD18] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18*, pages 6572–6583, Red Hook, NY, USA, 2018. Curran Associates Inc.
- [CVDS21] Jonathan Crabbé and Mihaela Van Der Schaar. Explaining time series predictions with dynamic masks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2166–2177. PMLR, 2021.

-
- [DCL⁺18] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Pai-Shun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Neural Information Processing Systems*, 2018.
- [DDDV22] Alexander Dolk, Hjalmar Davidsen, Hercules Dalianis, and Thomas Vakili. Evaluation of LIME and SHAP in explaining automatic ICD-10 classifications of swedish gastrointestinal discharge summaries. In *Linköping Electronic Conference Proceedings*. Linköping University Electronic Press, 2022.
- [DFB⁺21] Jamie Duell, Xiuyi Fan, Bruce Burnett, Gert Aarts, and Shang-Ming Zhou. A comparison of explanations given by explainable artificial intelligence methods on analysing electronic health records. In *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 1–4, 2021.
- [DFFS23] Jamie Duell, Xiuyi Fan, Hsuan Fu, and Monika Seisenberger. Batch integrated gradients: Explanations for temporal electronic health records. In Jose M. Juarez, Mar Marcos, Gregor Stiglic, and Allan Tucker, editors, *Artificial Intelligence in Medicine*, pages 120–124, Cham, 2023. Springer Nature Switzerland.
- [DFS22] Jamie Duell, Xiuyi Fan, and Monika Seisenberger. Towards polynomial adaptive local explanations for healthcare classifiers. In *Foundations of Intelligent Systems, Proceedings of ISMIS 2022*, volume LNCS/LNAI 13515. Springer, 2022.
- [DFS23] Jamie Duell, Xiuyi Fan, and Monika Seisenberger. A comparison of global explanations given on electronic health records. In *International Conference on Intelligent Autonomous Systems (IASIS-2023)*, 2023.
- [DFsY⁺21] Weinan Dong, Daniel Yee Tak Fong, Jin sun Yoon, Eric Yuk Fai Wan, Laura Elizabeth Bedford, Eric Ho Man Tang, and Cindy Lo Kuen Lam. Generative adversarial networks for imputing missing data for big data clinical research. *BMC Medical Research Methodology*, 21(1), 2021.
- [DIKT19] Diptesh Das, Junichi Ito, Tadashi Kadowaki, and Koji Tsuda. An interpretable machine learning model for diagnosis of Alzheimer’s disease. *PeerJ*, 7, 2019.

- [DK19] Thomas Davenport and Ravi Kalakota. The potential for artificial intelligence in healthcare. *Future Healthcare Journal*, 6(2):94–98, 2019.
- [DKW⁺21] Carlo Dindorf, Jürgen Konradi, Claudia Wolf, Bertram Taetz, Gabriele Bleser, Janine Huthwelker, Friederike Werthmann, Eva Bartaguiz, Johanna Kniepert, Philipp Drees, Ulrich Betz, and Michael Fröhlich. Classification and automated interpretation of spinal posture data using a pathology-independent classifier and explainable artificial intelligence (XAI). *Sensors (Basel)*, 21(18):6323, 2021.
- [DP21] Abhirup Dikshit and Biswajeet Pradhan. Interpretable and explainable ai (XAI) model for spatial drought prediction. *Science of The Total Environment*, 801:149797, 2021.
- [dPIZC⁺16] Sheng dong Pan, Ling ling Zhu, Meng Chen, Ping Xia, and Quan Zhou. Weight-based dosing in medication use: what should we know? *Patient Preference and Adherence*, 10:549–560, 2016.
- [dPM⁺21] Harry Freitas da Cruz, Boris Pfahringer, Tom Martensen, Frederic Schneider, Alexander Meyer, Erwin Bottinger, and Matthieu P. Schapranow. Using interpretability approaches to update “black-box” clinical prediction models: an external validation study in nephrology. *Artificial Intelligence in Medicine*, 111:101982, 2021.
- [DRA⁺20] Sanket S. Dhruva, Joseph S. Ross, Joseph G. Akar, Brittany Caldwell, Karla Childers, Wing Chow, Laura Ciaccio, Paul Coplan, Jun Dong, Hayley J. Dykhoff, Stephen Johnston, Todd Kellogg, Cynthia Long, Peter A. Noseworthy, Kurt Roberts, Anindita Saha, Andrew Yoo, and Nilay D. Shah. Aggregating multiple real-world data sources using a patient-centered health-data-sharing platform. *NPJ Digit Med*, 3:60, 2020.
- [DSF23] Jamie Duell, Monika Seisenberger, and Xiuyi Fan. Counterfactual-integrated gradients: Counterfactual feature attribution for medical records. In *IEEE International Conference on Bioinformatics and Biomedicine (IEEE BIBM)*, 2023.
- [DSZ⁺23] Jamie Duell, Monika Seisenberger, Tianlong Zhong, Hsuan Fu, and Xiuyi Fan. A formal introduction to batch-integrated gradients for temporal explanations. In *IEEE 35th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2023.

