International Journal of Data Science and Analytics
https://doi.org/10.1007/s41060-024-00526-9

REGULAR PAPER

Pseudo datasets explain artificial neural networks

Yi-Chi Chu[1] · **Yi-Hau Chen**[2] · **Chao-Yu Guo**[1]

Received: 6 January 2024 / Accepted: 19 February 2024

Abstract

Machine learning enhances predictive ability in various research compared to conventional statistical approaches. However, the advantage of the regression model is that it can effortlessly interpret the effect of each predictor. Therefore, interpretable machine-learning models are desirable as the deep-learning technique advances. Although many studies have proposed ways to explain neural networks, this research suggests an intuitive and feasible algorithm to interpret any set of input features of artificial neural networks at the population-mean level changes. The new algorithm provides a novel concept of generating pseudo datasets and evaluating the impact due to changes in the input features. Our approach can accurately obtain the effect estimate from single to multiple input neurons and depict the association between the predictive and outcome variables. According to computer simulation studies, the explanatory effects of the predictors derived by the neural network as a particular case could approximate the general linear model estimates. Besides, we applied the new method to three real-life analyzes. The results demonstrated that the new algorithm could obtain similar effect estimates from the neural networks and regression models. Besides, it yields better predictive errors than the conventional regression models. Again, it is worth noting that the new pipeline is much less computationally intensive than the SHapley Additive exPlanations (SHAP), which could not simultaneously measure the impact due to two or more inputs while adjusting for other features.

Keywords Interpretation · Neural networks · Machine learning · Predictive model · SHapley

1 Introduction

Theories and applications of Machine learning are expanding fast, and many industries use it to make better predictions, including the emergency department triage prediction and COVID-19 outcome risk assessment [1–3]. Machine learning models are better at predicting data in structural diversity than conventional statistical modeling [4]. Our previous work [5–8] revealed superior performance using novel machine learning models compared to the statistical approaches in numerous settings. However, researchers are highly concerned about machine learning models' interpretation mechanisms, such as support vector machines [9], random forests [10], K-neighborhood algorithms [11], and artificial neural networks (ANN) [12–14]. The landmark articles of White [13, 14] provide in-depth statistical perspectives and interpretability of ANN. Another goal is to identify the problematic points to solve the effectiveness of the models when the results diverge from the actual values [15].

Machine learning models are also essential in health care [16] or medical research such as acute pancreatitis in-hospital mortality [17], oral cancer risk [18], acute liver function mortality [19], diabetes screening [20], glaucoma progression in sleep apnea patients [21], and COVID19 diagnosis [22]. Further understanding how machine learning describes the relationship between variables will increase confidence in the model and allow people to use it for subsequent predictions. Even though these models can accurately predict disease progression or diagnosis, they are ineffective at understanding the impact of predicted input variables on the target.

Yi-Chi Chu and Yi-Hau Chen have authors contributed equally to this work.

✉ Chao-Yu Guo
cyguo@nycu.edu.tw

[1] Division of Biostatistics and Data Science, Institute of Public Health, College of Medicine, National Yang Ming Chiao Tung University, Taipei, Taiwan

[2] Institute of Statistical Science, Academia Sinica, Taipei, Taiwan

Published online: 10 April 2024

It is natural to construct models into an opaque box to attain the most accurate predicted performance with growing complexity in their setup due to the development of deep learning [23]. Therefore, many novel techniques aim to simplify model interpretation, including the Local Interpretable Model-Agnostic Explanation technique (LIME) [24]. The LIME approximates a machine learning model with a local and interpretable model. The LIME explains each individual's prediction. This model chooses only the most essential variables for interpretation, and how they affect the result is unclear. One of the most widely used models in machine learning is neural networks, which have gained popularity recently, particularly with the advent of deep neural networks [25].

In addition to the LIME, the SHapley Additive exPlanations (SHAP) [26] comprehend the significance of the variables in the data and show the direction of the impact of features on predictions. SHAP is applicable for both local and global explanations. The absolute Shapley values of all instances in the data are averaged for global explanations. Another method implemented the statistical techniques with the multinomial regression models to generate neural network topologies [27]. The issue is the instability of the data structure owing to Taylor's spreading effect affects the analysis of the actual data.

Recently, ridge regression was used to propose a neural network with feature sparsity [28], developed the generalized additive model into a Neural Additive Network [29], and derived a neural network architecture using polynomial regression [30]. Using statistical theory to calculate weights and specified activation functions efficiently explains the association between independent variables and the outcome feature in neural networks. Another strategy used statistical techniques with the multinomial regression models to generate neural network topologies [27]. Besides applying statistical theory to construct models with calculated weights and specified activation functions, these are efficient ways to comprehend the fundamental link between independent variables and contingent factors in neural networks.

A systematic review discussed the evaluation of Explainable Artificial Intelligence (XAI) results because it is essential for maximizing the value of Artificial Intelligence (AI)-based clinical decision support systems [31]. The previous techniques address the issue of neural networks' inability to explain how well a prediction worked and compute the weights or coefficients of various variables to comprehend how the model affects the predicted variables. In addition, a study compared explainable ensemble learning and logistic regression for predicting in-hospital mortality in the emergency department [32]. Yet another study improved patient mortality predictions in emergency departments with deep learning data-synthesis and ensemble models [33].

In this research, we aim to propose a novel and feasible approach named the interpretable neural network algorithm (INNA) for measuring the effect from one or multiple input features and estimating the population-level impact. The new method generates pseudo datasets according to perturbations in the input features and then evaluates the effects on the predicted values.

2 Methods

Medical science researchers commonly measure the impact of a specific predictive variable by one standard deviation (SD) change in the predictor and see how much difference is expected in the outcome variable [34–36]. Thus, we adopted the popular measure of one SD change for the interpretation algorithm.

When the input neuron is continuous, the INNA computes the expected change in the output neuron according to one standard deviation (SD) increase/decrease in the input neuron. Regarding the dichotomous input neuron, we obtain the expected difference of the output neuron based on the contrast (1 vs. 0) in the input neuron. Since the generalized linear model (GLM) [37] is a well-known interpretable statistical model, we aim to fine-tune the INNA to match the results of the GLM as a particular case for illustration purposes. We will conduct computer simulations and three real-life data analyzes to compare the results between the INNA and GLM. In the following, we carry out details of the INNA for continuous outcomes.

Consider an ANN with J input neurons and one output neuron. We normalize every continuous feature in the input layer. For categorical features, we generate dummy variables such that all input neurons become dichotomous with values of 1 or 0. Assuming the sample size is N. Let $X = (x_1, x_2, \ldots, x_J)$ be the collection of J input neurons, and Y be the output neuron, then $X_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$ denotes the observed feature/predictors of the ith subject, where $i =$ 1,2,…, N. Finally, Y_i is the actual outcome value of the ith subject.

Since the outcome is continuous, the loss function is the mean square error (MSE) defined as $\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \widehat{Y}_i\right)^2$, where $\widehat{Y}_i$ is the predicted value from the ANN. Since the output value affects the threshold in resilient back-propagations of the ANN, we set the cutoff as the maximum output value for resilient back-propagations, and the maximum number of iterations is 10,000 (stepmax:10,000). When training the ANN model, the initial weights of the ANN are randomly selected. Thus, we repeated the training process ten times (repetition:10) and chose the optimal one. As a result, the parameter setting is activation: Relu, hidden layer:1, nodes = 4:9, stepmax:10,000, threshold = 0.1, learningrate = 0.1, repetition:10 (algorithm: rprop+). The variation is minimal

after 10,000 iterations; ten repetitions provide consistent results. Therefore, we use these numbers as the threshold.

The INNA starts with the first input neuron. Then, the INNA iterates through the rest of the components of the input layer. For example, assuming we only have three predictors, such as $J = 3$ (x_1, x_2, x_3) and one outcome variable (y) in the training data. We begin the INNA with the first variable (x_1) and keep the other two variables (x_2 and x_3) as the original unchanged values. Next, run through the algorithm for the second variable (x_2) and keep the other two (x_1 and x_3) intact. Finally, we finish the algorithm for the last feature x_3. Figure 1 shows the detailed steps of the INNA from the original dataset to the last step that calculates the effect estimates for a continuous or dichotomous input neuron.

2.1 The reference ANN model

The first step is to obtain the optimal hyperparameters and parameters by fitting the ANN to the original dataset with the settings mentioned previously. The estimated results are INNA's reference ANN model (RANN). The predicted values $\widehat{Y}_i$, $i = 1, 2, \ldots, N$, from the RANN, are the baseline components for the effect estimate. We denote the mean of $\widehat{Y}_i$, $i = 1, 2, \ldots, N$, which is also the expected value of the output neuron, as $A0$. Therefore, $A0 = \frac{1}{N}\sum_{i=1}^{N}\widehat{Y}_i$. Mathematically, Let $\widehat{Y}_i = f(X_i; \widehat{\theta})$ where $X_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$ is the i-th subject input vector, $\widehat{\theta}$ is the trained parameters and $\widehat{Y}_i$ is the predicted value from the ANN for subject i. Then $A0 = \frac{1}{N}\sum_{i=1}^{N}\widehat{Y}_i$ serves as a "reference" estimate for the expected outcome $E^0(Y)$, where E^0 the denotes expectation from the ANN predicted outcome over the population whose input features have a joint distribution as in the observed data sample.

2.2 Effect estimate in the higher group

After obtaining the baseline $A0$, the next step is to measure how an input neuron impacts the predicted value when the input increases. Therefore, we generate a pseudo dataset assuming the input feature has a higher value, denoted as the Higher Group (HG).

For the continuous type of input feature, we add one SD of this input feature in the data sample to every individual's value for this specific feature. Note that the rest of the input neurons are unchanged, and this step maintains the correlation structure of all input neurons in the pseudo dataset.

Take the first input neuron x_1 as an example. We denote the standard deviation of x_1 as SD_{x1}. The pseudo vector of J input neurons for subject i is $X_i^+ = (x_{i1}+\mathrm{SD}_{x1}, x_{i2}, \ldots, x_{iJ})$, $i = 1,2,\ldots, N$. Note that the variance–covariance matrix of the pseudo vector is identical to that of the original $X = (x_1, x_2, \ldots, x_J)$.

When the pseudo dataset HG is ready, we use the RANN to predict the individual results $\widehat{Y}_i^+$, $i = 1,2,\ldots, N$. The mean of the predicted values $\widehat{Y}^+$ is also the output neuron's expected value, denoted as $A1$. Therefore, $A1 = \frac{1}{N}\sum_{i=1}^{N}\widehat{Y}_i^+$. Subsequently, we calculate the difference in expected values due to one SD increase in the input neuron x_1, which is $T1 = A1 - A0$. The difference in expected values due to one SD increase in input neuron x_2, …, x_J can be obtained in an entirely analogous and parallel way.

Mathematically, let $\widehat{Y}_i^+ = f(X_i^+; \widehat{\theta})$ with $X_i^+ = (x_{i1} + \mathrm{SD}_{x1}, x_{i2}, \ldots, x_{iJ})$, $\widehat{\theta}$ the trained parameters, then $A1 = \frac{1}{N}\sum_{i=1}^{N}\widehat{Y}_i^+$ is an estimate of the expected value for the outcome $\widehat{Y}^+$ when the input features have a joint distribution that has a one-SD increase in the input neuron x_1 while all the other input neurons $x_2, \ldots, x_J$ are fixed as they were in the observed data sample. Denote such an expected outcome by $E^+(Y)$, where E^+ represents the expectation of the output feature over the population whose input feature distribution is the same as that in the observed sample, except that the input x_1 has a one-SD increase. Recall that $A0$ estimates the expected outcome $E^0(Y)$ without such a shift in the input x_1, hence $T1 = A1–A0$ is an estimate of the effect $E^+(Y) - E^0(Y)$ owing to a one-SD increase in the input x_1 on the difference of the expected outcome.

2.3 Effect estimate in the lower group

Similar to the approach in the HG, we could evaluate how the predicted value changes when the input values decrease. Therefore, we subtract one SD from every individual's input neuron and generate the second pseudo dataset as the Lower Group (LG). The other input neurons remain the same. Thus, this process controls or adjusts for the other features in the model.

Consider the first input neuron x_1 (standard deviation = SD_{x1}), the pseudo vector X_i^- becomes$(x_{i1} - \mathrm{SD}_{x1}, x_{i2}, \ldots, x_{iJ})$, $i = 1, 2, \ldots, N$, and the predicted value is denoted as $\widehat{Y}_i^-$, $i = 1, 2, \ldots, N$. The expected value of LG is $A2$. Hence, $A2 = \frac{1}{N}\sum_{i=1}^{N}\widehat{Y}_i^-$.

