# An Explainable Model For Diabetes Risk Prediction

Alessandro Cabroni[1], Francesca Fallucchi[1]

[1]Guglielmo Marconi University, Via Plinio 44, 00193 Rome, Italy

**Abstract**

In Artificial Intelligence, one of the most important issue concerns the necessity to understand why a particular prediction is chosen by a model from the considered input data. In this work, we propose a model, named Global Prediction Architecture, based on three layers (MultiLayerPerceptron, Closest Classes and Elements, and a third layer to combine them), where first layer produces both a partial prediction and features extraction useful for the second layer. We are interested in analyzing the behavior of the model both for accuracy and for explainability in terms of input data. We apply our study in the healthcare context of diabetes. Diabetes (diabetes mellitus) is a disease present when a person has a high blood sugar level for a long period. One import issue is related to the possibility to do prevention of the disease. We analyze the possibility to determine the diabetes risk in respect to daily lifestyle and health parameters, such as Body Mass Index, age, waist circumference, use of blood pressure medication, history of high blood glucose, physical activity, consumption of vegetables/fruits/berries, and family history of diabetes. We produce datasets randomly generated according to the rule named Finnish Diabetes Risk Score. This work aim to produce random and anonymized diabetes risk datasets, to test a model in terms of improving accuracy for the prediction of diabetes risk, and, most of all, to propose and test a method for explainability in the context of diabetes prediction, using an approach initially derived from Layer-Wise Relevance Propagation and Deep Taylor Decomposition.

**Keywords**

Diabetes risk prediction, FINnish Diabetes RIsk SCore, Multilayer Perceptron, Explainability, Layer-Wise Relevance Propagation, Deep Taylor Decomposition

## 1. Introduction

In healthcare, one of the topics of interest is the diseases prevention. In this work, we consider the problem of identifying risks for type 2 diabetes for a person. We are interested in three principal issues: production of testing datasets, definition of a model to improve prediction accuracy, definition of an explainability method adequate to the prediction model. About first issue, we use dataset randomly generated according to the rule FINnish Diabetes RIsk Score (FINDRISC) [1]. Using random datasets, we have the possibility to establish controlled data useful to compare different models, and without any privacy problem. About second issue, we consider a new model based on three layers, first of all Multilayer Perceptron (MLP). This layer produces both prediction and features extraction. Extracted features are used by a second layer based on comparing one unlabeled node (testing node) with all labelled nodes (training nodes) in terms of similarities, considering class (diabetes risk level) similarity too. Third layer put together the predictions of first two layers in a weighted manner. We name the overall model Global Prediction Architecture (GPA). We obtain an accuracy better than using some algorithms of Waikato Environment for Knowledge Analysis (WEKA) tool [2], and an accuracy slightly better than using only MLP component. We establish the diabetes risk according to daily lifestyle and health parameters, such as Body Mass Index (BMI), age, waist circumference, use of blood pressure medication, history of high blood glucose, physical activity, consumption of vegetables/fruits/berries, and family history of diabetes. There are other works about the issue for diabetes (e.g. [3]). About third issue, we propose an explainability solution based on reasoning about relevance of input data in respect to the prediction. In particular, we combine a new solution conceptually derived from Layer-Wise Relevance Propagation (LRP) and Deep Taylor Decomposition (DTD) (e.g., [5], [7]), with the distribution of extracted features testing data, to capture the relevance of the features in the second layer. Hence, from the explainability point of view, we have a theoretical model considering first, and implicitly third, layers, and a model based on data distribution (we consider the standard deviation of the single feature in respect to training data) for second, and implicitly third, layers. We could add our solution to other studies in a similar context (e.g., [8]). We could also explore the use of the solution in an Internet of Things (IoT) context, considering the possibilities of 5G network too (about this last subject, e.g. see [9], [11]), and for other domain different architectures (e.g. see [12], [13, 14]). The following sections organize as follows. Related work section reports some works about prediction on diabetes and a summary of the major concepts behind LRP and DTD. Methodol-