-
- [EJS⁺21] Gabriel Erion, Joseph D. Janizek, Pascal Sturmfels, Scott M. Lundberg, and Su-In Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature Machine Intelligence*, 3(7):620–631, 2021.
- [EVS⁺22] Seda Polat Erdeniz, Sai Veeranki, Michael Schrempf, Stefanie Jauk, Thi Ngoc Trang Tran, Alexander Felfernig, Diether Kramer, and Werner Leodolter. Explaining machine learning predictions of decision support systems in healthcare. *Current Directions in Biomedical Engineering*, 8(2):117–120, 2022.
- [Fan22] Xiuyi Fan. Rule-psat: Relaxing rule constraints in probabilistic assumption-based argumentation. In Francesca Toni, Sylwia Polberg, Richard Booth, Martin Caminada, and Hiroyuki Kido, editors, *Computational Models of Argument - Proceedings of COMMA 2022, Cardiff, Wales, UK, 14-16 September 2022*, volume 353 of *Frontiers in Artificial Intelligence and Applications*, pages 152–163. IOS Press, 2022.
- [FMG⁺20] Seth Flaxman, Swapnil Mishra, Axel Gandy, H Unwin, H Coupland, T Mellan, H Zhu, T Berah, J Eaton, P Perez Guzman, et al. Report 13: Estimating the number of infections and the impact of non-pharmaceutical interventions on COVID-19 in 11 european countries. Technical report, Imperial College London, 2020.
- [Frä20] Kary Främling. Decision Theory Meets Explainable AI. In *Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, volume 12175, pages 57–74, Cham, 2020. Springer International Publishing.
- [FS80] Lawrence M. Fagan and E. Shortliffe. Computer-based medical decision making: From mycin to vm. *Automedica*, 3(2):97–108, 1980.
- [FT15] Xiuyi Fan and Francesca Toni. On computing explanations in argumentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), 2015.
- [Fu94] Limin Fu. Rule generation from neural networks. *IEEE Trans. Syst. Man Cybern. Syst.*, 24:1114–1124, 1994.
- [FZTN22] Tianshu Feng, Zhipu Zhou, Joshi Tarun, and Vijayan N. Nair. Comparing Baseline Shapley and Integrated Gradients for Local Explanation: Some Additional Insights. *arXiv*, 2022.

- [Gab11] Adam Gabbatt. IBM computer Watson wins Jeopardy clash. *The Guardian*, 2011.
- [GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [GCHW23] Luke Guerdan, Amanda Coston, Kenneth Holstein, and Zhiwei Steven Wu. Counterfactual prediction under outcome measurement error. In *FAccT*, pages 1584–1598, 2023.
- [GFBG21] Victor Guyomard, Françoise Fessant, Tassadit Bouadi, and Thomas Guyet. Post-hoc counterfactual generation with supervised autoencoder. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 105–114, Cham, 2021. Springer International Publishing.
- [GG21] Alex Gramegna and Paolo Giudici. SHAP and LIME: An evaluation of discriminative power in credit risk. *Frontiers in Artificial Intelligence*, 4, 2021.
- [GPAM⁺14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Y. Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 3, 2014.
- [Gui22] Riccardo Guidotti. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, 2022.
- [HAD21] Sebastien Haneuse, David Arterburn, and Michael J. Daniels. Assessing Missing Data Assumptions in EHR-Based Studies: A Complex and Underappreciated Task. *JAMA Network Open*, 4(2):e210184–e210184, 2021.
- [HC20] Shu-Fen Huang and Ching-Hsue Cheng. A safe-region imputation method for handling medical data with missing values. *Symmetry*, 12(11), 2020.
- [HEH21] Mahmood Shakir Hammoodi, Hasanain Ali Al Essa, and Wial Abbas Hanon. The Waikato Open Source Frameworks (WEKA and MOA) for Machine Learning Techniques. *Journal of Physics: Conference Series*, 1804(1):012133, 2021.
- [HFH⁺09] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software:

-
- an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.
- [HFL⁺21] James Hinns, Xiuyi Fan, Siyuan Liu, Veera Raghava Reddy Kovvuri, Mehmet Orcun Yalcin, and Markus Roggenbach. An initial study of machine learning underspecification using feature attribution explainable AI algorithms: A COVID-19 virus transmission case study. In *PRICAI 2021: Trends in Artificial Intelligence*, pages 323–335. Springer International Publishing, 2021.
- [Hin07] Geoffrey E. Hinton. Learning multiple layers of representation. *Trends in Cognitive Sciences*, 11(10):428–434, 2007.
- [Hin22] Geoffrey Hinton. The forward-forward algorithm: Some preliminary investigations. *arXiv*, 2022.
- [HLA⁺21] Ramin Hasani, Mathias Lechner, Alexander Amini, Daniela Rus, and Radu Grosu. Liquid time-constant networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):7657–7666, 2021.
- [Höf05] Michael Höfler. Causal inference based on counterfactuals. *BMC Medical Research Methodology*, 5(1), 2005.
- [HOT06] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.
- [HSP⁺19] Harshad Hegde, Neel Shimpi, Alokshagar Panny, Ingrid Glurich, Pamela Christie, and Amit Acharya. Mice vs ppca: Missing data imputation in healthcare. *Informatics in Medicine Unlocked*, 17:100275, 2019.
- [JAB21] Sebastian Jäger, Arndt Allhorn, and Felix Bießmann. A benchmark for data imputation methods. *Frontiers in Big Data*, 4, 2021.
- [JCL⁺22] Daniel Jarrett, Bogdan Cebere, Tennison Liu, Alicia Curth, and Mihaela van der Schaar. Hyperimpute: Generalized iterative imputation with automatic model selection. *International Conference on Machine Learning (ICML)*, 2022.
- [JGVM⁺20] Tom Jansen, Gijs Geleijnse, Marissa Van Maaren, Mathijs P Hendriks, Annette Ten Teije, and Arturo Moncada-Torres. Machine learning explainability in breast cancer survival. *Stud. Health Technol. Inform.*, 270:307–311, 2020.

- [JSL21] Joseph D. Janizek, Pascal Sturmfels, and Su-In Lee. Explaining explanations: Axiomatic feature interactions for deep networks. *Journal of Machine Learning Research*, 22(1), 2021.
- [Kad15] Purna Kadel. Role of thinking in learning. *Journal of NELTA Surkhet*, 4, 2015.
- [KED⁺21] Marcin Kapcia, Hassan Eshkiki, Jamie Duell, Xiuyi Fan, Shangming Zhou, and Benjamin Mora. Exmed: An ai tool for experimenting explainable ai techniques on medical data analytics. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 841–845, 2021.
- [KH20] Shahidul Islam Khan and Abu Sayed Md. Latiful Hoque. SICE: an improved missing data imputation technique. *J. Big Data*, 7(1):37, 2020.
- [KLS⁺22] Veera Raghava Reddy Kovvuri, Siyuan Liu, Monika Seisenberger, Xiuyi Fan, Berndt Müller, and Hsuan Fu. On understanding the influence of controllable factors with a feature attribution algorithm: a medical case study. In *2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pages 1–8, 2022.
- [KMMTS21] Ramaravind Kommiya Mothilal, Divyat Mahajan, Chenhao Tan, and Amit Sharma. Towards unifying feature attribution and counterfactual explanations: Different means to the same end. *AAAI/ACM Conference on AI, Ethics, and Society*, pages 652–663, 2021.
- [KTKA20] Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Hiroki Arimura. Dace: Distribution-aware counterfactual explanation by mixed-integer linear optimization. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 2855–2862. International Joint Conferences on Artificial Intelligence Organization, 2020.
- [KVA⁺21] Andrei Kapishnikov, Subhashini Venugopalan, Besim Avcı, Ben Wedin, Michael Terry, and Tolga Bolukbasi. Guided integrated gradients: an adaptive path method for removing noise. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5048–5056, Los Alamitos, CA, USA, 2021. IEEE Computer Society.