The next step finds the difference in predicted values due to the loss of one SD change in the input neuron, which is $T2 = A0 - A2$. The difference in expected values due to one SD decrease in input neuron x_2, …, x_J can be obtained in a fully analogous and parallel way.

As we have explained above for the measures $A1$ and $T1$, the measure $A2$ is an estimate of the expected value of the outcome $\widehat{Y}^-$ when the input features have a joint distribution that has a one-SD decrease in the input neuron x_1 while all the other input neurons $x_2, \ldots, x_J$ are fixed as they were in the observed data sample. Let $E^-(Y)$ be such an expected outcome, which is taken over the population whose input feature distribution is the same as that in the observed sample

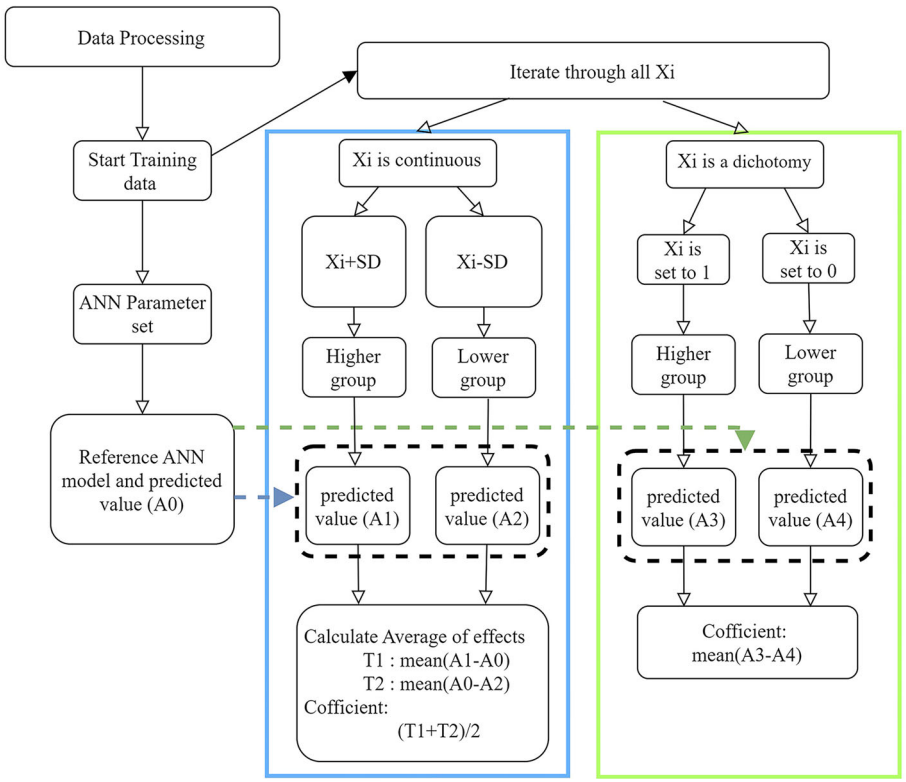


Fig. 1 Statistical Analysis Steps of the INNA

except that the input x_1 has a one-SD decrease. Since $A0$ estimates the expected outcome $E^0(Y)$ without such a change in the input neuron x_1, $T2 = A0–A2$ estimates $E^0(Y) - E^-(Y)$ which reflects the effect of the input x_1 owing to such a one-SD change.

2.4 Overall effect estimate

Since both $T1$ (for HG) and $T2$ (for LG) provided above measure the effect of the input neuron x_1 when it has a one-SD change, we consider the effect estimate of this continuous input neuron x_1 given by the mean of $T1$ and $T2$, which is $\frac{1}{2}(T1 + T2) = \frac{1}{2}(A1 - A2)$. Obtaining the average of $T1$ and $T2$ provides a more consistent estimate of the model effect.

2.5 Pseudo datasets for dichotomous neurons

Regarding the effect of a dichotomous input neuron, we change the neuron value to 1 for the HG and evaluate the difference in prediction due to such a change. This way, we could evaluate the effect of a dichotomous input neuron due to the change from 0 to 1. This formula is similar to the continuous input neurons. Take the first input neuron x_1 as an example, the pseudo vector X_i^+becomes$(1, x_{i2}, \ldots, x_{iJ})$, $i = 1, 2, \ldots, N$, and the predicted value is denoted as $\widehat{Y}_i^+$, $i = 1, 2, \ldots, N$, and $A3 = \frac{1}{N}\sum_{i=1}^{N}\widehat{Y}_i^+$. The $A3$ is then compared to the RANN $A0$ to obtain the effect estimate $A3–A0$ in the HG. Similarly to the continuous feature case, $A3$ estimates the expected outcome $\widehat{Y}^+$ when the input features have a joint distribution where the dichotomous neuron x_1 is set to 1 while all the other neurons $x_2, \ldots, x_J$ are fixed as they were in the observed data sample. We still use the notation $E^-(Y)$ to denote the expectation of the outcome over the population whose input features have a joint distribution where the dichotomous neuron x_1 is set to 1 and all the other neurons $x_2, \ldots, x_J$ are fixed as they were in the observed data sample. Recall that $A0$ estimates the expected outcome $E^0(Y)$ without such manipulation in the neuron x_1 (i.e., without purposely setting x_1 to 1), hence $T1 = A1–A0$ estimates $E^+(Y) - E^0(Y)$, and can assess the effect of the neuron x_1 owing to the manipulation.

Similarly, we can set a dichotomous input neuron value to 0 for the LG and evaluate the resulting change in the outcome prediction. Take the first input neuron x_1 as an example. The pseudo vector X_i^-becomes$(0, x_{i2}, \ldots, x_{iJ})$, $i = 1, 2, \ldots, N$, and the predicted value is denoted as $\widehat{Y}_i^-$, $i = 1, 2, \ldots, N$, and $A4 = \frac{1}{N}\sum_{i=1}^{N}\widehat{Y}_i^-$. Then, among the LG, we obtain the effect estimate given as $A0 - A4$. It can be seen that $A4$ is an estimate of the expected value of the outcome $\widehat{Y}_i^-$ over the input feature distribution same as the input distribution in the observed data sample, except that the input x_1 always keeps a value of 0. Denote the expected outcome over such a feature distribution as $E^-(Y)$. Since A0, the expected value $E^0(Y)$ of the outcome over the input distribution exactly the

same as the input distribution in the observed data sample, $A0$–$A4$ is the effect estimate for this input feature.

As in the continuous input neuron case, the final effect estimate for the dichotomous input neuron case is obtained as $(A3 - A0) + (A0 - A4) = A3 - A4$. Therefore, we could find the effect estimate due to the status change in the dichotomous input features.

2.6 Pseudo datasets for multiple neurons

Suppose the study wants to know how two or more input features jointly impact the outcome. In that case, the INNA creates pseudo datasets with these inputs that have increased or decreased SDs depending on the direction of effects specified by the research questions. For example, suppose we wonder how an older man would benefit from a dietary program. In that case, this research hypothesis involves three features, including age (continuous variable), sex (dichotomous variable: male = 1, female = 0), and the calories of daily food intake (continuous variable). The corresponding pseudo dataset increases the age variable by one SD, sets the sex variable to 1, and decreases the calories of daily food intake by one SD while holding the rest of the features unchanged in the HG. The effect estimate in the HG can then be obtained by comparing the average of the predicted outcomes over the pseudo dataset with that in the original data. In the LG, the manipulations are in the opposite direction, and the effect estimate can also be obtained by comparing the average of the predicted outcomes over the LG pseudo dataset with that in the original data. The final effect estimate can be obtained by the average of the effect estimates from the HG and the LG. Therefore, the INNA is flexible for various research issues, and such an extension is intuitive and feasible.

2.7 Simulation study

Since we are comparing the performance of INNA to SHAP with local explanations, which could only interpret each single input neuron, the simulation study only evaluates the effect of a single neuron. The extension of INNA for multiple neurons is intuitive and straightforward by generating the pseudo datasets with two features modified simultaneously.

The simulated data consists of one dependent variable (Y) and five independent variables (X) according to the multivariate normal distribution. We examined Scenario I for Type-I error when Y is independent of X and Scenario II for statistical power when Y is associated with X. Under Scenario I, we expect INNA to generate the effect estimates around zeros. In contrast, the INNA should yield non-zero estimates of predictive effects under Scenario II.

The sample size in medical research is much less than one thousand due to expensive measures in the predictors

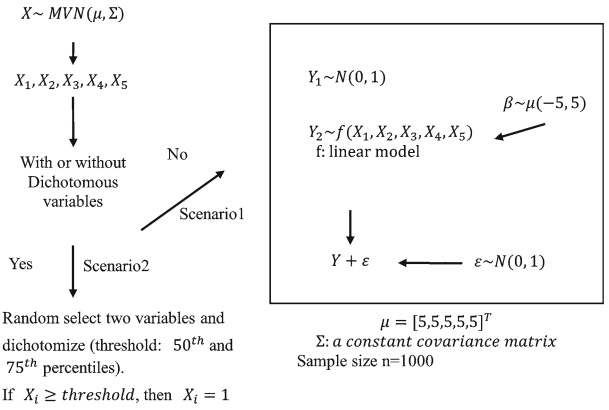


Fig. 2 Process of the simulated data

and outcomes. Therefore, the sample size is 1000 for simulation studies. The mean vector is indicated in Fig. 2, and the covariance matrix is displayed in Appendix 1. We considered continuous and categorical predictors with two different scenarios: (1) all five Xs are continuous; (2) two of the five Xs are dichotomous. Note that two variables are randomly selected. We dichotomized the two variables according to the 50th and 75th percentiles. If the variable is higher than the threshold, the transformed value is 1; otherwise, it is 0.

For continuous input neurons, we normalize them to expedite the training process and avoid the situation of falling into the optimal local maximum or minimum, resulting in subpar interpretation and prediction. Details of the simulation study are in Fig. 2.

The INNA could fit a piece-wise non-linear activation function (ReLU) or a non-linear activation function like sigmoid. Therefore, the INNA is flexible for various models attributable to the nature of ANN. To illustrate the interpretation ability of the INNA, we fine-tuned the ANN with Relu to approximate the beta coefficients of the GLM. Therefore, the simulation compares the effect estimates of the INNA and GLM models as exceptional cases. The simulation does mean that the INNA is only valid for the linear model interpretation.

The model is constructed using 80% of the training dataset. The remaining 20% of the dataset serves as the testing data. Since the result of this study is based on continuous variables, the predictive ability is evaluated by the mean square error (MSE).

2.8 Analysis of real data

We applied the INNA to three different datasets. The US Census Bureau collected the first dataset on housing in Boston, Massachusetts, which includes 506 pieces of information and 14 housing-related variables [38]. In the following, the variable names in the dataset are presented in parentheses. The

median owner-occupied housing value (medv) is the output variable. There are five predictors: (1) the per capita crime rate of each town (crim); (2) the presence or absence of the Charles River on the boundary (chas); (3) the average number of rooms per dwelling (rm); (4) the lower population status percentage (lstat); (5) the weighted average of the distance to the five Boston employment centers(dis).

The second dataset is from ACME Insurance on access to affordable health insurance for clients across the United States [39]. The outcome variable is insurance costs. Predictive variables and the coding of the variables presented in the parenthesis were gender (SEX), weight (BMI), number of children (Children), whether or not they smoked (Smoker), and region of the United States (Region).

The last dataset is from the Bukavu blood pressure observation data [40], which includes 10,866 samples and ten variables. The Bukavu dataset focuses on predicting diastolic and systolic blood pressure. The predictive variables include age (AGE), sex (SEX), body mass index (BMI), waist circumference (WC), year of survey (Year), region (Region), whether taking antihypertensive medication (ANTI_HTN) and pulse rate (PULSE). Variable descriptions for the three datasets are summarized in Table 1.

We used the logarithmic transformation for skewed independent variables and standardized them because this study uses two open-ended data from different sources. In addition, categorical predictors are transformed into dummy variables. This transformation step ensures the convergence of the INNA model. The effect estimates obtained from the INNA model are compared with the beta coefficients of the GLM.

In the Boston data, only the variable "crim" is log-transformed and standardized, and the distribution of all variables is shown in Appendix 1.1. Considering the second ACME data, the distribution of premiums is skewed. Thus, premiums were log-transformed, and the variables in the data were normalized (Appendix 1.2). In the last Bukavu data, missing data will be excluded. The missing BMI and pulse rates are 3.9% and 2.0%, respectively. As a result, the sample size is 10,428, and the distribution is in Appendix 1.3.

