Interpretable Machine Learning Models for Assisting Clinicians in the Analysis of Physiological Data

P. Nisha, Urja Pawar and Ruairi O'Reilly

Cork Institute of Technology, Ireland, p.nisha@mycit.ie, urja.pawar@mycit.ie, ruairi.oreilly@cit.ie, www.cs.cit.ie

Abstract. The analysis of physiological data plays a significant role in medical diagnostics. While state-of-the-art machine learning models demonstrate high levels of performance in classifying physiological data clinicians are slow to adopt them. A contributing factor to the slow rate of adoption is the "black-box" nature of the underlying model whereby the clinician is presented with a prediction result, but the rationale for that result omitted or not presented in an interpretable manner.

This gives rise to the need for interpretable machine learning models such that clinicians can verify, and rationalise, the predictions made by a model. If a clinician understands why a model makes a prediction, they will be more inclined to accept a models assistance in analysing physiological data.

This paper discusses some of the latest findings in interpretable machine learning. Thereafter, based on these findings, three models are selected and implemented to analyse ECG data that are both accurate and exhibit a high level of interpretability.

Keywords: Interpretable Machine Learning, Decision Trees, Random Forest, Feature Engineering, ECG, Medical Diagnostics

1 Introduction

In recent years there has been an increased push towards utilising machine learning as part of healthcare solutions [2] but its adoption has been slow. A major obstacle to the adoption of machine learning in the clinical decision-making process is the black-box nature of the algorithms upon which a machine learning model may rely upon [3].

A clinician is hesitant to treat a patient's diagnosis as an input-output process whereby they feed patient data into a system, and a diagnosis is returned without any insight as to how that diagnosis was arrived at. Given the significance of the diagnostic decision-making process, it is understandable that a clinician would not trust something they do not understand, particularly as patient care needs explanation and the algorithm or models these systems are built upon do not always provide clarification or a rationale for the prediction made [3].

As such traditional performance metrics (accuracy, precision, f1-Score, recall) should not be the only consideration when designing a model intended to assist clinicians in diagnostics, where data can be heterogeneous and where edge cases cannot be easily identified beforehand. In the clinical world, the stakes are much higher and as such, the performance of a model should be robust across the target patient population, its implementation ought to ensure proper use, and its analysis or predictions should provide context that aids interpretability [6].

In healthcare, interpretability is of utmost importance as the prediction of a model must be backed by plausible and explainable reasoning as a patients life stands at risk. However, studies have shown that there is a trade-off between interpretability and performance for machine learning models. Incorporating interpretability as a performance metric for a machine learning could be a significant step in addressing this problem. Providing clinicians with an insight as to the suitability of a model to their needs, assisting in predictions and leaving room for the domain experts to debug and improve the underlying machine learning models being utilised.

2 Related Work

In the last few years there has been research carried out to come up with a model that is highly accurate as well as interpretable e.g., GA2M [4], falling rule lists [11], ensemble decision trees [12] and model distillation [10]. Although these models have exhibited good performance, they have not yet been used in the health care industry due to the rarity of their application [3]. A number of interpretable models are reviewed to form an understanding of the characteristics that make a model interpretable.

2.1 Decision Trees

A decision tree is a machine learning model that distributes data into subsets and predicts outcomes based on decision rules (if-then-else rules). The partitioning of data begins with a binary split and continues until it cannot be split any further. Different branches of different lengths are formed.

A decision tree captures the training data in the smallest possible tree. This is done in to simplify the explanation of the problem. Smaller decision trees provide faster decisions and are easier to understand.

The reasoning approach behind the decision tree readily approachable when browsing through the tree. This approachability makes the decision trees an interpretable machine learning model. The attributes that contribute towards the accuracy and decision making are only included in the rules of the decision tree. The other attributes are all ignored. This reduced focus helps to provide useful information about the features. A decision tree has low bias and high variance. Combining several decision trees will decrease variance and maintain low bias. The technique of combining multiple decision trees (models typically) is known as an ensemble method. Ensemble methods perform better than a single decision tree and provide accurate predictions.