-
- [LAHYB15] Hassan Lemjabbar-Alaoui, Omer Hassan, Yi-Wei Yang, and Petra Buchanana. Lung cancer: biology and treatment options. *Biochim Biophys Acta*, 1856(2):189–210, 2015.
- [LAT96] Ahmad Lotfi, Hans C. Andersen, and Ah Chung Tsoi. Interpretation preservation of adaptive fuzzy inference systems. *Int. J. Approx. Reason.*, 15:379–394, 1996.
- [LGZ⁺21] Xiao Luo, Priyanka Gandhi, Zuoyi Zhang, Wei Shao, Zhi Han, Vasu Chandrasekaran, Vladimir Turzhitsky, Vishal Bali, Anna R. Roberts, Megan Metzger, Jarod Baker, Carmen La Rosa, Jessica Weaver, Paul Dexter, and Kun Huang. Applying interpretable deep learning models to identify chronic cough patients using EHR data. *Computer Methods and Programs in Biomedicine*, 210:106395, 2021.
- [LHZL20] Juan Li, Jin Huang, Lanbo Zheng, and Xia Li. Application of artificial intelligence in diabetes education and management: Present status and promising prospect. *Frontiers in Public Health*, 8:173, 2020.
- [LKO⁺20] Simon Meyer Lauritsen, Mads Kristensen, Mathias Vassard Olsen, Morten Skaarup Larsen, Katrine Meyer Lauritsen, Marianne Johansson Jørgensen, Jeppe Lange, and Bo Thiesson. Explainable artificial intelligence model to predict acute critical illness from electronic health records. *Nature Communications*, 11(1), 2020.
- [LL17] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [LL20] Yulong Lu and Jianfeng Lu. A universal approximation theorem of deep neural networks for expressing probability distributions. *arXiv*, 2020.
- [LLY⁺23] Mingxuan Liu, Siqi Li, Han Yuan, Marcus Eng Hock Ong, Yilin Ning, Feng Xie, Seyed Ehsan Saffari, Yuqing Shang, Victor Volovici, Bibhas Chakraborty, and Nan Liu. Handling missing values in healthcare data: A systematic review of deep learning-based imputation techniques. *Artificial Intelligence in Medicine*, 142:102587, 2023.

- [LOHdR19] Ana Lucic, Harrie Oosterhuis, Hinda Haned, and M. de Rijke. Focus: Flexible optimizable counterfactual explanations for tree ensembles. In *AAAI*, 2019.
- [LOS⁺22] Hui Wen Loh, Chui Ping Ooi, Silvia Seoni, Prabal Datta Barua, Filippo Molinari, and U Rajendra Acharya. Application of explainable artificial intelligence for healthcare: A systematic review of the last decade (2011-2022). *Computer Methods and Programs in Biomedicine*, 226:107161, 2022.
- [LSG⁺19] Jean-Baptiste Lamy, Boomadevi Sekar, Gilles Guezennec, Jacques Bouaud, and Brigitte Séroussi. Explainable artificial intelligence for breast cancer: A visual case-based reasoning approach. *Artificial Intelligence in Medicine*, 94:42–53, 2019.
- [MC21] Xin Man and Ernest P. Chan. The best way to select features? comparing mda, lime, and shap. *The Journal of Financial Data Science*, 3(1):127–139, 2021.
- [MCR⁺23] Abhishek Madaan, Tanya Chowdhury, Neha Rana, James Allan, and Tanmoy Chakraborty. Uncertainty in additive feature attribution methods. *arXiv*, 2023.
- [Mei12] Thorsten Meinl. What’s new in KNIME? *Journal of Cheminformatics*, 4(S1), 2012.
- [MGM⁺22] Jing Ma, Ruocheng Guo, Saumitra Mishra, Aidong Zhang, and Jundong Li. Clear: Generative counterfactual explanations on graphs. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 25895–25907. Curran Associates, Inc., 2022.
- [MHT10] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11(80):2287–2322, 2010.
- [Mil19] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267:1–38, 2019.
- [MLM04] William Fisher M. Lent and Michael Mancuso. An explainable artificial intelligence system for small-unit tactical behavior. *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence*, 2004.