In these real-life examples, we included the SHAP method with local explanations to compare the effect of the Shapley value to the results of the INNA model. We expect the predictive impact of the two methods to be comparable. Unlike the INNA that directly generates the population-mean level effect, the Shapley value analysis implements the 'iml' package in R and outputs the individual level effect as the default. After the package derived every individual's predicted results, we could calculate the population average. However, this process is very time-consuming. Thus, we only present the first individual's results and ignore the rest of the subjects in the data.

The parameter settings for INNA were approximately the same as those for the simulated data, except the repetition parameter (repetition:20). All the statistical analyzes and simulations were conducted by the R Software (4.0.3) [41]. An R function of the INNA is freely available with an example dataset for practice. Researchers could effortlessly implement the INNA function for various datasets from different research fields.

This INNA function requires the installation of the "devtools" package and the "neuralnet" activation function Relu. The next step brings in the analysis data, which must be in a matrix format and convert the categorical and continuous variables. The following symbols are the parameters in the INNA function:

- 'Y': the variable name of the output variable in the dataset.
- 'hidden': the number of neurons in the hidden layers.
- 'type': data type of input neurons 1: continuous; 0: binary.
- 'threshold': back-propagation delivery boundary. Default is the maximum value of the outcome variable.
- 'rep':the number of times to retrain the model. The default is 20.
- 'locate':path of the dataset. The default is the current path location.
- 'act.fct':the choice of the activation function. The default is "Relu" such that the results approximate the GLM. Other settings of the activation function could fit other non-linear models.
- 'algorithm':choice of the back-propagation algorithm. The default is "rprop + ".
- 'learningrate': back-propagation learning rate. The default is 0.1.
- 'stepmax': the maximum number of iterations. The default is 10,000.

3 Results

3.1 Simulation studies

According to the computer simulations with 1000 repetitions, Fig. 3 presents the results when all predictors are continuous with six neurons in one hidden layer. Other settings with more hidden neurons or two hidden layers with different numbers of neurons show a similar comparison.

In addition, when two predictors are dichotomous, the results are also similar (not shown due to page limits). Figure 3A and the upper section of Table 2 display Hypothesis I, assuming the null hypothesis that X and Y are independent (i.e., the frequency of effects not equal to zero indicates Type-I errors). Figure 3B and the lower section of

Table 1 Variable information on Boston, ACME, and Bakura data

BOSTON		ACME		Bakura	
Attributes	Values	Attributes	Values	Attributes	Values
CRIM %	0.01–88.98	AGE	18–64	Age	18–113
ZN %	0–100	SEX	0: female, 1: male	Sex	0: female, 1: male
INDUS %	0.46–27.74	BMI	15.96–53.13	BMI	10.54–58.05
CHAS	0: No, 1: Yes	Children	0–5	WC	24–139
NOX (10^{-6})	0.39–0.87	Smoker	0: No, 1: Yes	Year	0:2012, 1:2016
RM	3.56–8.78	Region	0: northeast, 1: northwest, 2: southeast, 3: southwest	REGION	0:city, 1:rural
AGE %	2.90–100	Charges $	1122–63,770	ANTI_HTN	0:No, 1:Yes
DIS	1.13–12.13			PULSE	30.5–141.0
RAD	1–24			SBP	42.67–230.0
TAX	187–711			DBP	25.0–155.67
PTRATIO	12.6–22.0				
Black	0.32–396.9				
LSTAT %	1.73–37.97				
MEDV	5–50				

Table 2 show Hypothesis II assuming the alternative hypothesis (i.e., the effect frequency not equal to zero represents statistical power).

In Fig. 3A and B, the *X*-axis indicates the five input features, and the *Y*-axis represents the Box plot of the 1000 differences of simulation results in the effect estimates between INNA and GLM. We can see that the two models have very similar effect estimates close to zero. The coefficient difference is zero when the two methods yield the exact estimate. Therefore, if the coefficient difference is negligible, our new approach could provide excellent effect estimates.

The simulation results suggest that the difference in the coefficients estimated by the INNA and GLM is expected to be 0. Thus, the INNA population-mean level effect estimates are very close to the beta coefficients (effects of the predictors) derived by the GLM. The difference between the two methods falls between − 0.5 and 0.5. We observed similar effect estimates for the five input neurons with 4–9 neurons in one hidden layer. Under the non-linear situation of neural networks with two hidden layers, only a few effect estimates have more considerable differences. Most effect estimates are close to the beta coefficients of the GLM.

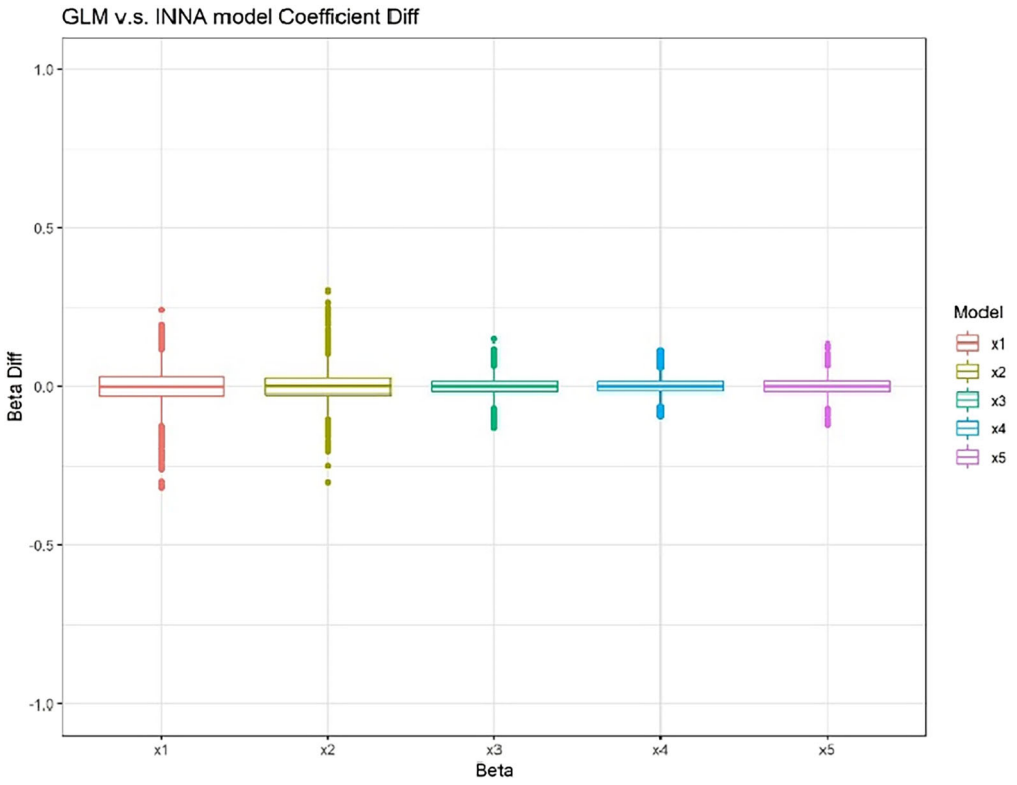
3.2 Real-life analyzes

We summarized applications to three real datasets in Table 3. The first two columns of Table 3 display the feature and beta estimates of the GLM using the Boston dataset. The INNA fits only the significant regression coefficients for comparisons.

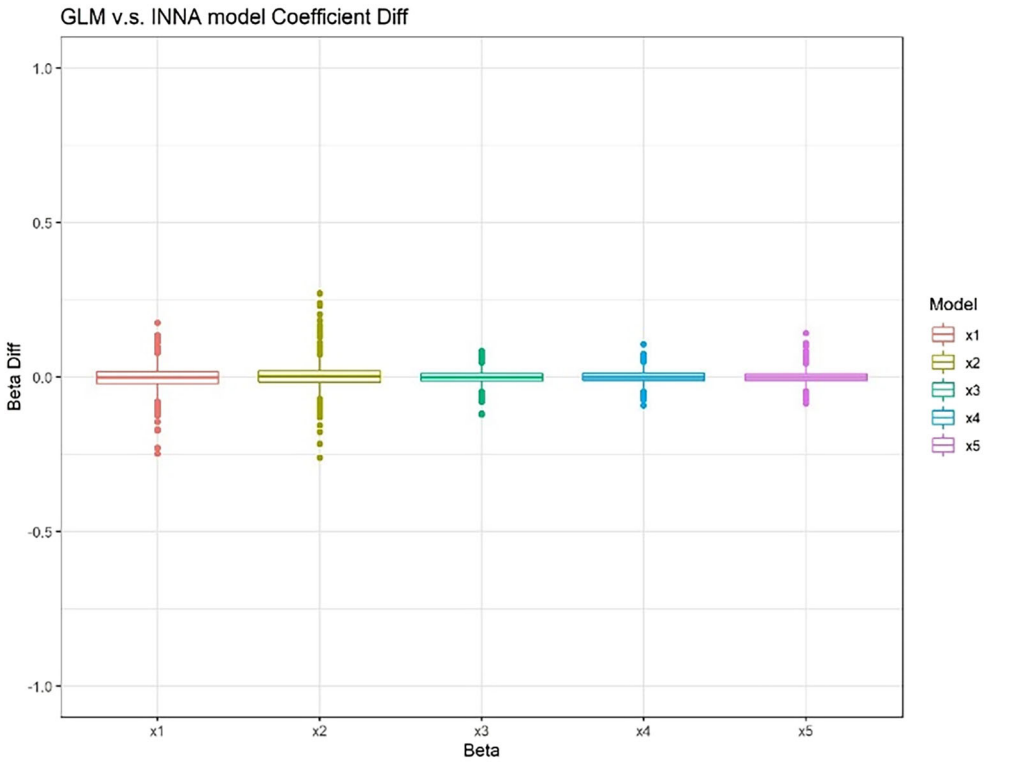
The INNA tried various numbers of hidden neurons in one hidden layer (see Appendix 2). Figure 4A demonstrates the best prediction performance (MSE = 15.7). Therefore, the INNA could approximate the beta coefficients of the GLM with the activation function of ReLU. The two methods show a very similar interpretation of the effect estimates. Figure 4B shows the effect of SHAP for the first subject in the data. The effect direction and magnitude are comparable to the results estimated by the INNA.

The GLM results for the ACME insurance dataset are in the third and fourth columns of Table 3. Figure 5(A1) and (A2) show the comparisons between the INNA and GLM. The INNA has six or seven neurons in one hidden layer. With dummy variables, the effect estimates of the INNA are comparable to the results of the GLM. The directions of the effects are consistent with minor differences for significant predictors. Insignificant predictors have small effects, and the directions may be different. We also fitted INNA with two hidden layers (see Appendix 3). The optimal model is one hidden layer with six hidden neurons (MSE = 0.14). With the optimal model selection, Fig. 5B shows the effect of SHAP for the first subject in the data. The results are very close to the coefficients estimated by INNA. For example, "age = − 0.3" means that the input feature "age" has a negative impact on the predicted value. The higher absolute value indicates a more substantial effect on the outcome prediction.

Finally, the GLM results of the Bukavu dataset are shown in the fifth and sixth columns of Table 3. Figure 6 analyzes the systolic blood pressure (SBP) in 6(A1) and diastolic blood



(A) Hypothesis I



(B) Hypothesis II

Fig. 3 Comparisons of the INNA and GLM coefficients for five input neurons with six neurons in one hidden layer under Scenario1

Table 2 Comparisons of the INNA and GLM coefficients for five input neurons and one hidden layer of 6 neurons for Scenario I and II

Hypothesis I	× 1	× 2	× 3	× 4	× 5
INNA	0.001 ± 0.079	0.001 ± 0.079	− 0.001 ± 0.048	0.001 ± 0.047	− 0.001 ± 0.047
GLM	0.002 ± 0.059	− 0.001 ± 0.058	− 0.001 ± 0.039	0.000 ± 0.039	− 0.002 ± 0.038
Hypothesis II	× 1	× 2	× 3	× 4	× 5
INNA	− 0.059 ± 2.830	− 0.126 ± 2.857	0.143 ± 2.836	0.065 ± 2.834	0.039 ± 2.904
GLM	− 0.057 ± 2.847	− 0.128 ± 2.876	0.145 ± 2.846	0.065 ± 2.842	0.039 ± 2.914