Ensemble Decision Trees Ensemble decision trees can be formed using two techniques:

Bagging — Bagging is a general technique for combining the predictions of many models. This technique uses randomly sampled training sets to train each tree. An ensemble of different models is obtained as a result.

Bootstrap Aggregation — Boosting trains the models iteratively such that training of any model or tree at the current step depends on the previously trained model or tree. Each new model or tree gives more importance to train the observations that were badly handled by a previous model. This helps in obtaining a strong predictor with low bias.

For a classification problem, the aggregated prediction will be generated by a majority vote from all the different models or trees. Random Forests are an extension of bagging. Random features are selected to build a tree instead of using all the features. The feature that gives the best split is used to split the node of the tree. A group of random decision trees constitute a random forest.

In [12], an ensemble of decision trees was used to classify ECG data, and an accuracy of 90.4% was obtained. The ECG signal of frequency 0.5Hz 40Hz was only considered for the implementation as it was considered as the most important part of the signal. Apart from this frequency range, everything else was considered to fall under noise and was eliminated. The ensemble decision tree was generated using the Bootstrap Aggregation method.

2.2 GA2MS

Generalised Additive Model (GAM) is an additive modelling technique which can be used to capture nonlinearities in the data. The contribution of each independent variable to the prediction of GAM model is clearly stated, making it highly interpretable. GA2MS is an extended form of the GAM model. GA2MS model is obtained when pairwise interactions (considering the interactions between two different features) are added to the GAM model.

In paper [4], two case studies were presented where generalised additive models with pairwise interactions (GA2MS) were applied to healthcare problems, and state-of-the-art accuracies were obtained.

The GA2MS model in [4] was used to detect the probability of death due to pneumonia in patients. This was done to ensure the patients with high risk can be immediately attended. Every feature or term in the model returned a risk score. A risk score above zero indicated a higher risk, and below zero indicated a lower risk. All the risk scores for a particular patient were added together (to a baseline risk). The aggregate risk was converted to a probability score. Features were selected on the basis of the risk score. Selecting the most critical features made the patient's status more understandable (by the clinical expert) and contributed to the interpretability of the model.

Spline interpolation was used to overcome irregularities in the data. The model was trained with spline, which considers each point in the data and represents it to form a smooth curve; this also helps tackle the overfitting of the model. The GA2MS model detected patterns in the data which were missed before by other models. The feature selection technique of this model provided an accuracy of 85.7%. Most importantly, this paper demonstrated how predictions made by the model could be explained for an individual patient by considering only the most critical features (depending on the risk score)[4].

2.3 Model Distillation

Model distillation is a technique which makes use of two different machine learning models, namely - a student model and a teacher model. A teacher model is a complex machine learning model like neural networks, whereas a student model is an interpretable model like decision trees. The main aim of model distillation is to transfer the generalisations or learnings of the complex teacher model to the interpretable student model. This way, the reasons behind the predictions made by the complex black-box models can be made easily understandable by the student model.

The complex teacher model is well trained and regularised (to avoid overfitting) to perform well on unseen data. The training knowledge obtained by the teacher model is then distilled to the student model.

In this paper [10], a transparent model distillation technique was used to understand and detect the bias in black-box models. This was achieved by training a transparent student model to mimic the black-box model and then comparing the transparent mimic model to a transparent model using the same features on "true" outcomes instead of the labels predicted by the black-box model. Difference between the transparent mimic and true-label model portray how a black-box model predicts and how a model trained on "true" outcomes highlights potential bias in the black-box model.

Interpretability mainly means transparency of the features used and easily understandable algorithm, and it differs from person to person and the use case. In a broader sense, interpretability can be described as transparency of the machine learning model, i.e., the algorithm, features, parameters and the model should be comprehensible by the end-user [8].

3 Methodology

Studies have shown that there is a trade-off between interpretability and performance in machine learning models [5]. This means that while models like decision trees and regression models are highly interpretable, they are less accurate when compared to less interpretable models like neural networks and other deep learning models. Thus, one has to compromise on either of the two attributes (interpretability and accuracy).