-
- [Mol19] Christoph Molnar. *Interpretable Machine Learning*. Leanpub, 2019. <https://christophm.github.io/interpretable-ml-book/>.
- [MP99] Vitaly Maiorov and Allan Pinkus. Lower bounds for approximation by mlp neural networks. *Neurocomputing*, 25(1):81–91, 1999.
- [MQS⁺20] Pawel Mroz, Alexandre Quemy, Mateusz Slazynski, Krzysztof Kluza, and Pawel Jemiolo. Gbex - towards graph-based explanations. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 112–117, 2020.
- [MST20] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, FAT* '20*, pages 607–617, New York, NY, USA, 2020. Association for Computing Machinery.
- [MTvMH⁺21] Arturo Moncada-Torres, Marissa C. van Maaren, Mathijs P. Hendriks, Sabine Siesling, and Gijs Geleijnse. Explainable machine learning can outperform cox regression predictions and provide insights in breast cancer survival. *Scientific Reports*, 11(1), 2021.
- [MUKK20] Anna Meldo, Lev Utkin, Maxim Kovalev, and Ernest Kasimov. The natural language explanation algorithms for the lung cancer computer-aided diagnosis system. *Artificial Intelligence in Medicine*, 108:101952, 2020.
- [NCC⁺21] Sam Nguyen, Ryan Chan, Jose Cadena, Braden Soper, Paul Kiszka, Lucas Womack, Joan Duggan, Steven Haller, Jennifer Hanrahan, David Kennedy, Deepa Mukundan, and Priyadip Ray. Budget constrained machine learning for early prediction of adverse outcomes for COVID-19 patients. *Scientific Reports 11*, 2021.
- [NJKC19] Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. Interpretml: A unified framework for machine learning interpretability. *arXiv*, abs/1909.09223, 2019.
- [NK99] Detlef Nauck and Rudolf Kruse. Obtaining interpretable fuzzy classification rules from medical data. *Artificial intelligence in medicine*, 16 2:149–69, 1999.
- [NSBL21] Michael Neely, Stefan Schouten, Maurits Bleeker, and Ana Lucic. Order in the court: Explainable ai methods prone to disagreement. *arXiv*, 2021.

- [NW23] Ashley I. Naimi and Brian W. Whitcomb. Defining and Identifying Average Treatment Effects. *American Journal of Epidemiology*, 192(5):685–687, 2023.
- [ON15] Keiron O’Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv*, abs/1511.08458, 2015.
- [Pel14] Marcello Pelillo. Alhazen and the nearest neighbor rule. *Pattern Recognition Letters*, 38:34–37, 2014.
- [PGS⁺20] Mattia Prosperi, Yi Guo, Matt Sperrin, James S. Koopman, Jae S. Min, Xing He, Shannan Rich, Mo Wang, Iain E. Buchan, and Jiang Bian. Causal inference and counterfactual prediction in machine learning for actionable healthcare. *Nature Machine Intelligence*, 2(7):369–375, 2020.
- [Pin99] Allan Pinkus. Approximation theory of the mlp model in neural networks. *Acta Numerica*, 8:143–195, 1999.
- [PLZ21] Deng Pan, Xin Li, and Dongxiao Zhu. Explaining deep neural network models with adversarial gradient integration. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2876–2883. International Joint Conferences on Artificial Intelligence Organization, 2021. Main Track.
- [PMT18] Gregory Plumb, Denali Molitor, and Ameet Talwalkar. Model agnostic supervised local explanations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS’18*, pages 2520–2529, Red Hook, NY, USA, 2018. Curran Associates Inc.
- [Pri23] Simon J.D. Prince. *Understanding Deep Learning*. MIT Press, 2023.
- [PRMT20] Richard Delwin Myloth Pavan Rajkumar Magesh and Rijo Jackson Tom. An explainable machine learning model for early detection of parkinson’s disease using lime on datscan imagery. *Computers in Biology and Medicine*, 126:104041, 2020.
- [PSK22] Eva I. Prakash, Avanti Shrikumar, and Anshul Kundaje. Towards More Realistic Simulated Datasets for Benchmarking Deep Learning Models in Regulatory Genomics. In David A. Knowles, Sara Mostafavi, and Su-In Lee, editors, *Proceedings of the 16th Machine Learning in Computational Biology meeting*, volume 165 of