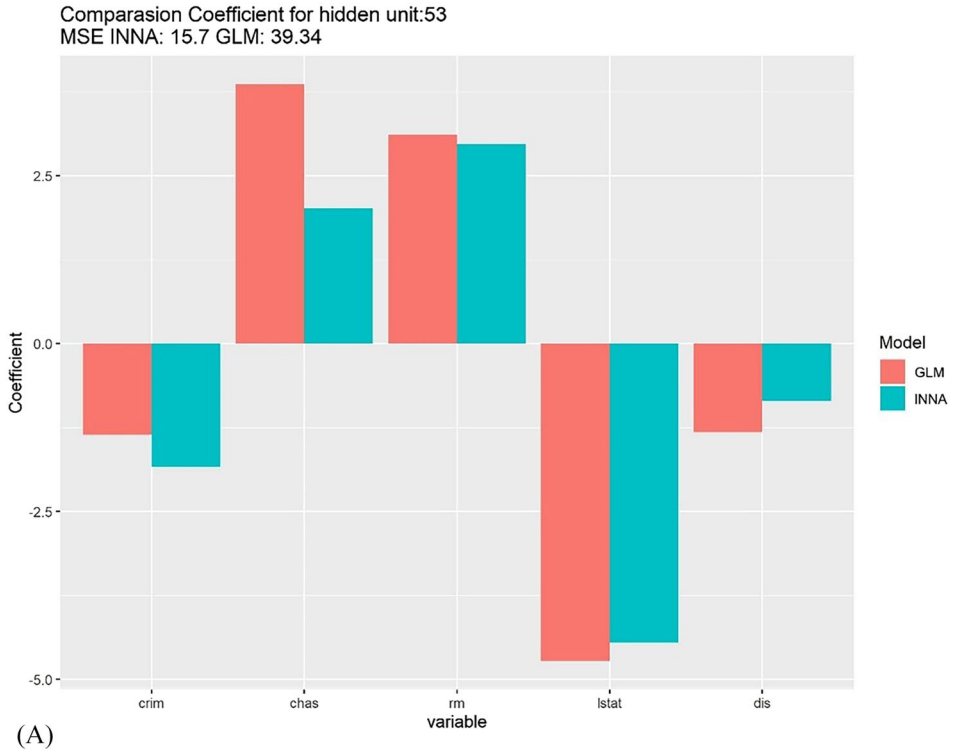
Table 3 Coefficients and the p-values of the regression model (*$p \leq 0.05$, **$p \leq 0.01$, ***$p \leq 0.001$)

BOSTON		ACME		Bakura	
Variables	coefficient	Attributes	coefficient	Attributes	coefficient
CRIM	− 1.35***	Age	0.49***	SBP	
CHAS	3.86***	Sex(male)	− 0.07***	AGE	7.12***
RM	3.11***	BMI	0.08***	SEX	4.41***
LSTAT	− 4.72***	Children(1)	0.15***	BMI	− 0.34
DIS	− 1.32***	Children(2)	0.29***	WC	2.36***
		Children(3)	0.25***	Year	0.76*
		Children(4)	0.50***	REGION	− 4.90***
		Children(5)	0.42***	ANTI_HTN	13.49***
		Smoke(yes)	1.54***	PULSE	0.33
		Region(northwest)	− 0.05	DBP	
		Region(southeast)	− 0.15***	AGE	3.27***
		Region(southwest)	− 0.13***	SEX	0.98***
				BMI	0.50*
				WC	1.60***
				Year	0.16
				REGION	− 2.68***
				ANTI_HTN	7.13***
				PULSE	1.14***

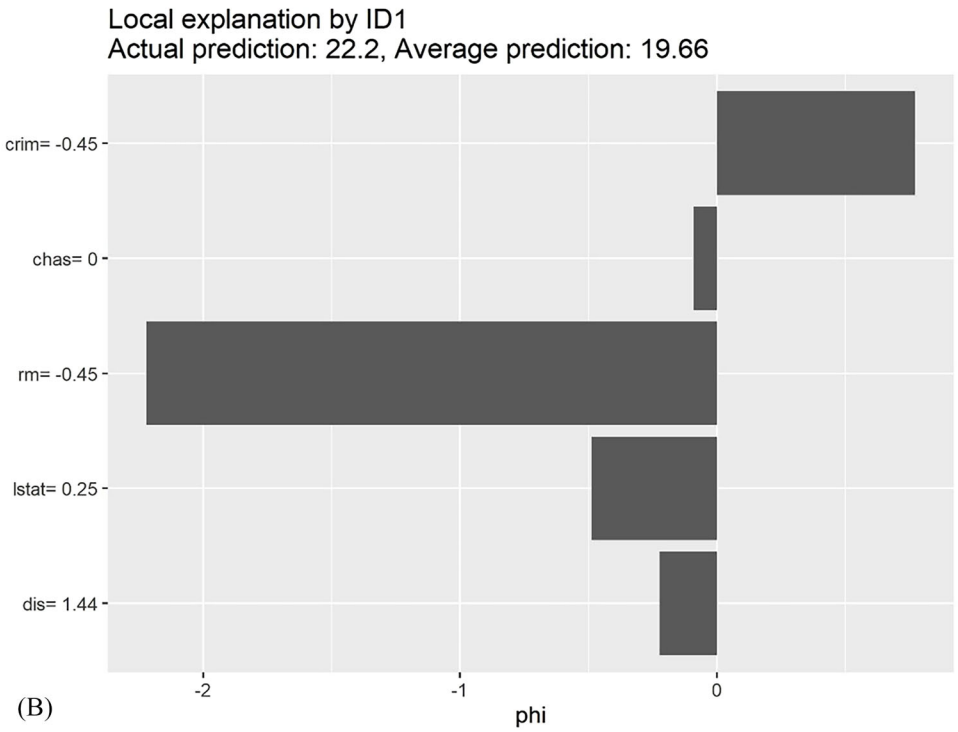
pressure (DBP) in 6(A2). The two methods yielded similar effect estimates under other parameter settings of the INNA (see Appendix 4). The optimal result of SBP is one hidden layer with six neurons (MSE = 261.3). Regarding the DBP, the smallest MSE is 112.39 with two hidden layers (5 and 3 neurons in the first and second hidden layers). Figure 6(B1) and (B2) show the SHAP effect for the first subject in the data. In the case of the SBP and DBP analyzes, the first participant's information reveals the overall effect estimates of SHAP values in close agreement with the INNA estimates.

4 Discussion

This research proposed a novel study design in machine learning by generating pseudo datasets with perturbations in the input features. Our approach is an intuitive algorithm to estimate the effect of one or multiple input neurons. It measures the expected mean difference in the output node due to one SD change in the continuous neurons or one vs. 0 changes in the dichotomous neurons while keeping the other inputs constant. Therefore, the INNA solely evaluates the adjusted mean level change in the outcome variable. Most importantly, the INNA could assess the simultaneous effect

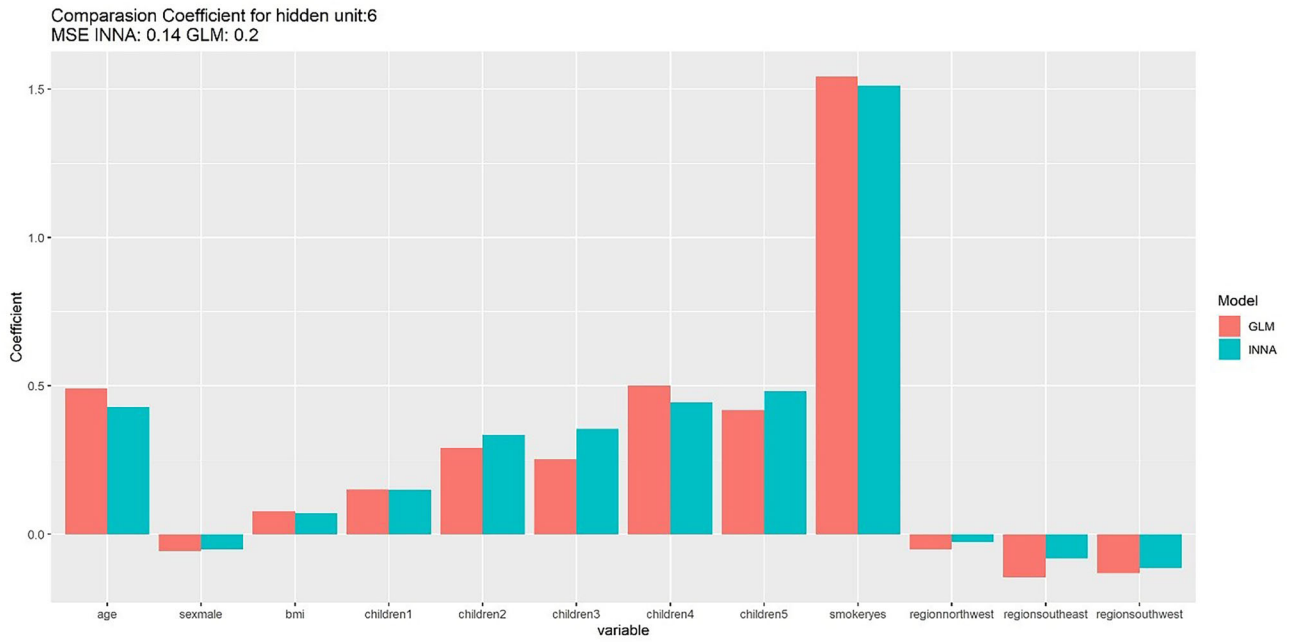


(A)

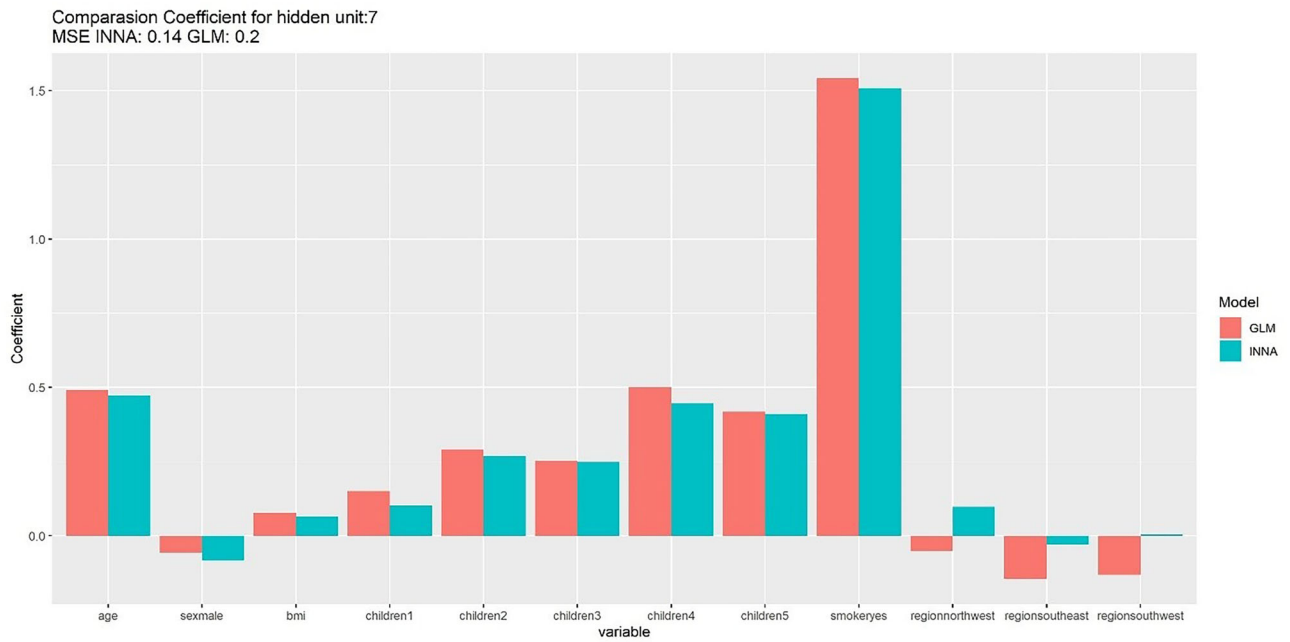


(B)

Fig. 4 Comparisons of the Boston analysis between linear regression, SHAP, and INNA with two hidden layers with five and three hidden neurons