*SYSTEM 2021 @ Scholar's Yearly Symposium of Technology, Engineering and Mathematics. July 27–29, 2021, Catania, IT*

a.cabroni@studenti.unimarconi.it (A. Cabroni);
f.fallucchi@unimarconi.it (F. Fallucchi)
http://www.gei.de/en/mitarbeiter/francesca-fallucchi-phd.html (F. Fallucchi)
0000-0002-0428-3234 (A. Cabroni); 0000-0002-3288-044X (F. Fallucchi)

ogy section reports the step followed in the research for this work. Tools and environments section reports the principal tools and environments used to implement and test our solution. Dataset definition section outlines the rule used to implement random generated datasets, with a visual distribution in terms of mean and standard deviation over all input attributes. Dataset analysis section reports the result of an analysis conducted on the greatest training dataset. It presents the results of a prediction test using some algorithms available on WEKA tool. Prediction model section describes the model defined and tested in this work in the context of diabetes risk prediction. In this section, we present also the used accuracies definition, the values of hyper-parameters which instantiate the model, and the prediction results. Explainability model section presents our solution to explain the behavior of the prediction model in terms of input data. This section reports also the hyper-parameters used in the test finalized to explainability and the results in terms of input data relevance for the prediction. In the conclusion section, we briefly summarize the obtained results about dataset creation, accuracy prediction, and explainability.

## 2. Related work

In this section, we first briefly cite three chosen works about prediction in the context of diabetes. Then, we present some concepts about explainability, in particular for LRP and DTD.

In [15], they use Pima Indian Diabetes (PID) dataset and they test seven Machine Learning (ML) algorithms for predictions related to diabetes, using WEKA tool too. They obtain the best results by using Logistic Regression (LR) and Support Vector Machine (SVM) for diabetes prediction. They also implemented a Neural Network (NN) with two hidden layers for the accuracy. In [16], they evaluate the risk of diabetes based on lifestyles and family background. They consider 952 instances produced by questionnaire related to health, lifestyle and family background. They applied different ML algorithms both to this dataset and to PID dataset. Most accurate performance is for Random Forest (RF) Classifier. Also in [17], they trained the ML models using PID dataset. They propose a framework based on pre-processing, K-fold Cross-validation (KCV), Grid search for hyper-parameters, to select the best model among different algorithms. In future work they are interested in applying their results in other medical context to verify the general usefulness. In the general context of explainability, as basis for our studying we are interested on LRP and DTD. In this section, we review some of the concepts described in [5], [7], [18], [19], and [20]. In LRP, prediction back propagate in the NN. Each propagation redistributes in the lower level of the NN by a conservation rule:

$$R_j = \sum_k \frac{z_{ik}}{\sum_j z_{jk}} R_k \tag{1}$$

$z_{ik}$ corresponds to how much the neuron j contributes to be relevant for neuron k. The recursive propagation finishes at the input data. One single step can be defined as a Taylor decomposition. In our context, we consider a MLP as an acyclic graph based on Rectified Linear Unit (ReLU) activation function at each layer with input data not less than zero. Supposing to have a neuron N receiving the scalar $x_{input} = (x_1, \ldots, x_n)$ and producing the scalar $y_{output}$, we have:

$$y_{output} = \max(0, \sum_{k=1}^{n} x_i \, w_{ij} + b_j) \tag{2}$$

with $b_j <= 0$. Considering DTD, we have that LRP corresponds to a succession of Taylor expansions local for each neuron. We now consider that the output can be described as a first-order Taylor expansion. Defining $[y_{output}]_i$ as the redistribution of $y_{output}$ on the neuron i of the lower layer, we have the rule of redistribution ($z^+$-rule) when the lower level of N is a ReLU layer:

$$[y_{output}]_i = \frac{x_i \, w^+_{i,j}}{\sum_{k=1}^{n} x_k \, w^+_{kj}} \, y_{output} \tag{3}$$

n equals to the number of neurons in the lower level of N; $v^+ = |v|$. Defining $x_f$ as the final output of the NN for a particular input data, we have that $[[x_f]_j]_i$ corresponds to the quantity of $x_f$ distributed from one node j to one node i, where i is an input node for node j. $[x_f]_i$ corresponds to the quantity of $x_f$ distributed on node i:

$$[x_f]_i = \sum_{j=1}^{n^1} [[x_f]_j]_i \tag{4}$$

$n^1$ equals to the number of nodes in the higher level for node i. Considering $[x_f]_j = x_j \, c_j$ (neuron activation and constant value), we have:

$$[x_f]_i = \sum_{j=1}^{n^1} [[x_f]_j]_i = \sum_{j=1}^{n^1} [x_j \, c_j]_i = \sum_{j=1}^{n^1} [x_j]_i \, c_j = x_i \, c_i \tag{5}$$