Generalized Linear Model was selected because it is fast, computationally inexpensive and interpretable in nature. A Decision Tree model was selected as it requires very little data preparation and is very intuitive and easily explainable. Thirdly, Random Forests was selected as it is one of the most accurate learning algorithms and can handle data imbalance and variance in data implicitly.

The ECG dataset was preprocessed in a way that removed all noise such that more accurate results could be obtained. After having preprocessed the signals in the dataset, it was trained and validated on three different interpretable models Generalized Linear Model, Decision Trees and Random Forests. Since Random Forest is not interpretable by its nature, Graphviz was used to generate the tree structure to aid interpretation as to why a prediction was made.

While analysing ECG data, making an accurate prediction only partially solves the problem. The reason why it is considered as an accurate prediction or why a certain prediction was made adds value to the analysis.

Section 3.1 contains the details about the dataset used. Section 3.2 discusses the various pre-processing techniques employed. Section 3.3 discusses the feature engineering techniques and section 3.4 provides a detailed description of each of the models used.

3.1 Data

The dataset being used is the Physionet MIT-BIH dataset [7] available from Kaggle [1] and originally presented in [9]. The dataset is already normalized, and the R-R interval is extracted by applying a threshold of 0.9 on the normalized value. Since the signals before normalization were of different morphologies, the R-R beats that were extracted were padded with zeroes to achieve an equal length. The R-R beats present in the dataset are all of equal length. There are 5 classes in the dataset and the normal class is heavily oversampled that is, a disproportionately high sample of normal class is present in the dataset. There are total of 5 classes with each corresponding a particular heart condition as denoted in Table 1.

Class	Heart condition			
Ν	Normal, Left/Right bundle branch block, Atrial escape, Nodal escape			
S	Atrial premature, Aberrant atrial premature, Nodal premature, Supra-ventricular premature			
V	Premature ventricular contraction, Ventricular escape			
F	Fusion of ventricular and normal			
Q	Paced, Fusion of paced and normal, Unclassifiable			
Table 1. The different classes in the MIT-BIH dataset.				

While recording an ECG signal, it can be contaminated by a variety of interfering signals that are classified as noise. The source of this noise can be a patients movement, respiration, surrounding disturbances, muscle movements etc. Noise degrades the signal quality which leads to misinterpretation. Thus, it is necessary to de-noise the signal before it can be used in diagnosis which is discussed in section 3.2.

3.2 Preprocessing

Signal preprocessing is used to eliminate the noise from the signal and is an important process in increasing the performance of heart beat classification. The following are the signal preprocessing techniques carried out to the data.

- 1. Differencing
- 2. Normalising
- 3. Smoothing

Differencing – A stationary time series signal is one in which the components (mean, variance and co-variance) do not vary with time (independent of time). The underlying assumption in signal preprocessing techniques is that the signal is stationary. It is simpler to analyse stationary signals as the complexity of the time component is not taken into account.

Non-stationary time series data can be transformed into stationary time series data by applying a preprocessing technique called Differencing. Differencing is carried out by subtracting the previous observation (data point lying in front in the series) from the current one. It can help stabilize the mean of time-series data by removing or eliminating the effects of trend and seasonality.

Differencing is carried out by using the Pandas diff() function. The function is useful as it maintains the date and time information and satisfies the underlying assumption when processing the signal data. After differencing the data in the first column it is filled with NaN (Not a number) values as there are no values to the left of it to find the difference. To address this, Pandas fillna() function is used to fill in the NaN values, the backward filling method is used on the columns (i.e. using the next valid data observation to fill the gap in the dataset).

Normalising the signal — A time series signal is normalised to rescale the data so that all distributions are alike and relevant comparison can be done. Normalising also reduces noise in the signal. The data was normalized by dividing the column value with the maximum value of the column. This operation transform the data to the same scale.

Smoothing using Moving Average Function — Smoothing a signal reduces the noise. While performing smoothing data points of the signal are adjusted. The individual data points that are higher than the immediately adjacent datapoints (assumed to be due to noise), are lowered. In case the individual data points are lower than their neighbouring data points, they are increased. This results in eliminating distortion and a smoother signal is obtained.