(A1) six neurons in one hidden layer for the INNA



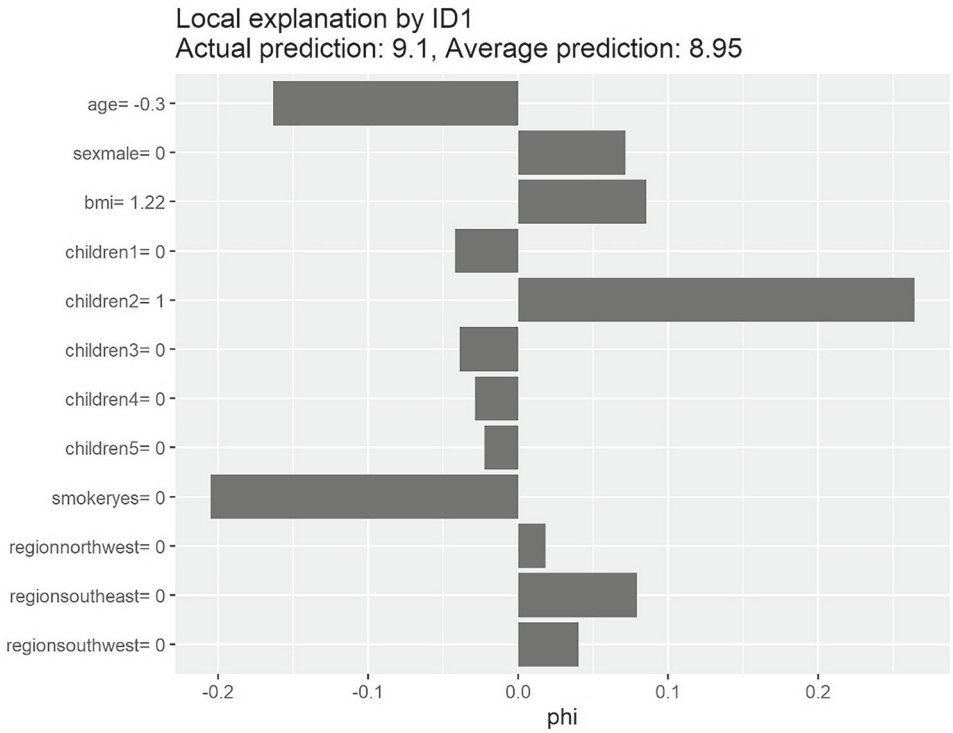
(A2) seven neurons in one hidden layer for the INNA

Fig. 5 Comparison of the ACME analysis between linear regression, SHAP, and INNA

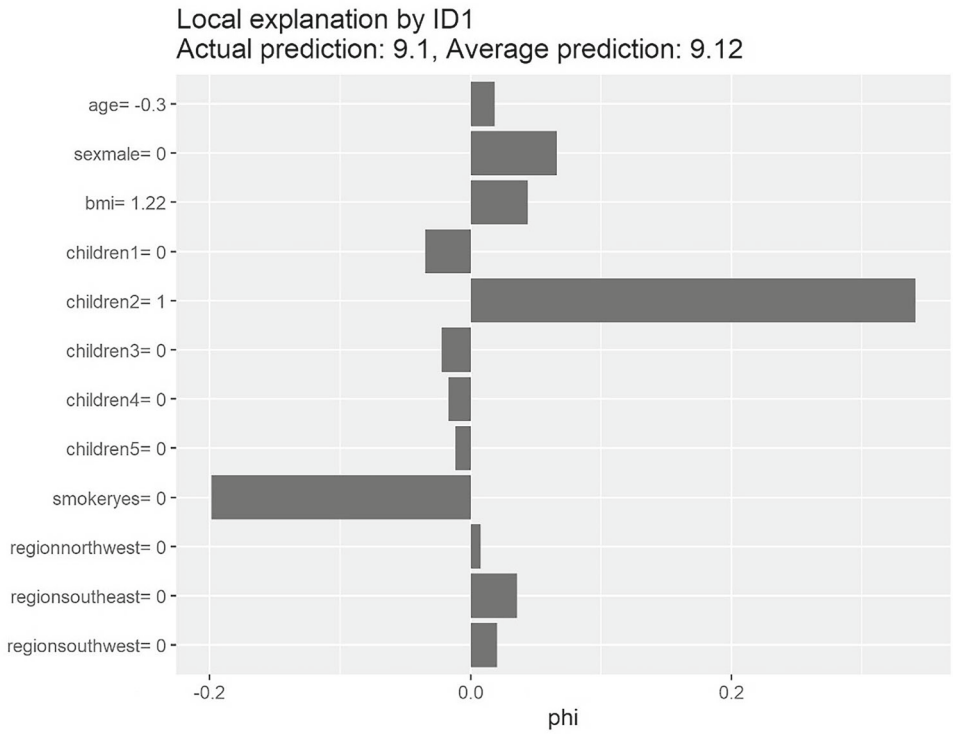
from multiple input neurons by changing two or more features in the pseudo dataset. If the pseudo datasets modify only one subject's input features, our new algorithm could also estimate the individual's effect.

Our approach is the first algorithm in data science, and more extensions are expected. For example, this algorithm could serve only as a preprocessing step, and any downstream classifiers or regressions models can be applied to the tasks,

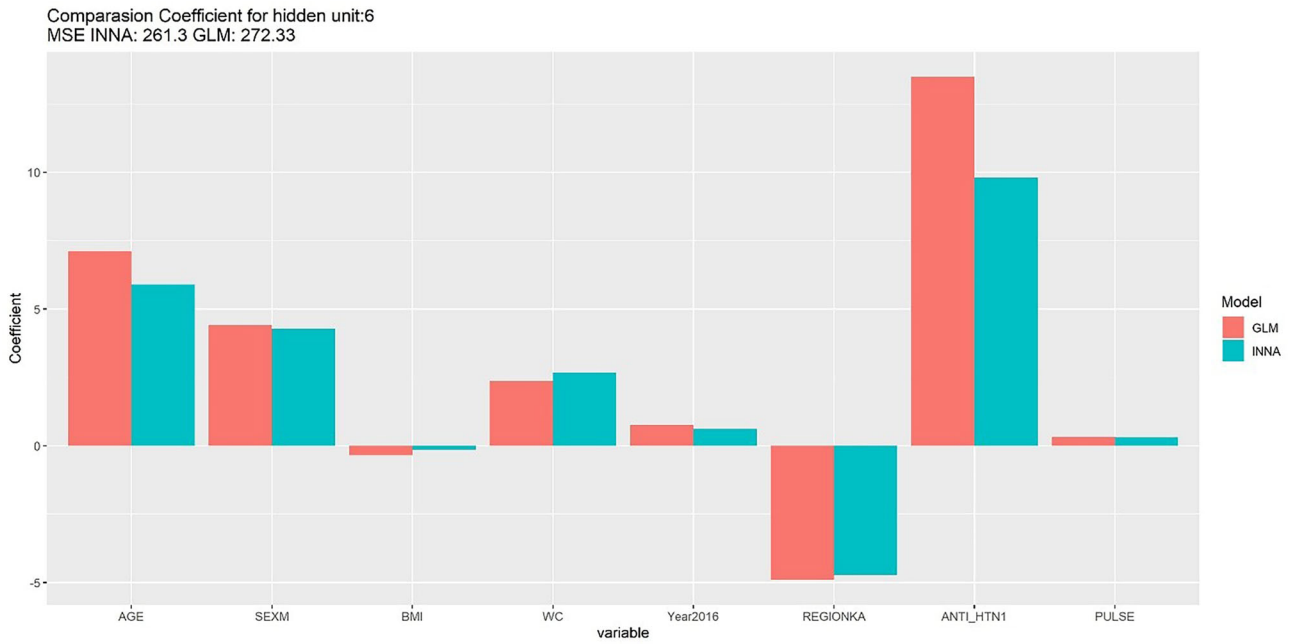
Fig. 5 continued



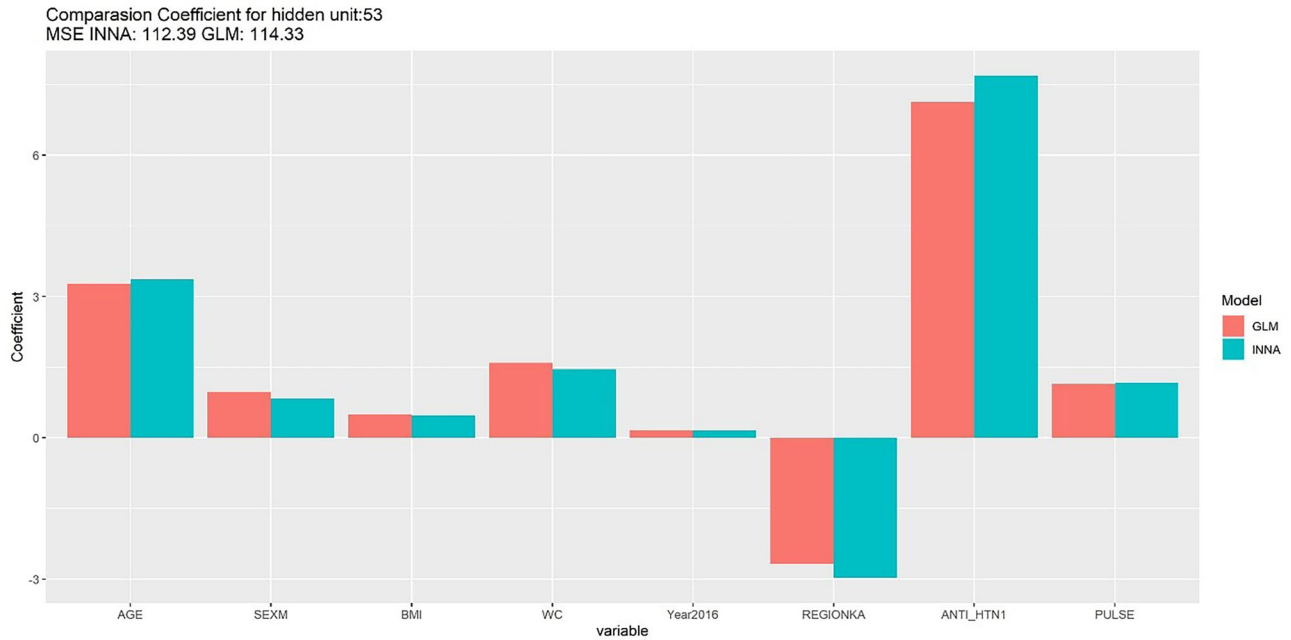
(B1) six neurons in one hidden layer for the SHAP



(B2) six neurons in one hidden layer for the SHAP



(A1) SBP analysis: six neurons in one hidden layer for the INNA



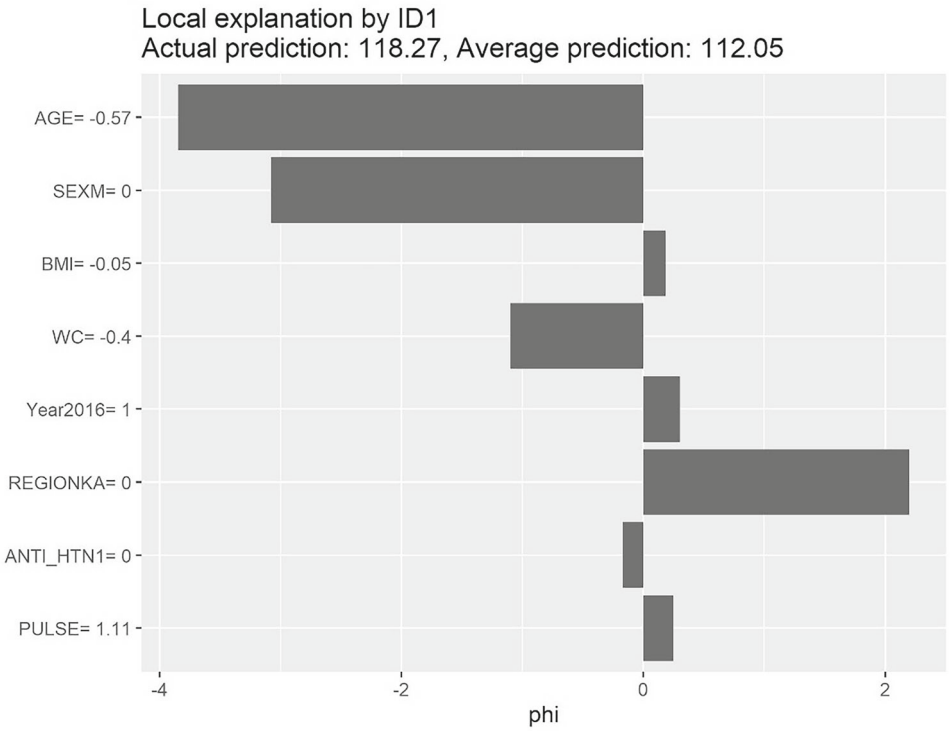
(A2) DBP analysis: five and three neurons in two hidden layers for the INNA

Fig. 6 Comparisons of the Bukavu analysis between linear regression, SHAP, and INNA

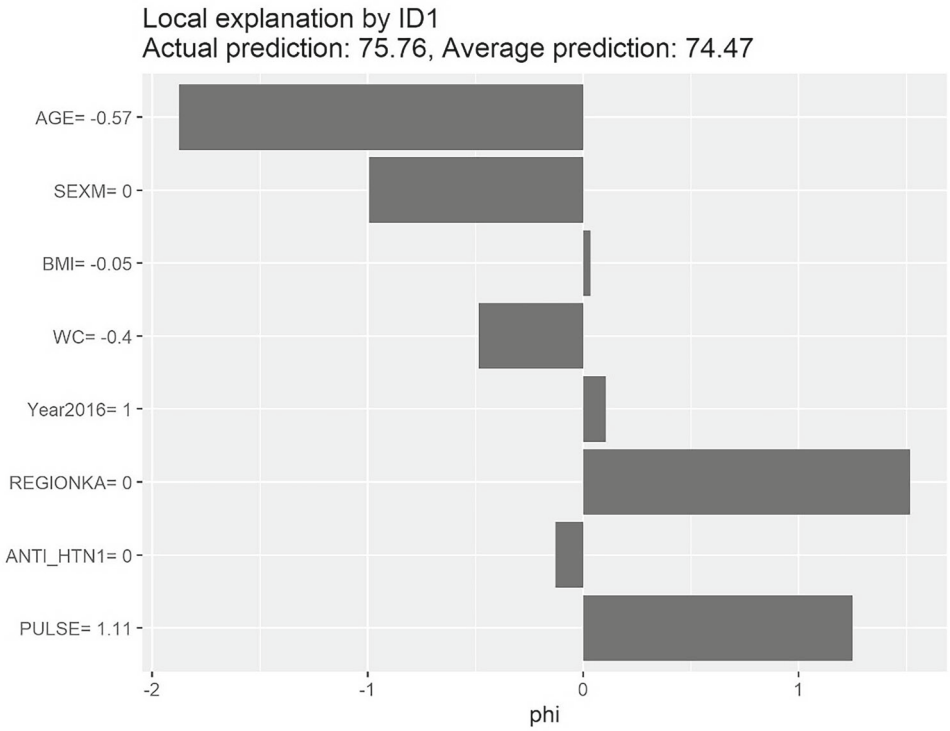
or the proposed method serves as both preprocessing step and regression model.