$n^1$ equals to the number of nodes in the higher level of node i. Moreover,

$$c_i = \sum_{j=1}^{n^1} \frac{w^+_{ij} \, [x_f]_j}{\sum_{i^1=1}^{n} x_{i^1} \, w^+_{i^1 j}} \tag{6}$$

$n^1$ equals to the number of nodes in the higher level of node i, and n is the number of neurons in the lower level of j (the same level of i). At the beginning, we have $[x_f]_f = x_f \, c_f$ and $c_f = 1$. By induction, there is a product structure with a backward propagation rule and the conservation of the output (redistribution on input nodes).

**Table 1**
Used environments and tools.

| Set | Tool and environment |
|---|---|
| Datasets definition<br>Prediction<br>Explainability | Colaboratory (Colab) - Backend Google Compute Engine Python™ 3<br>RAM: 0.75GB out of 12.69GB; available disk space: 38.47GB<br>out of 107.72GB Tensor Processing Unit (TPU) |
| Dataset analysis | Weka 3.8.5 - Windows 8.1 (64 bit) – Intel® Celeron® CPU 1007U<br>1.50Ghz RAM 4G (3.88G usable) |

## 3. Methodology

In this research, we firstly analyzed some papers about diabetes risk predictions and explainability for LRP and DTD. We selected a rule (FINDRISC) to produce random datasets. We defined our prediction model, named GPA, with three layers: MLP for partial prediction and features extraction, Closest Classes and Elements (CCE) for partial prediction, Weighted Sum (WS) for final prediction. CCE evaluates similarity between one unlabeled node and all labelled nodes, using extracted features. WS sums the two partial predictions, adequately weighted, to define the final prediction by argmax. We analyzed the best hyper-parameters, using GridSearchCV of scikit-learn tool too. We analyzed the test predictions comparing GPA accuracies against MLP accuracies and some WEKA algorithms accuracies. We defined explainability solution based on a forward DTD derived component for first and implicitly third layer, and on weight (standard deviation) of extracted features (related to the training data) for second and implicitly third layer. We tested the explainability using a simplified MLP.

## 4. Tools and environments

Table 1 reports the used environments and tools distinguishing between first set (datasets definition, prediction, explainability) and second set (only dataset analysis).

## 5. Dataset definition

We generate eight datasets according to FINDRISC [1]. Four datasets are for prediction experiments (2500 elements for testing; 1000, 1500, and 2000 for training), and other four are for explainability experiments (1750 elements for testing; 1000, 1250, and 1500 for training). The rule identifies risk individuals, without laboratory tests. It considers five risk levels in respect to score: very low (0-3), low (4-8), moderate (9-12), high (13-20) and very high (21-26). All datasets are equally balanced in respect to the possible scores. These are the attributes to be considered: BMI (weight (kg) / height squared (m2)), age (years), waist circumference (differentiating for gender), use of blood pressure medication, history of high blood glucose, physical activity expressed in hours/week, daily consumption of vegetables, fruits or berries, family history of diabetes. The score is calculated according to the rule. The random input data are normalized to [0, 1]. These are the input data of our prediction model, while the risk score is the right prediction. In Figure 1, we can see the distribution (mean and standard deviation) of the generated datasets.

## 6. Dataset analysis

For a preliminary analysis of the produced datasets, we considered the dataset with 2000 elements and we analyzed in details both their data distribution and accuracy results, by using WEKA tool [2]. In Table 2 we can see accuracy results for the considered algorithms: J48, KStar, MLP, Naïve Bayes (NB), RandomTree (RT). We used 10-fold cross-validation for the analysis.