-
- Proceedings of Machine Learning Research*, pages 58–77. PMLR, 2022.
- [PSSR⁺20] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. Face: Feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, AIES '20*, pages 344–350, New York, NY, USA, 2020. Association for Computing Machinery.
- [Pub] Public Health England’s National Cancer Registration and Analysis Service. Simulacrum. <https://simulacrum.healthdatainsight.org.uk/>, Last accessed on 2019-6-30.
- [PYT23] Nico Potyka, Xiang Yin, and Francesca Toni. Explaining random forests using bipolar argumentation and markov networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8):9453–9460, 2023.
- [PZZ⁺21] Junfeng Peng, Kaiqiang Zou, Mi Zhou, Yi Teng, Xiongyong Zhu, Feifei Zhang, and Jun Xu. An explainable artificial intelligence framework for the deterioration risk prediction of hepatitis patients. *Journal of Medical Systems*, 45(5), 2021.
- [RHW86] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.
- [RLS15] Panteha Hayati Rezvan, Katherine J Lee, and Julie A Simpson. The rise of multiple imputation: a review of the reporting and implementation of the method in medical research. *BMC Medical Research Methodology*, 15(1), 2015.
- [Ros58] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.
- [RS22] Niloofar Ranjbar and Reza Safabakhsh. Using decision tree as local interpretable model in autoencoder-based lime. In *2022 27th International Computer Conference, Computer Society of Iran (CSICC)*, pages 1–7, 2022.
- [RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ”why should i trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on*

- Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.
- [RSG18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018.
- [RTVA22] Jaber Rad, Karthik K. Tennankore, Amanda Vinson, and Syed Sibte Raza Abidi. Extracting surrogate decision trees from black-box models to explain the temporal importance of clinical features in predicting kidney graft survival. In Martin Michalowski, Syed Sibte Raza Abidi, and Samina Abidi, editors, *Artificial Intelligence in Medicine*, pages 88–98, Cham, 2022. Springer International Publishing.
- [SACF⁺12] Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, Jean-Yves Tinevez, Daniel James White, Volker Hartenstein, Kevin Eliceiri, Pavel Tomancak, and Albert Cardona. Fiji: an open-source platform for biological-image analysis. *Nature Methods*, 9(7):676–682, 2012.
- [Sag22] Abhinav Sagar. Uncertainty quantification using variational inference for biomedical image segmentation. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pages 44–51, Los Alamitos, CA, USA, 2022. IEEE Computer Society.
- [SB11] Daniel J. Stekhoven and Peter Bühlmann. MissForest-non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2011.
- [SCD⁺17] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [Sch90] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.
- [SFS⁺21] Hyuna Sung, Jacques Ferlay, Rebecca L. Siegel, Mathieu Laveranne, Isabelle Soerjomataram, Ahmedin Jemal, and Freddie Bray.

-
- Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, pages 209–249, 2021.
- [SG71] Naresh K. Sinha and Michael P. Griscik. A stochastic approximation method. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1(4):338–344, 1971.
- [SGK17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 3145–3153. JMLR.org, 2017.
- [SHSL21] Dylan Slack, Anna Hilgard, Sameer Singh, and Himabindu Lakkaraju. Reliable post hoc explanations: Modeling uncertainty in explainability. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 9391–9404. Curran Associates, Inc., 2021.
- [ŠK14] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowl. Inf. Syst.*, 41(3):647–665, 2014.
- [SKW⁺21] Salih Sarp, Murat Kuzlu, Emmanuel Wilson, Umit Cali, and Ozgur Guler. The enlightening role of explainable artificial intelligence in chronic wound classification. *Electronics*, 10(12), 2021.
- [SN20] Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9269–9278. PMLR, 2020.
- [SP17] Andrew D. Selbst and Julia Powles. Meaningful information and the right to explanation. *International Data Privacy Law*, 7(4):233–242, 2017.
- [SP22] Ruey-Kai Sheu and Mayuresh Sunil Pardeshi. A survey on medical explainable ai (xai): Recent progress, explainability approach, human interaction and scoring system. *Sensors*, 22(20), 2022.
- [SSV21] Ilija Simic, Vedran Sabol, and Eduardo E. Veas. XAI methods for neural time series classification: A brief review. *CoRR*, abs/2108.08009, 2021.