According to computer simulations, the INNA validly interprets the predictive variables. The INNA and GLM effect estimates are very close in the illustrations using three real-life examples. The prediction performance of INNA is generally better than the GLM with two hidden layers. Therefore, we suggest the INNA for the analysis of the actual data.

SHAP differs from INNA because it does not have an overall population-mean-level estimate of the coefficients.



(B1) SBP analysis: six neurons in one hidden layer for the SHAP



(B2) DBP analysis: five and three neurons in two hidden layers for the SHAP

Fig. 6 continued

The SHAP package outputs only individual-level impact, and the individual differences may vary significantly. The average SHAP value of each variable over all samples may be similar to the population-mean level effect estimation of the proposed pipeline. However, this step in SHAP requires a tremendous computational time. In contrast, our approach is intuitive and not computation-intensive since the INNA only fits/trains the model once. In addition, the INNA proposes a novel concept of pseudo datasets with variations of one standard deviation in the input features. As a result, the INNA provides a parametric-free algorithm and can effortlessly accommodate all types of data structure or distribution in neural networks.

The simulation study only focuses on the INNA and GLM. Although other methodologies might interpret the neural networks, the GLM is the reference model in this research since it is the conventional interpretable model.

We prepare a free R function of the INNA and provide an example dataset for illustrations. Thus, researchers could quickly implement this novel algorithm in medical science, public health, or various research fields.

In this research, the simulation study and real-life examples of the INNA aimed for continuous outcomes. However, in the future, the INNA could be easily extended for categorical outcomes by changing the activation functions in the output layer, such as the sigmoid function. Besides, the loss function must be suitable for dichotomous output features like cross-entropy. We could compare the INNA to the logistic regression model in this setting and see if the two methods yield similar and consistent effect estimates. This topic requires much effort in future research.

Instead of comparing the effect estimates of generalized linear models (GLM) to the INNA, future research could assess the similarities and differences using the random forest and XGBoosting machine.

The new algorithm INNA aims to accurately obtain the population-mean level effect estimate of every input neuron in a neural network (continuous, ordinal, or nominal). The results provide a simple yet clear picture of how the input neurons change the population-mean level predictive probability of a specific category in the output neurons.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s41060-024-00526-9.

Acknowledgements None.

Author contributions YCC conducted the analysis and prepared the first draft. YHC Edits the manuscript and provides critical comments. CYG proposed the research methodology, supervised the project, and finalized the manuscript and revisions. All authors read and approved the final manuscript.

Funding Open Access funding enabled and organized by National Yang Ming Chiao Tung University. The National Science and Technology Council supports this work. Grant ID: 112-2118-M-A49-003.

Availability of data and material "BP_Data.csv" is included as supplement materials.

Code availability An R Code (INNA code.R) that implements the proposed approach is included as supplement materials.

Declarations

Conflict of interest The authors declare no conflicts of interest related to this article's subject matter or materials.

Ethical approval Our study did not require ethical board approval because It is a computer simulation study.

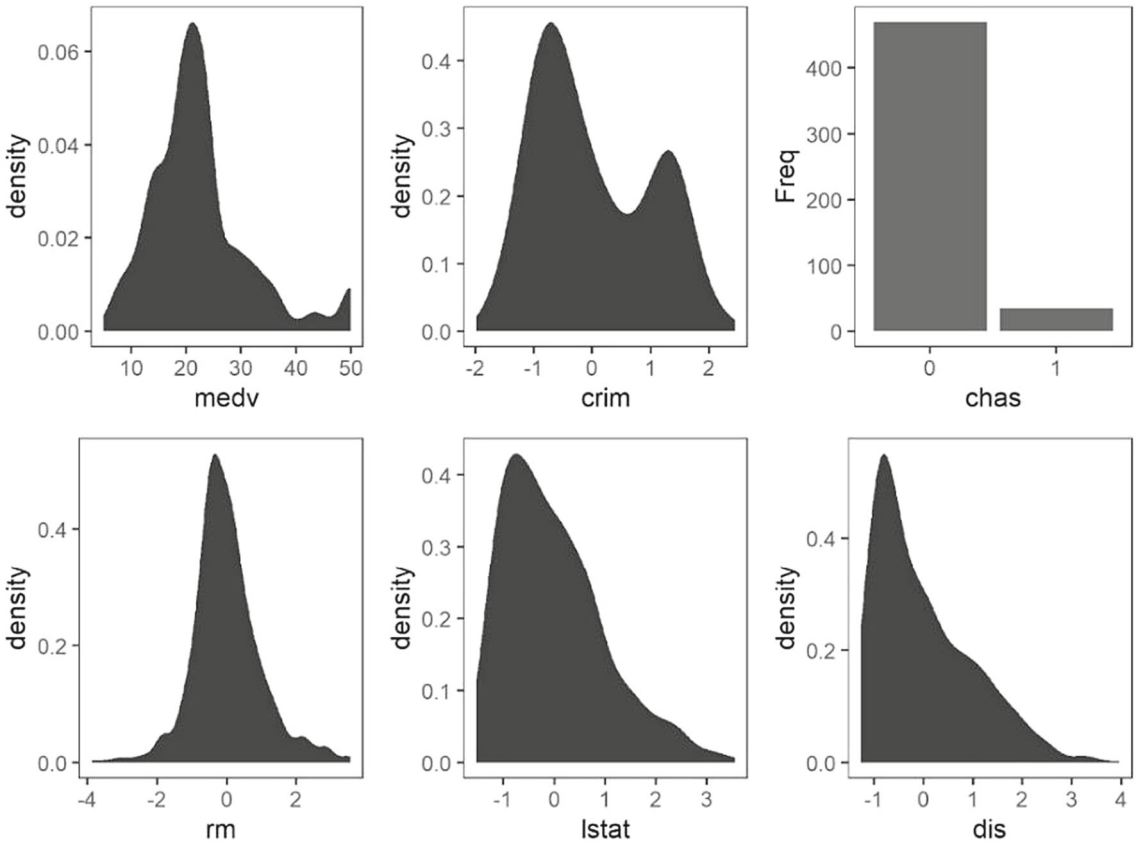
Consent to participate Not applicable. It is a computer simulation study with open-source datasets.

Consent for publication Not applicable. It is a computer simulation study.

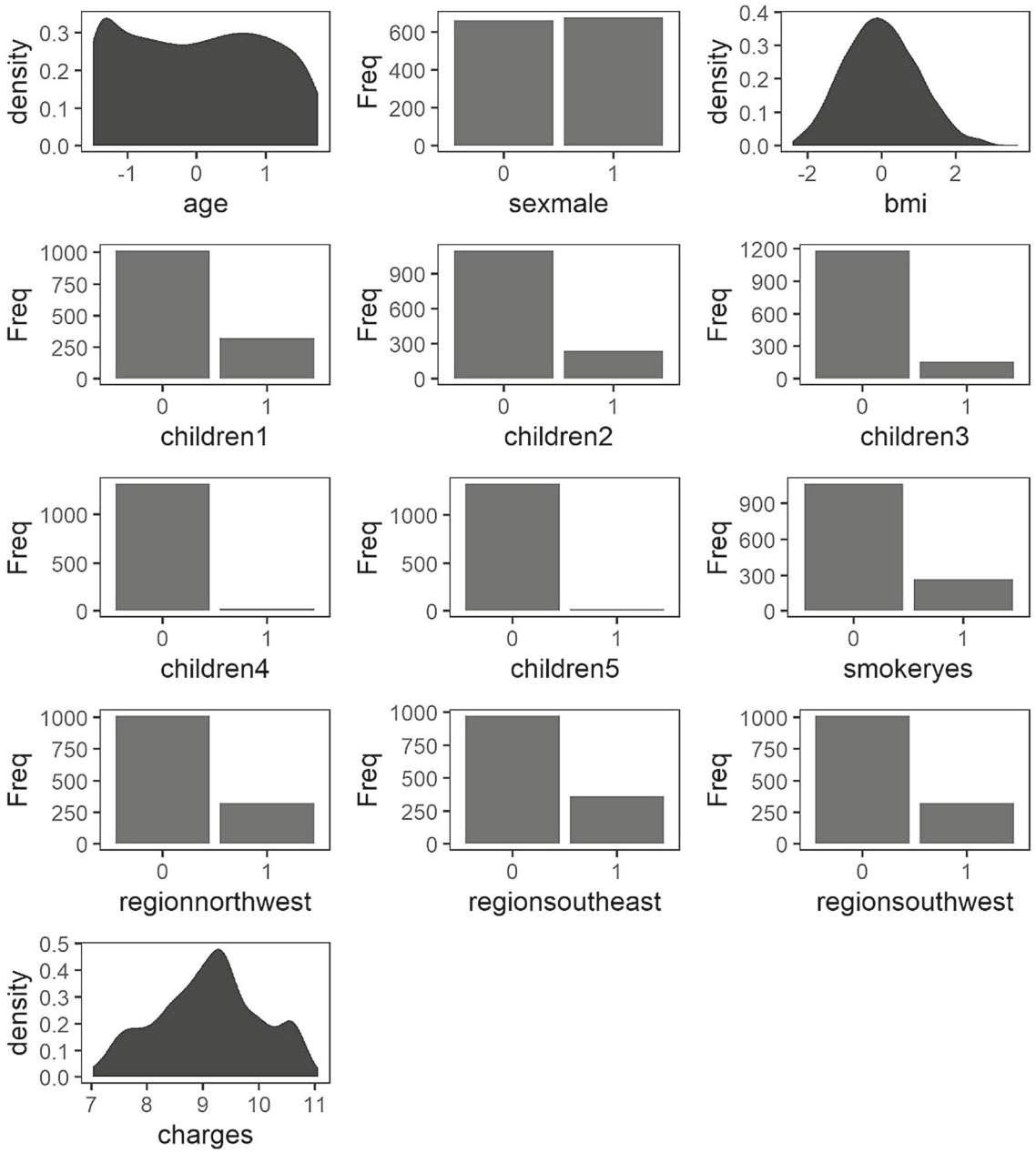
Appendix 1: Covariance matrix

	$X1$	$X2$	$X3$	$X4$	$X5$
$X1$	1.0	0.8	0.3	0.3	0.3
$X2$	0.8	1.0	0.3	0.3	0.3
$X3$	0.3	0.3	1.0	0.3	0.3
$X4$	0.3	0.3	0.3	1.0	0.2
$X5$	0.3	0.3	0.3	0.2	1.0