## 7. Prediction model

Our prediction model have three layers: first layer is MLP, second layer is CCE (it uses features extracted from MLP), and third layer combines predictions of both MLP and CCE. MLP have the following elements: dense, batch normalization, activation–ReLU, dropout, dense, batch normalization, activation–ReLU, dropout, dense, batch normalization, activation–ReLU, dropout, dense, activation–Softmax. CCE uses the features extracted from third dense layer; for each testing node we implement this algorithm:

- Calculate Euclidean distance between the considered testing node and all training nodes
- Normalize these distances to [0,1]
- Using normalized distances, calculate Gaussian kernel similarity between the considered testing node and all training nodes:

$$sim_{i,j} = e^{-\frac{d(i,j)^2}{2\,\sigma^2}} \tag{9}$$

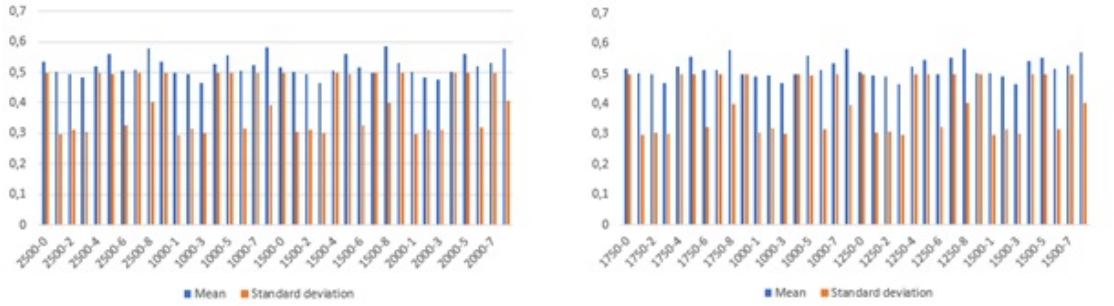


**Figure 1:** Dataset distribution for input data (from left to right: prediction and explainability experiments).

**Table 2**
WEKA prediction tests using dataset with 2000 nodes (10-fold cross-validation).

| Model | Accuracy |
|---|---|
| J48 | 0.826 |
| KStar | 0.71 |
| MLP | 0.7965 |
| NB | 0.713 |
| RT | 0.7595 |

- Normalize these similarities to [0, 1]
- For each possible label (risk class), calculate the sum of similarities
- Normalize all sums to the overall sum
- Recalculate sum distribution, considering the similarity between labels too, according to the following algorithm (m is the number of labels/risk classes):

*STemp=NP.copy(S)*
*S[0]=STemp[0]+STemp[1]*(m-1)/m*
*for h in range(1,m-1): S[h]=STemp[h]+(STemp[h-1]+STemp[h+1])/2*(m-1)/m*
*S[m-1]=STemp[m-1]+STemp[m-2]*(m-1)/m*
*S=S/NP.sum(S)*

The third layer of the prediction model, considering the single testing node, for each class produces the weighted sum of probabilities obtained by both MLP and CCE for that class. Prediction is obtained by argmax function.

## 7.1. Accuracy definition

We consider two accuracy definitions (eq. 7 and eq. 8). The second definition uses the similarities between labels, because risk levels are orderable. In particular: np is the number of unlabeled nodes, $RL_i$ is the right label for i node, $PL_i(j)$ is the value of probability distribution for unlabeled node i considering label j, and m is the number of possible labels (classes, risk levels).