- [STY17] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML'17: Proceedings of the 34th International Conference on Machine Learning*, pages 3319–3328. JMLR.org, 2017.
- [SZLF19] Sheng Shi, Xinfeng Zhang, Haisheng Li, and Wei Fan. Explaining the predictions of any image classifier via decision trees. *arXiv*, abs/1911.01058, 2019.
- [SZX⁺20] Yucheng Shu, Jing Zhang, Bin Xiao, Xiao Luan, Linghui Liu, and Chunlong Hu. Aft-net: Active fusion-transduction for multi-stream medical image segmentation. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 753–760, 2020.
- [TAGD98] Alan B. Tickle, R. Andrews, M. Golea, and J. Diederich. The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural networks. *IEEE transactions on neural networks*, 96:1057–68, 1998.
- [TBL⁺22] Owen Trigueros, Alberto Blanco, Nuria Lebeña, Arantza Casillas, and Alicia Pérez. Explainable ICD multi-label classification of EHRs in spanish with convolutional attention. *International Journal of Medical Informatics*, 157:104615, 2022.
- [TEN19] JING TENG. Seer breast cancer data, 2019.
- [TG21] Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE transactions on neural networks and learning systems*, 32(11):4793–4813, 2021.
- [TJMG19] Sana Tonekaboni, Shalmali Joshi, Melissa D. McCradden, and Anna Goldenberg. What clinicians want: Contextualizing explainable machine learning for clinical end use. In Finale Doshi-Velez, Jim Fackler, Ken Jung, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 4th Machine Learning for Healthcare Conference*, volume 106 of *Proceedings of Machine Learning Research*, pages 359–380, Ann Arbor, Michigan, 2019. PMLR.
- [TMG23] Hasan Torabi, Seyedeh Leili Mirtaheri, and Sergio Greco. Practical autoencoder based anomaly detection by using vector reconstruction error. *Cybersecurity*, 6(1), 2023.

-
- [TRO22] Yue Ting Tang and Roman Romero-Ortuno. Using explainable ai (xai) for the prediction of falls in the older population. *Algorithms*, 15(10), 2022.
- [Tur50] Alan M. Turing. I-COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460, 1950.
- [TVA⁺22] Lucas M. Thimoteo, Marley M. Vellasco, Jorge Amaral, Karla Figueiredo, Cátia Lie Yokoyama, and Erito Marques. Explainable artificial intelligence for COVID-19 diagnosis through blood test variables. *Journal of Control, Automation and Electrical Systems*, 33(2):625–644, 2022.
- [unk20] LightSide Researcher’s Workbench, 2020.
- [VABH22] Manjunatha Veerappa, Mathias Anneken, Nadia Burkart, and Marco F. Huber. Validation of xai explanations for multivariate time series classification in the maritime domain. *Journal of Computational Science*, 58:101539, 2022.
- [VBGO11] Stef Van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, 45:1–67, 2011.
- [VDH20] Sahil Verma, John P. Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: A review. volume abs/2010.10596, 2020.
- [WCNK13] Brian J. Wells, Kevin M. Chagin, Amy S. Nowacki, and Michael W. Kattan. Strategies for handling missing data in electronic health record derived data. *EGEMS (Wash DC)*, 1(3):1035, 2013.
- [WGGP23] Weronika Wrazen, Kordian Gontarska, Felix Grzelka, and Andreas Polze. Explainable AI for medical event prediction for heart failure patients. In Jose M. Juarez, Mar Marcos, Gregor Stiglic, and Allan Tucker, editors, *Artificial Intelligence in Medicine*, pages 97–107, Cham, 2023. Springer Nature Switzerland.
- [WJJ16] Gavitt A. Woodard, Kirk D. Jones, and David M. Jablons. Lung cancer staging and prognosis. *Cancer Treat. Res.*, 170:47–75, 2016.
- [WKSP22] Wenlong Wu, James M. Keller, Marjorie Skubic, and Mihail Popescu. Explainable ai for early detection of health changes via streaming clustering. In *2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6, 2022.