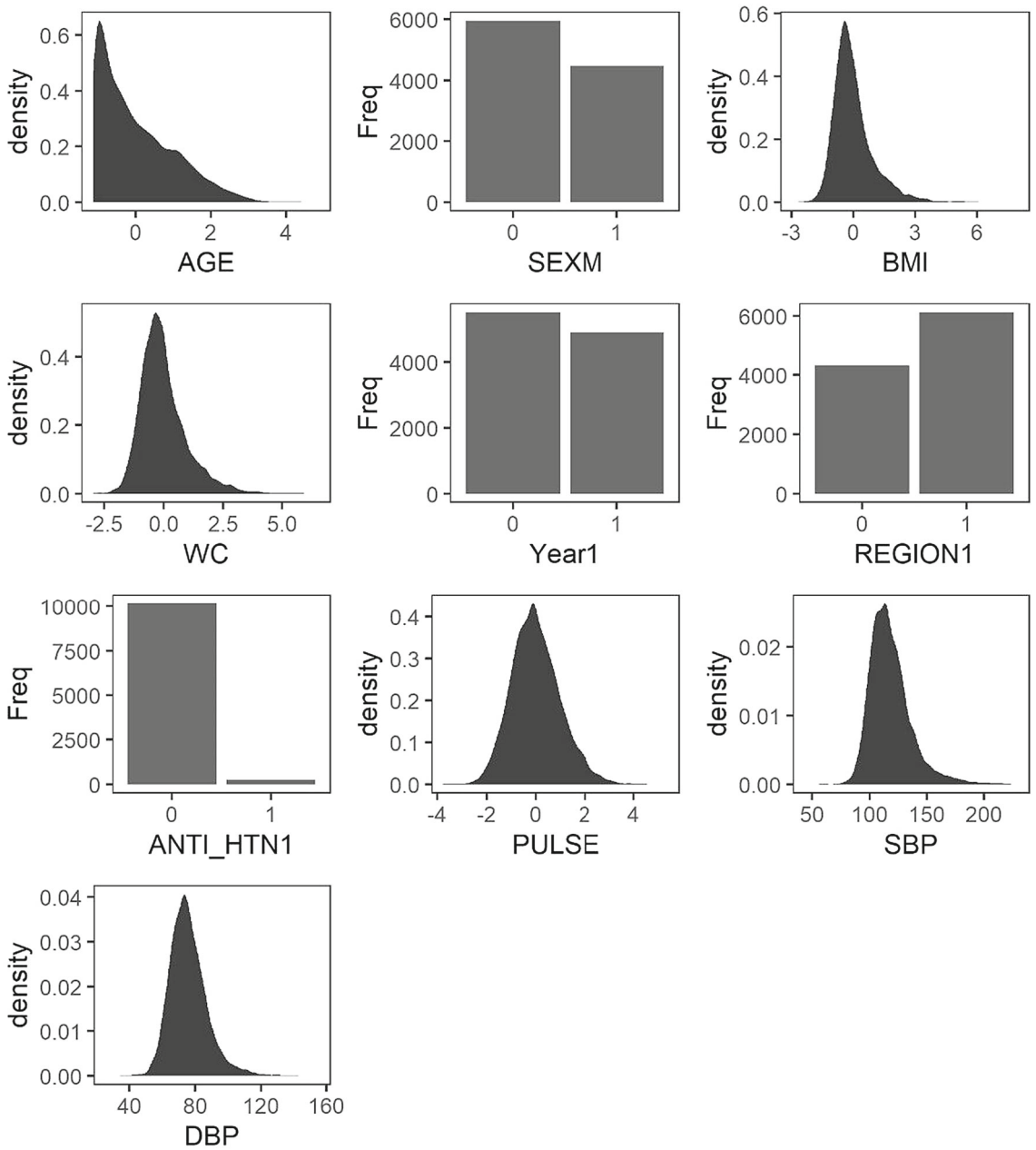
Appendix 1.1:. The distribution pattern of the Boston variable after transformation



Appendix 1.2:. The distribution pattern of the ACME variable after transformation



Appendix 1.3



Appendix 2

See Figs. 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18.

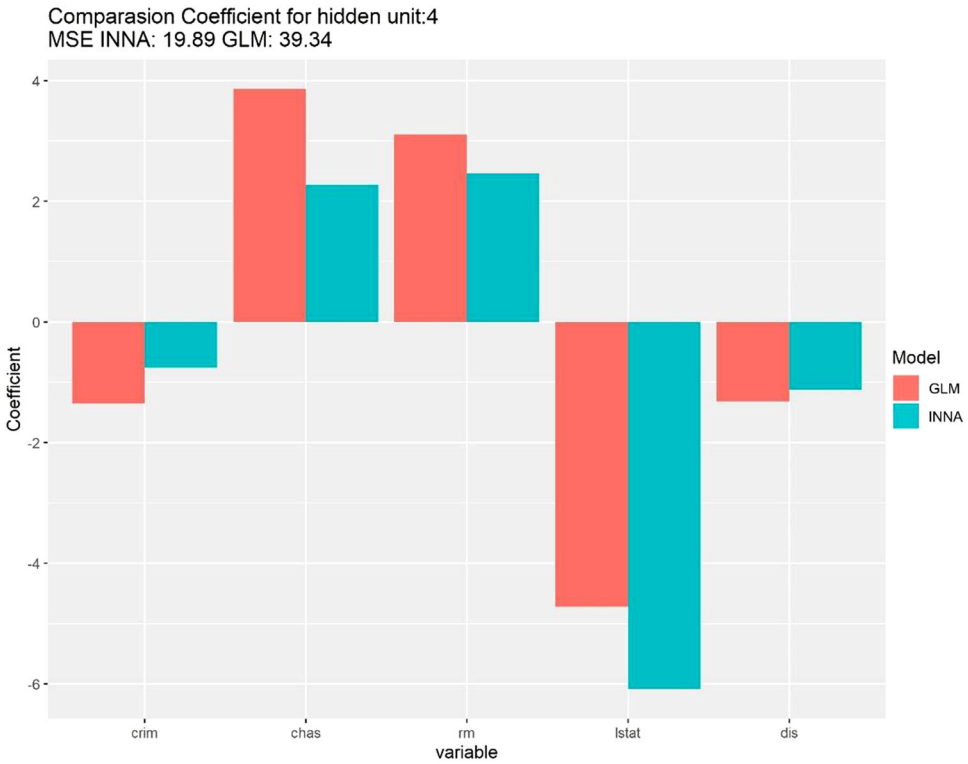


Fig. 7 Comparison of INNA and GLM coefficients for Boston's one hidden layers of four nodes

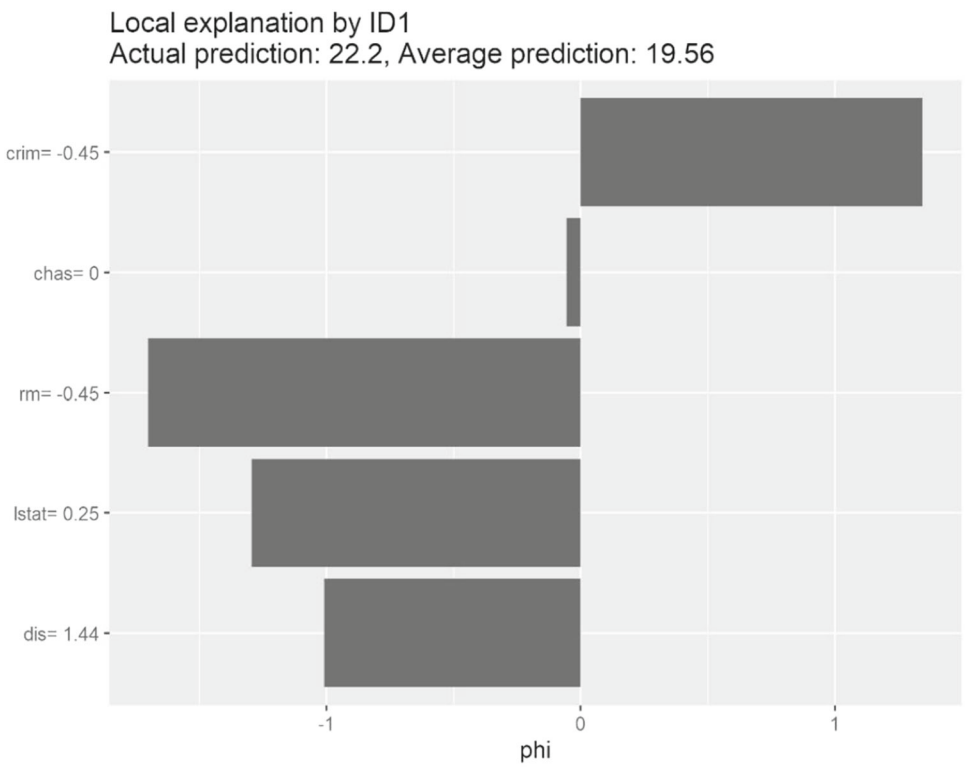


Fig. 8 SHAP effect of one hidden layer's INNA four nodes from the Boston

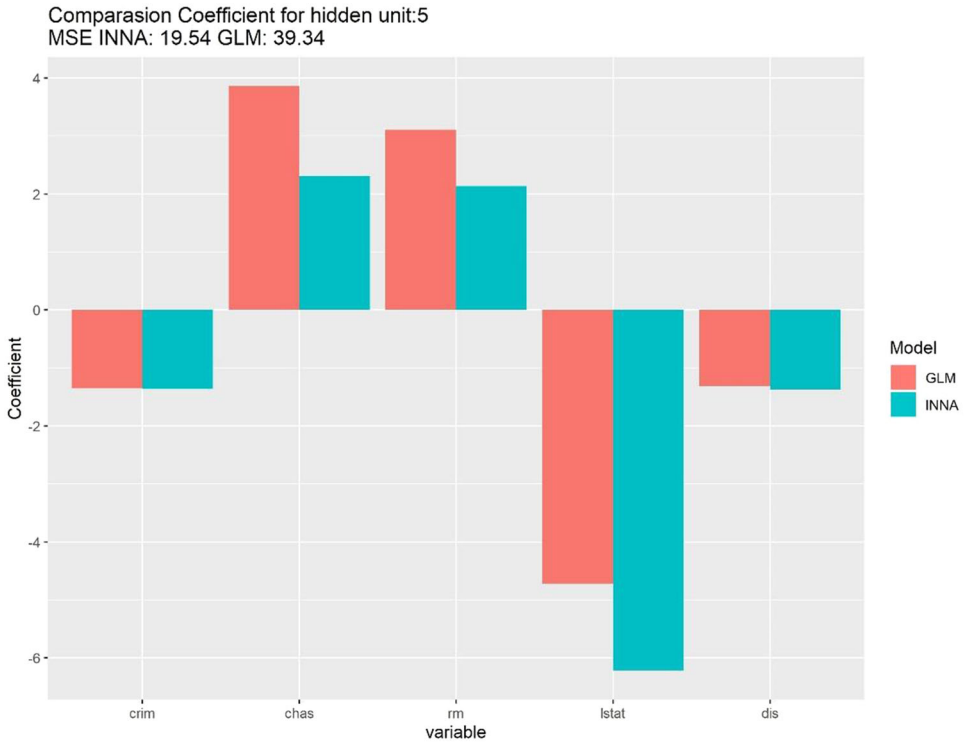


Fig. 9 Comparison of INNA and GLM coefficients for Boston's one hidden layers of five nodes