## 7.2. Hyper-parameters

Hyper-parameters have been chosen with some preliminary tests. Initially, for MLP component we considered GridSearchCV for a first analysis. We produced a random dataset of 1000 nodes using mlpClassifier with 200 max iterations and the following parameter space: hiddenLayerSizes with (128,256,32), (256,512,32), (512,1024,32); learningRateInit with 0.01, 0.1; validationFraction with 0.1, 0.2; batchSize with 50, 100. For this analysis, we obtained this results as best: batchSize=100, hiddenLay-

$$accuracy = \frac{\sum_{i=1}^{np} if(argmax_{j\in\{1,\dots,m\}} PL_i(j) = RL_i, 1, 0)}{np} \tag{7}$$

$$accuracy1 = \frac{\sum_{i=1}^{np} (1 - \frac{|argmax_{j\in\{1,\dots,m\}} PL_i(j) - RL_i|}{m-1})}{np} \tag{8}$$

**Table 3**
Hyper-parameters for prediction tests.

| Component | Parameter | Value |
|---|---|---|
| input | numberOfAttributes | 9 |
| MLP | batchSizeMLP | 100 |
| MLP | decayMLP | 1e-6 |
| MLP | dropoutParameterMLP | 0.25 |
| MLP | epochsMLP | 1000 |
| MLP | learningRateMLP | 0.01 |
| MLP | unitsFirstDenseMLP | 512 |
| MLP | unitsSecondDenseMLP | 1024 |
| MLP | unitsThirdDenseMLP | 16, 32 (extracted features) |
| MLP | unitsFourthDenseMLP | 5 (prediction classes) |
| MLP | validationSplitMLP | 0.2 |
| CCE | gaussianKernelWidth | 0.5 |
| WS | modelWeightMLP | 0.05 |
| WS | modelWeightCCE | 0.95 |

**Table 4**
Accuracies results for MLP and GPA

| Model | Number of labelled nodes | Number of extracted features | Accuracy | Accuracy1 |
|---|---|---|---|---|
| MLP | 1000 | 16 | 0.8252 | 0.9559 |
| MLP | 1000 | 32 | 0.8188 | 0.9543 |
| MLP | 1500 | 16 | 0.8356 | 0.9587 |
| MLP | 1500 | 32 | 0.8392 | 0.9597 |
| MLP | 2000 | 16 | 0.8552 | 0.9638 |
| MLP | 2000 | 32 | 0.8564 | 0.9641 |
| GPA | 1000 | 16 | 0.82673008 | 0.9562 |
| GPA | 1000 | 32 | 0.82512752 | 0.9558 |
| GPA | 1500 | 16 | 0.8399342399999999 | 0.9597 |
| GPA | 1500 | 32 | 0.8463356799999999 | 0.9614 |
| GPA | 2000 | 16 | 0.8599420799999999 | 0.9649 |
| GPA | 2000 | 32 | 0.8583425600000001 | 0.9645 |

erSizes=(128, 256, 32), learningRateInit=0.01, validationFraction=0.1. After other empirical tests, we chose the hyper-parameters described in Table 3.

### 7.3. Prediction results

In Table 4, we present accuracy results for both only MLP component and all model GPA. In Figure 2, we outline the differences between accuracy of GPA and MLP. As we can see, we have a slightly better performance with GPA. Moreover, the results are better than the results obtained with the algorithms tested with WEKA tool. Of course, we must remember that we are reasoning with restricted random datasets and so our conclusions are useful only from a testing point of view and not for healthcare formal deductions. In Table 5, we present execution times for prediction tests.

## 8. Explainability model

Considering DTD theory, we define a simplified rule to calculate the relevance of the single input parameter against the single feature extracted from MLP first layer. We consider the weights of the edges for the trained MLP. We do not consider biases. Moreover, we manage the possible weight of the features for the Gaussian kernel distances in the CCE layer. The potential weight is calculated according to labelled nodes, corresponding to training data. We weight the standard deviation of a feature. In fact, features with a high variation give a high contribute to the substantial distances and so they have a significant contribute for prediction classification (we could also analyze better the possibility to normalize training features values). We first obtain a formula for explainability which does not depend from the particular input data and prediction. Then, we apply the formula to a single input data by multiplication, so to calculate the percentage of relevance for that parameter in the