- [WLB⁺20] Joseph T Wu, Kathy Leung, Mary Bushman, Nishant Kishore, Rene Niehus, Pablo M de Salazar, Benjamin J Cowling, Marc Lipsitch, and Gabriel M Leung. Estimating clinical severity of covid-19 from the transmission dynamics in wuhan, china. *Nature Medicine*, pages 1–5, 2020.
- [WLW20] Linda Wang, Zhong Q. Lin, and Alexander Wong. COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. *Sci Rep*, 10(1):19549, 2020.
- [WMR18] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law & Technology*, 2018.
- [WS95] Mangasarian Olvi Street Nick Wolberg, William and W. Street. Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository, 1995. DOI: <https://doi.org/10.24432/C5DW2B>.
- [YAWM23] Peiyu Yang, Naveed Akhtar, Zeyi Wen, and Ajmal Mian. Local path integration for attribution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(3):3173–3180, 2023.
- [YFL21] Orcun Yalcin, Xiuyi Fan, and Siyuan Liu. Evaluating the correctness of explainable AI algorithms for classification. *arXiv*, 2021.
- [YLH15] Yoshua Bengio Yann LeCun and Geoffrey Hinton. Deep learning. *Nature*, 521:436–444, 2015.
- [YLXH22] Wenzhuo Yang, Jia Li, Caiming Xiong, and Steven Hoi. Mace: An efficient model-agnostic framework for counterfactual explanation. *arXiv*, 2022.
- [YPT23] Xiang Yin, Nico Potyka, and Francesca Toni. Argument attribution explanations in quantitative bipolar argumentation frameworks (technical report). *European Conference on Artificial Intelligence (ECAI)*, 2023.
- [YRC⁺20] Tae Keun Yoo, Ik Hee Ryu, Hannuy Choi, Jin Kuk Kim, In Sik Lee, Jung Sub Kim, Geunyoung Lee, and Tyler Hyungtaek Rim. Explainable Machine Learning Approach as a Tool to Understand Factors Used to Select the Refractive Surgery Technique on the

-
- Expert Level. *Translational Vision Science & Technology*, 9(2):8, 2020.
- [YWB23] Ruo Yang, Binghui Wang, and Mustafa Bilgic. IDGI: A framework to eliminate explanation noise from integrated gradients. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [YWG98] John Yen, Liang Wang, and Charles W. Gillespie. Improving the interpretability of tsf fuzzy models by combining global learning and local learning. *IEEE Trans. Fuzzy Syst.*, 6:530–537, 1998.
- [ZG08] Shang-Ming Zhou and John Q. Gan. Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modelling. *Fuzzy Sets Syst.*, 159:3091–3131, 2008.
- [ZG09] Shang-Ming Zhou and John Q. Gan. Extracting takagi-sugeno fuzzy rules with interpretable submodels via regularization of linguistic modifiers. *IEEE Transactions on Knowledge and Data Engineering*, 21:1191–1204, 2009.
- [ZHH⁺21] Xingyu Zhao, Wei Huang, Xiaowei Huang, Valentin Robu, and David Flynn. Baylime: Bayesian local interpretable model-agnostic explanations. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 887–896. PMLR, 2021.
- [ZHW21] Zhengze Zhou, Giles Hooker, and Fei Wang. S-LIME: Stabilized-LIME for Model Explanation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD '21*, pages 2429–2438, New York, NY, USA, 2021. Association for Computing Machinery.
- [ZK21] Muhammad Rehman Zafar and Naimul Khan. Deterministic local interpretable model-agnostic explanations for stable explainability. *Machine Learning and Knowledge Extraction*, 3(3):525–541, 2021.
- [ZLF⁺23] Gaoxia Zhu, Mengyu Lim, Xiuyi Fan, Chenyu Hou, Guangji Yuan, and Atiqah Azhari. The difficulty of collaborative interdisciplinary learning and its impact on undergraduates' epistemic emotions and problem solving. In *International Society of the Learning Sciences (ISLS)*. ISLS Annual Meeting, 2023.

- [ZWL22] Yiming Zhang, Ying Weng, and Jonathan Lund. Applications of explainable artificial intelligence in diagnosis and surgery. *Diagnostics*, 12(2), 2022.