The moving average technique was the smoothing technique used, it has an underlying assumption that independent noise will not change the signal. According to this assumption, if a few data points are averaged the noise can be eliminated. Moving average makes use of a window, which is slid across the whole time series data to calculate the average values. It transforms the old time series data to a new time series data after averaging the values. The rolling function available in pandas was used as the moving average function for this work. The rolling function automatically groups observations into a window where a window size can be specified, and a trailing window can be created. For the purpose of this work, the window size of 7 samples was taken. Trailing window makes use of historical observations and are used for time-series data. After the trailing window is created it takes the max value and changes the dataset. Rolling window operations are an important transformation that can be done on a dataset containing time-series data. The transformed data retains the same frequency as the original data.

3.3 Feature Engineering

Feature engineering has been used to create a much bigger feature space and gather more information from the data. More features were manually created from previously existing ones, to make the predictive model better. It also helped to increase the model accuracy on unseen data.

The frequency of the time series signal has been changed in this research by decreasing the frequency of the signal (Downsampling). Downsampling is a feature engineering technique which helps reduce the signal processing time.

It is done by using scipy signal decimate function which uses the anti-aliasing filter. The anti-aliasing filter is a low pass filter which only lets low frequencies pass through and attenuates higher frequencies. The dataset had a sampling frequency of 125Hz, it was downsampled by a factor of 5 such that the first element and every fifth element then onwards was persisted.

3.4 Models

Generalised Linear Model A linear regression model predicts the target as the weighted aggregate of all the input features. Logistic regression is used in classification problems and it is an extension of linear regression where the model predicts probabilities based on two possible outcomes. GLM (Generalised Linear Model) is an extension to linear regression without the assumption that the outcome distributions will be gaussian in nature.

GLM calculates the expected mean of the non-gaussian outcome distribution and connects it to the weighted sum of input features by passing it through a non-linear function. GLM can be defined as a more flexible model by keeping the interpretability intact. Modelling based on the weighted sum makes the model transparent and provides an explanation as to why certain predictions are made. Not only predictions on weights, but confidence intervals of the weights themselves can be derived from analysing the contributing features. The given problem is a multiclass problem and logistic regression is used for classification. GLM has been modelled using Logistic Regression.

A solver helps to fit or train the data. Below are the different types of solvers available in Logistic Regression:

- 1. Newton-CG solver: The Newton Solver has a very fast converge range (learns much faster). It uses the principle of gradient descent with the Hessian (a squared matrix of second order partial derivatives) to achieve faster convergence.
- 2. Limited-memory-Broyden-Fletcher-Goldfarb algorithm (LBFGS solver): This solver is similar to a newton solver. The only difference being it uses an estimation to the inverse hessian matrix. This saves significant memory but a major disadvantage is that in some cases it may not converge to anything.
- 3. A Library for Large Linear Classification or (LibLinear solver): This solver is a linear classifier that makes decisions based on the linear combination of features. It performs approximate minimizations along the co-ordinate directions. The main drawback of this solver is that it does not perform well for multi-class problems.
- 4. Stochastic Average Gradient or (SAG solver): SAG solvers are best suited for large datasets with large number of features. Its memory cost is too high making it impractical most of the time.
- 5. SAGA solver: Saga solver is a variant of SAG that is suitable for very large datasets.

The multi-class parameter has two variants in logistic regression:

- 1. Multinomial: It is a classification method used for data when data has nominal or categorical dependant variables.
- 2. One vs Rest Approach (Ovr): One vs rest approach can be used to convert any problem into binary classification problem. This method trains different distinct binary classifiers, each classifier is designed to predict or recognize a particular class.

The different Class weights parameters available in logistic regression are:

- 1. Balanced: It ensures that there is a balance mix of classes by weighing the classes inversely proportional to their frequency.
- 2. None: If the class weight is specified as None then the class weights will be uniform in nature.

The parameter inverse of regularization strength is named as C. Regularization prevents over-fitting of the model. This parameter has been used in the research and has been assigned a smaller value to specify better regularization. The smaller the value of C, the stronger the regularization ensuring the model does not overfit. The dataset is heavily imbalanced and since imbalance was not addressed during the preprocessing of the data, the balanced class weight method has been used to handle this.