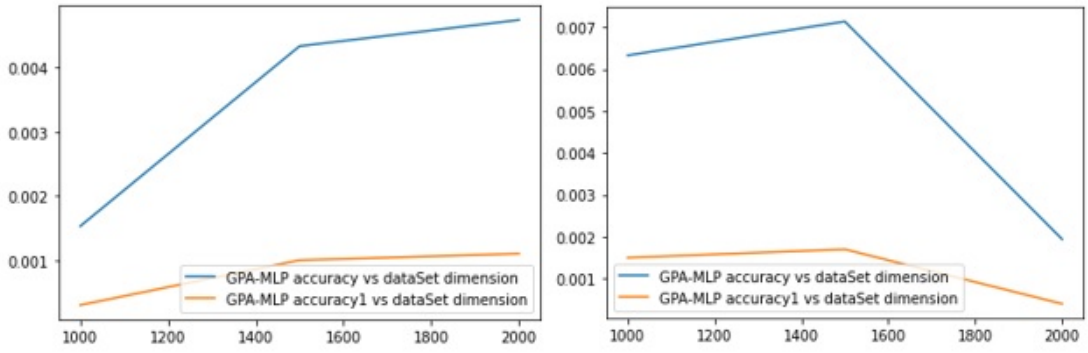


**Figure 2:** GPA-MLP accuracy/accuracy1 vs number of extracted feature (left: 16; right: 32).

**Table 5**
Execution times of prediction model.

| Number of labelled nodes | Number of extracted features | Execution time |
|---|---|---|
| 1000 | 16 | 00:03:28 |
| 1000 | 32 | 00:03:31 |
| 1500 | 16 | 00:04:50 |
| 1500 | 32 | 00:04:53 |
| 2000 | 16 | 00:06:14 |
| 2000 | 32 | 00:06:21 |

considered prediction. Here, we use a forward definition for explainability, maintaining the conservation of total relevance. Formally, if we have a normalized unlabeled node represented by the input data $(v_1, \ldots, v_n)$, with $\forall i \in \{1, \ldots, n\}\ v_i \in [0, 1]$, we can establish the relevance of input i in the prediction as $R_{i,v}$ ddefined in the equations (10), (11), (12), (13). Where:

- F is the number of extracted features
- $x_f^i$ is the value of feature f for labelled node i
- N is the number of labelled nodes (training dataset)
- $C_{l+1}$ is the number of neurons for layer l+1 of MLP (layer 0 is the input data)
- LF is the layer of MLP for features extraction; for a particular f, this layer has only one neuron

$$R_{i,v} = \frac{v_i \sum_{f=1}^{F} [Rnorm_{i,0}^{f}\, \sigma_f]}{\sum_{j=1}^{n} v_j \sum_{f=1}^{F} [Rnorm_{j,0}^{f}\, \sigma_f]} \tag{10}$$

$$\sigma_f = \frac{\sqrt{\sum_{i=1}^{N} (x_f^i - \frac{\sum_{i=1}^{N} x_f^i}{N})^2} - min_{f\in\{1,\ldots f\}} \sqrt{\sum_{i=1}^{N} (x_f^i - \frac{\sum_{i=1}^{N} x_f^i}{N})^2}}{max_{f\in\{1,\ldots f\}} \sqrt{\sum_{i=1}^{N} (x_f^i - \frac{\sum_{i=1}^{N} x_f^i}{N})^2} - min_{f\in\{1,\ldots f\}} \sqrt{\sum_{i=1}^{N} (x_f^i - \frac{\sum_{i=1}^{N} x_f^i}{N})^2}} \tag{11}$$

$$Rnorm_{i,l}^{f} = \sum_{j=1}^{C_{l+1}} \frac{Rnorm_{j,l+1}^{f}\, w_{i^l j^{l+1}}^{+}}{\sum_{k=1}^{C_l} w_{k^l j^{l+1}}^{+}} \tag{12}$$

$$Rnorm_{1,LF}^{f} = 1 \tag{13}$$

**Table 6**
Hyper-parameters for explainability tests.

| Component | Parameter | Value |
|---|---|---|
| input | numberOfAttributes | 9 |
| MLP | batchSizeMLP | 100 |
| MLP | decayMLP | 1e-6 |
| MLP | dropoutParameterMLP | 0.25 |
| MLP | epochsMLP | 1000 |
| MLP | learningRateMLP | 0.01 |
| MLP | unitsFirstDenseMLP | 64 |
| MLP | unitsSecondDenseMLP | 128 |
| MLP | unitsThirdDenseMLP | 8 (extracted features) |
| MLP | unitsFourthDenseMLP | 5 (prediction classes) |
| MLP | validationSplitMLP | 0.2 |
| CCE | gaussianKernelWidth | 0.5 |
| WS | modelWeightMLP | 0.05 |
| WS | modelWeightCCE | 0.95 |
| explainability | C | [9,64,128,1] |
| explainability | LF | 3 |
| explainability | MLPLevelsForExplainability | [0,4,8] |

corresponding to the particular feature extracted f

- $w^{+}_{i^l j^{l+1}}$ is the absolute value of the weight of MLP for the edge which connect neuron i of layer l with neuron j of layer l+1

As we can see, $v_i$ is the only element related to the particular input data. All the other elements expressed in the formulas depends only on fixed hyper-parameters and on original training dataset. Moreover, in our tests, we calculate the constant components of $R_{i,v}$ only once, so to optimize the tests computation.

## 8.1. Hyper-parameters

Starting from hyper-parameters used for prediction tests, we simplified MLP component establishing new hyper-parameters for explainability tests, where we repeated training and prediction process too. We report the chosen configuration values in Table 6.

## 8.2. Explainability results

In Figure 3, we present the results of explainability tests. In particular, we can see the average relevancies for all input data in respect to all testing predictions. E.g., we can see particular relevancies for parameter 1 (age), 2 (BMI), 3 (waist circumference), 8 (family history) and less relevancies for parameter 2 (gender). Of course, we must remember that we are reasoning with restricted random datasets and so our conclusions are useful only from a testing point of view and not for healthcare formal deductions. In Table 7, we present execution times for explainability tests.

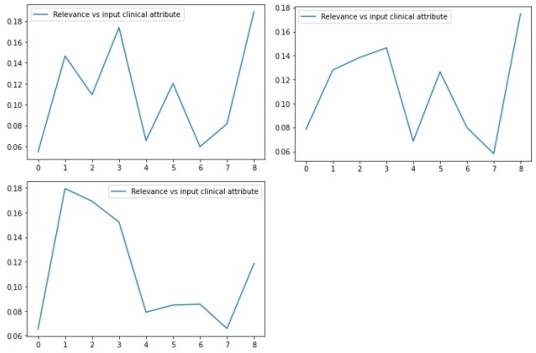


**Figure 3:** Average relevancies vs number of training nodes (left/top: 1000; right/top: 1250; left/bottom: 1500).

# 9. Conclusion

In this work, we have proposed an explainable model to predict diabetes risk. We have tested our model using random defined datasets produced according to a healthcare rule named FINDRISC. We chose to define random data to have the possibility to evaluate our model in a controlled manner (input data are sufficiently distributed and risk predictions are equally distributed) and to overcome any privacy problem. We defined our model, named GPA, using three layers. First layer considers a MLP module and it is used both to produce a first partial mixed prediction and to extract features for the second layer. This layer, named CCE, produces a partial mixed prediction considering the similarity between a single unlabeled node in

**Table 7**
Execution times of explainability model.

| Number of labelled nodes | Number of extracted features | Execution time |
|---|---|---|
| 1000 | 8 | 00:02:44 |
| 1250 | 8 | 00:02:32 |
| 1500 | 8 | 00:02:39 |

respect to all labelled nodes, managing class distances too. In this layer, a node is represented by the features extracted from first layer. Third layer, named WS, considers the sum of the partial mixed prediction of the first and second layer (in a weighted manner) to obtain the final prediction by argmax. Experimentally, we noticed that accuracy improves using the all GPA model in respect to using only MLP layer. Moreover, we noticed that accuracy results are better than considering accuracy results produced using some algorithms of WEKA tool. The most contribute of our research is the explainability of our model in terms of input parameters, useful for a MD (medical doctor) understanding, also considering more predictions together. Generally, we must remember that now our conclusions are useful only from a testing point of view and not for real deductions.