The selection of the parameters for the Logistic Regression model used are Newton-cg solver, OVR, C and Balanced.

Decision Trees learn through **if-then-else** decision rules making the outcome of the model interpretable and the root-cause of a prediction easy to follow. Decision tree models split the data depending on certain cut-off values in their features. Different subsets of the dataset are created when the nodes are split and an associated tree gets generated incrementally. Finally, a tree with leaf nodes and decision nodes is obtained. A decision node has two or more branches.

A leaf node represents the classification or decision. The top most decision node is called the root node and is considered as the best predictor. The interpretation of the decision tree is also very simple. The root node is the starting point and the next nodes classify all the subsets. Once the leaf node is reached the predicted output can be obtained.

The simplicity of the interpretation is often contributed to as data ends up in distinct groups making it easier to understand. Sklearn's decision tree classifier is used to model a Decision tree. The export graphviz function is used to visualize the decision tree. The graphviz function provides a detailed graph with the tree's structure containing the if-then-rules. This helps to understand why a certain decision was taken.

Random Forest Random forest is a very flexible machine learning model. It creates a forest, which is an ensemble of decision trees, trained with the bagging method. Bagging employs the idea that the combination of learning models increases the overall results. Random Forests generates multiple decision trees and merges them together to get a more accurate and stable prediction.

The importance of each feature on the prediction or outcome can be found out using sklearns feature importance. This technique is used to analyse each features importance and to predict what led to the outcome of the model. Thus, knowing the contribution of each feature turns this into a white-box model.

4 Results

Performance Measures: The performance of the classification algorithms were evaluated using five measures:

Confusion Matrix: Terms associated — True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN)

Precision — Precision of a model can be defined as the ratio of correctly predicted positive observations to the total predicted positive observations. Precision = TP/TP+FP

Recall — Recall of a model can be defined as the ratio of correctly predicted positive observations to the all observations in actual true class. Recall = TP/TP+FN

Classification accuracy — Accuracy of a model can be defined as a ratio of correctly predicted observation to the total number of observations made. Accuracy = TP+TN/TP+FP+FN+TN

F1 score — F1 score of a model can defined as the weighted average of Precision and Recall. F1 Score = $2^{*}(\text{Recall * Precision}) / (\text{Recall + Precision})$

	GLM			Random Forest			Decision Tree		
Classes	Precision	Recall	F1	Precision	Recall	$\mathbf{F1}$	Precision	Recall	F1
N	0.98	0.89	0.93	0.97	1.00	0.98	0.97	0.98	0.98
S	0.31	0.65	0.42	0.99	0.60	0.75	0.65	0.65	0.65
V	0.69	0.83	0.75	0.97	0.84	0.90	0.86	0.84	0.85
F	0.15	0.78	0.25	0.87	0.56	0.68	0.52	0.60	0.56
\mathbf{Q}	0.89	0.94	0.91	0.99	0.93	0.96	0.94	0.94	0.94
Average	0.93	0.88	0.90	0.97	0.97	0.97	0.95	0.95	0.95

Table 2. Table depicting performance results for the three models implemented: Generalized Linear Model (GLM), the Random Forest and the Decision Tree.

Model	Accuracy
GLM	88.29%
Random Forest	97%
Decision Tree	95%

Table 3. Accuracy scores of the three models evaluated.

Table 2 depicts the performance metrics of GLM model. It can be seen that the model has a high precision average of 0.93 and a recall of 0.88. The accuracy obtained by the model is 88.29%. The Random Forest model also has a high precision average of 0.97 and a recall of 0.97. The accuracy obtained by the model is 97.00% and the model could classify almost all of the classes correctly. Finally, the Decision Tree model was shown to have a high precision average of 0.95 and a recall of 0.95. The accuracy obtained by the model is 95.00%.

Table 3 denotes the accuracy scores of all the three models. Random Forest performed the best out of all the models obtaining an accuracy 97%. The model has a very high precision rate for all the classes as can be seen from the table.