## References


[1] https://www.mdcalc.com/findrisc-finnish-diabetes-risk-score, last accessed 2021/16/07

[2] Weka 3: Machine Learning Software in Java, https://www.cs.waikato.ac.nz/ml/weka/index.html, last accessed 2021/16/07

[3] Xiong, Xl., Zhang, Rx., Bi, Y. et al. Machine Learning Models in Type 2 Diabetes Risk Prediction: Results from a Cross-sectional Retrospective Study in Chinese Adults. CURR MED SCI 39, 582–588 (2019). https://doi.org/10.1007/s11596-019-2077-4

[4] Illari, S.I., Russo, S., Avanzato, R., Napoli, C. A cloud-oriented architecture for the remote assessment and follow-up of hospitalized patients. (2020) CEUR Workshop Proceedings, Vol. 2694, pp. 29-35.

[5] Bach S, Binder A, Montavon G, Klauschen F, Müller K-R, Samek W, On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation, PLOS ONE, 2015, https://doi.org/10.1371/journal.pone.0130140

[6] Napoli C, Pappalardo G, Tramontana E, A hybrid neuro–wavelet predictor for qos control and stability. (2013) Lecture Notes in Computer Science, Vol. 8249 LNAI, pp. 527–538.

[7] Montavon G, Lapuschkin S, Binder A, Samek W, Müller K-R, Explaining nonlinear classification decisions with deep Taylor decomposition, Pattern Recognition, 2017

[8] L. Kopitar, et al. (2019) Local vs. Global Interpretability of Machine Learning Models in Type 2 Diabetes Mellitus Screening. https://doi.org/10.1007/978-3-030-37446-4_9

[9] F Mazzenga, R Giuliano, F Vatalaro, "FttC-based fronthaul for 5G dense/ultra-dense access network: Performance and costs in realistic scenarios", Future Internet 9 (4), 71, 2017, https://doi.org/10.3390/fi9040071

[10] Russo, S., Illari, S.I., Avanzato, R., Napoli, C. Reducing the psychological burden of isolated oncological patients by means of decision trees (2020) CEUR Workshop Proceedings, 2768, pp. 46-53.

[11] Giuliano, R., Mazzenga, F., Vizzarri, A., "Satellite-Based Capillary 5G-mMTC Networks for Environmental Applications", IEEE Aerospace and Electronic Systems Magazine, 2019, 34(10), pp. 40–48, https://doi.org/10.1109/MAES.2019.2923295

[12] Capizzi, G., Lo Sciuto, G., Napoli, C., Tramontana, E. A multithread nested neural network architecture to model surface plasmon polaritons propagation (2016) Micromachines, 7 (7), art. no. 110 (7)

[13] Capizzi, G., Lo Sciuto, G., Napoli, C., Tramontana, E., Woźniak, M. A novel neural networks-based texture image processing algorithm for orange defects classification (2016) International Journal of Computer Science and Applications, 13 (2), pp. 45-60. (7)

[14] Napoli, C., Bonanno, F., Capizzi, G. Exploiting solar wind time series correlation with magnetospheric response by using an hybrid neuro-wavelet approach. (2010) Proceedings of the International Astronomical Union, 6 (S274), pp. 156-158. DOI: 10.1017/S1743921311006806

[15] Jobeda Jamal Khanam, Simon Y. Foo, A comparison of machine learning algorithms for diabetes prediction, ICT Express, 2021, ISSN 2405-9595

[16] Neha Prerna Tigga, Shruti Garg, Prediction of Type 2 Diabetes using Machine Learning Classification Methods, Procedia Computer Science, Volume 167, 2020, Pages 706-716, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.03.336

[17] M. K. Hasan, et al., "Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers",

2020, doi: 10.1109/ACCESS.2020.2989857

[18] Montavon G, et al., Layer-Wise Relevance Propagation: An Overview, Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, https://doi.org/10.1007/978-3-030-28954-6_10

[19] Montavon G, Bach S, Binder A, Samek W, Müller K-R, Deep Taylor Decomposition of Neural Networks, Proceedings of the ICML'16 Workshop on Visualization for Deep Learning, 2016

[20] Kauffmann J, et al.,Towards explaining anomalies: A deep Taylor decomposition of one-class models,Pattern Recognition,https://doi.org/10.1016/j.patcog.2020.107198