4.1 Hyper Parameter Tuning of Generalized Linear Model

The Generalized Linear Model was evaluated on a variety of parameters such as solver, multi-class, regularization factor(c) and class-weight. The model was tested on four different solvers. The GLM model performed the best using the following parameters: (multi_class = ovr, solver = newton-cg, class_weight = balanced, c = 0.5). Table 4 denotes the accuracy obtained for the solvers evaluated.

Solver	Accuracy
SAG	73.41%
SAGA	82.40%
Newton-cg	88.29%
LBFGS	87.23%

Table 4. Results of the different solvers evaluated for the GLM model.

5 Conclusion

In areas such as medical diagnostics, accuracy is not the only factor that determines the performance of a machine learning model. Interpretability plays a crucial role due to its importance in understanding the rationales for a model's predictions. As such, interpretability is particularly crucial in models designed for the analysis of physiological data and clinical purposes.

In this paper, MIT-BIH dataset is used for heartbeat classification. Prior to modelling of the data, various preprocessing techniques have been used to eliminate noise in the ECG signal. The noise present in the ECG signal leads to misinterpretation. Feature engineering techniques have been employed to improve the performance of the models.

The data has then been trained on three different models namely - Generalized Linear Model, Random Forests and Decision Trees. Graphviz has been used to convert the black box Random Forest model to an interpretable model. All the three models provide comprehensible decisions of the predictions made. An accuracy of 97% on using Random Forest model was obtained. The obtained accuracy is comparable to state of the art models in ECG classification.

The models presented achieved a high level of accuracy as well as a high level of interpretability. The ability to employ techniques such as feature importance that identify the underlying features contributing towards the decision taken by a model increases the transparency in the classification process, making it more akin to a white-box model. Providing clinicians with an overview of features, and associated values, contributing to a decision enable the clinician to deduce the rationale behind a prediction. It is envisaged that this would increase clinicians' trust and confidence in a prediction and assist them in providing excellent patient care.

Acknowledgement: This material is based upon works supported by Science Foundation Ireland under Grant No. SFI CRT 18/CRT/6222

References

- Mit-bih arrhythmia database. https://www.kaggle.com/mondejar/ mitbih-database
- Ahmad, M.A., Eckert, C., Teredesai, A.: Interpretable machine learning in healthcare. In: Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics. pp. 559–560. ACM (2018)
- Ahmad, M.A., Eckert, C., Teredesai, A.: Interpretable machine learning in healthcare. In: Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics. pp. 559–560. ACM (2018)
- Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., Elhadad, N.: Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1721–1730. ACM (2015)
- Doshi-Velez, F., Kortz, M., Budish, R., Bavitz, C., Gershman, S., O'Brien, D., Schieber, S., Waldo, J., Weinberger, D., Wood, A.: Accountability of ai under the law: The role of explanation. arXiv preprint arXiv:1711.01134 (2017)

- Glass, A., McGuinness, D.L., Wolverton, M.: Toward establishing trust in adaptive agents. In: Proceedings of the 13th international conference on Intelligent user interfaces. pp. 227–236. ACM (2008)
- Goldberger AL, Amaral LAN, G.L.H.J.I.P.M.R.M.J.M.G.P.C.K.S.H.: Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. IEEE Engineering in Medicine and Biology Magazine 101(23), 215–220 (2003)
- 8. Lipton, Z.C.: The mythos of model interpretability. arXiv preprint arXiv:1606.03490 (2016)
- 9. Moody, G.B., Mark, R.G.: The impact of the mit-bih arrhythmia database. IEEE Engineering in Medicine and Biology Magazine 20(3), 45–50 (2001)
- 10. Tan, S., Caruana, R., Hooker, G., Lou, Y.: Detecting bias in black-box models using transparent model distillation. arXiv preprint arXiv:1710.06169 (2017)
- Wang, F., Rudin, C.: Falling rule lists. In: Artificial Intelligence and Statistics. pp. 1013–1022 (2015)
- Zaunseder, S., Huhle, R., Malberg, H.: Cinc challenge assessing the usability of ecg by ensemble decision trees. In: 2011 Computing in Cardiology. pp. 277–280. IEEE (2011)