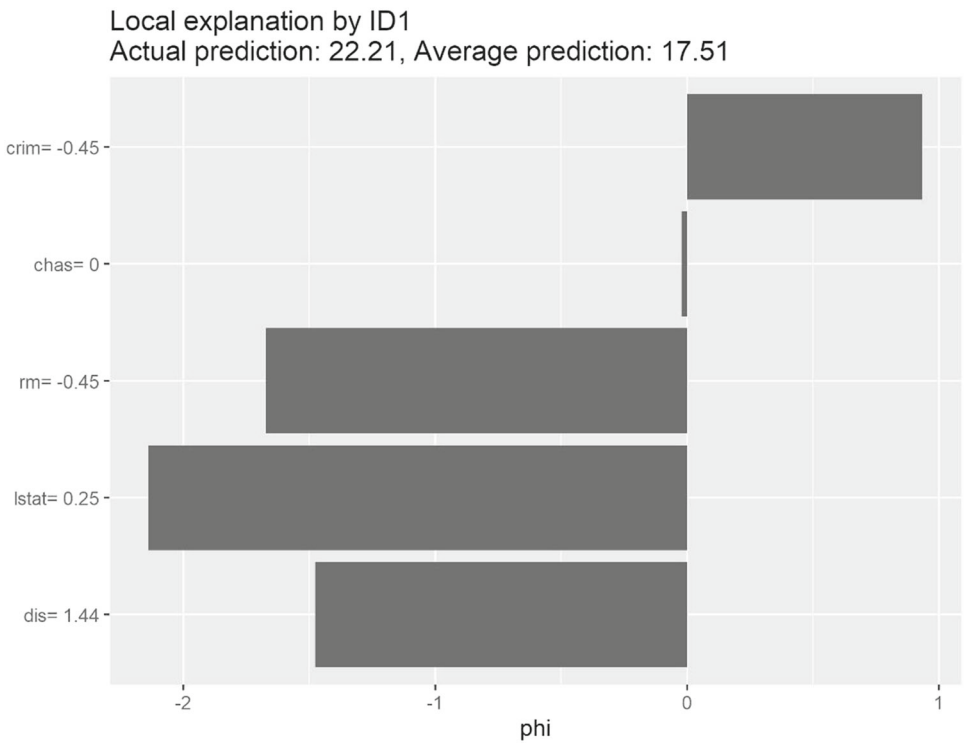


Fig. 10 SHAP effect of one hidden layer's INNA five nodes from the Boston

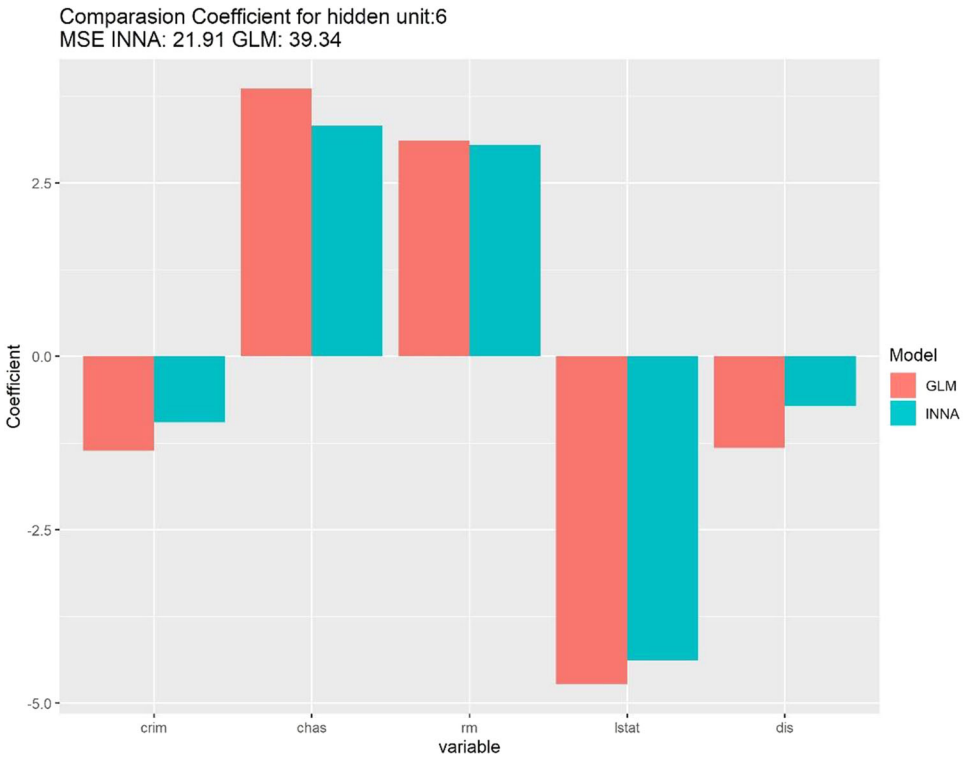


Fig. 11 Comparison of INNA and GLM coefficients for Boston's one hidden layers of six nodes

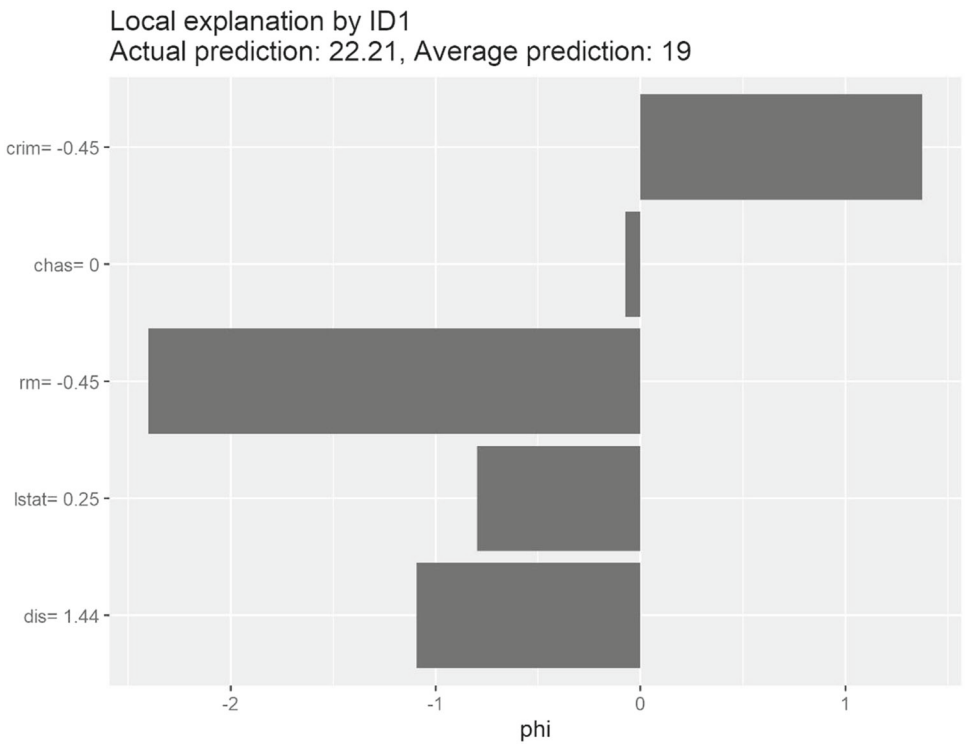


Fig. 12 SHAP effect of one hidden layer's INNA six nodes from the Boston

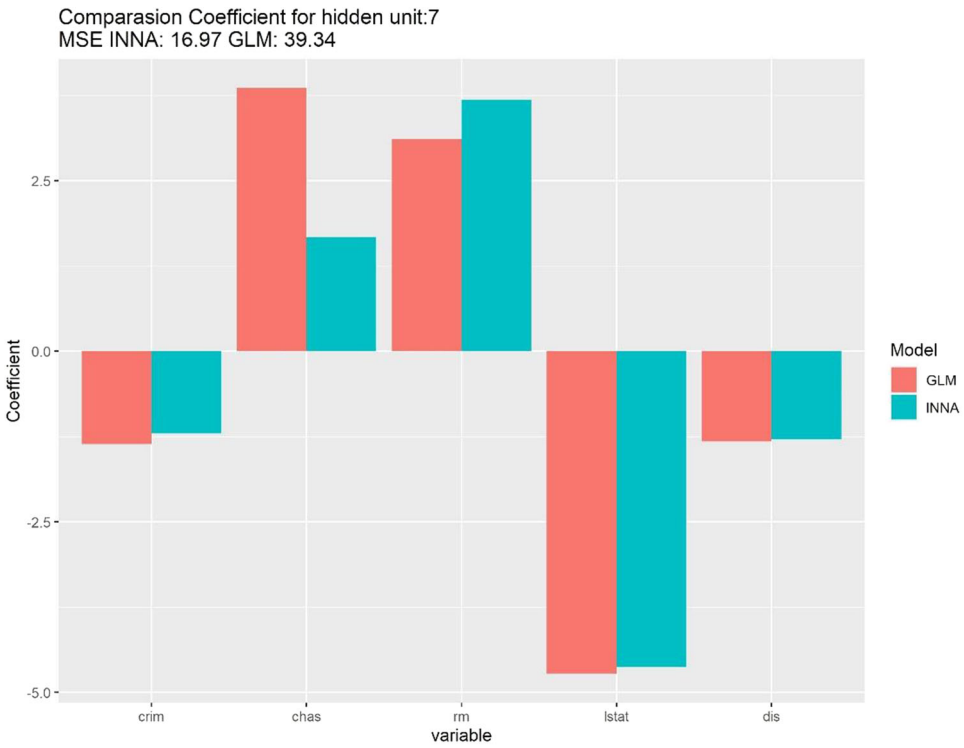


Fig. 13 Comparison of INNA and GLM coefficients for Boston's one hidden layers of seven nodes

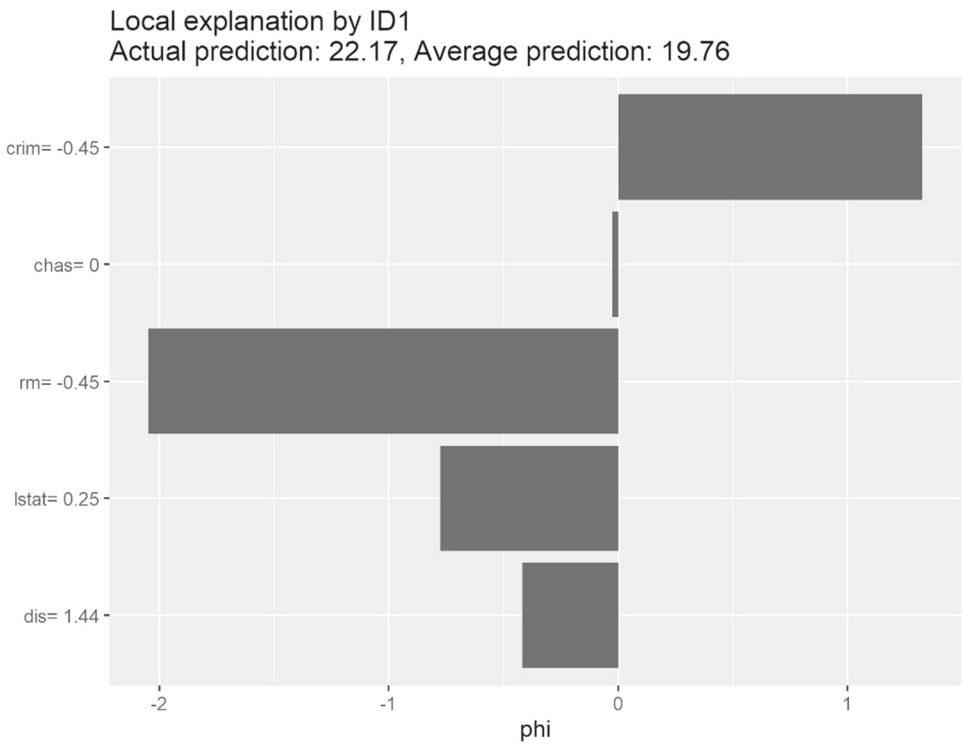


Fig. 14 SHAP effect of one hidden layer's INNA seven nodes from the Boston

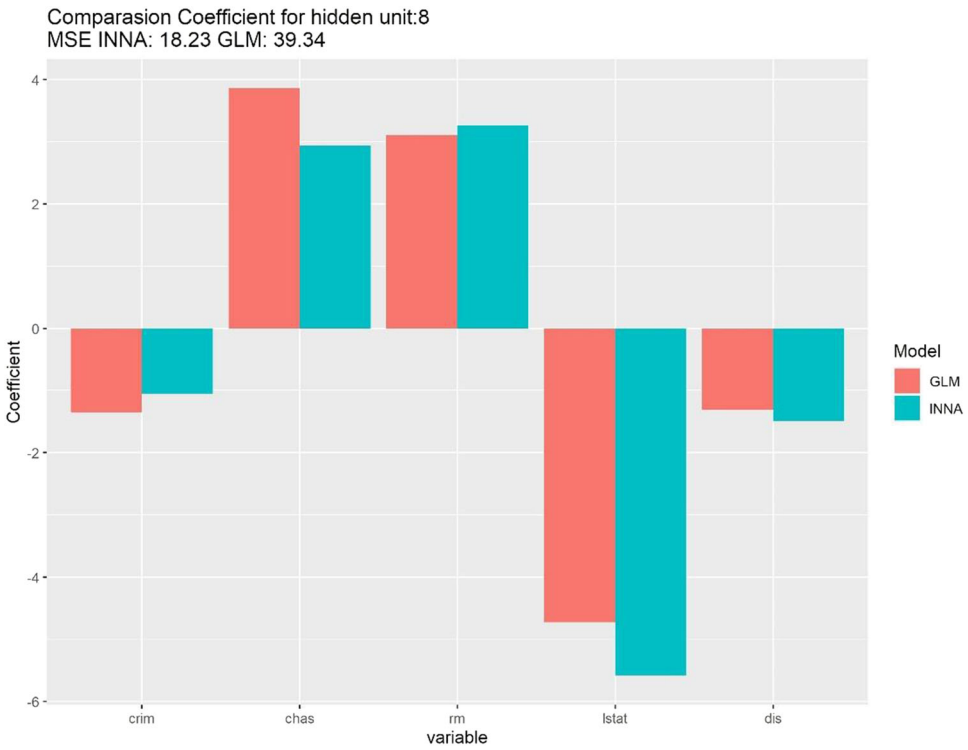


Fig. 15 Comparison of INNA and GLM coefficients for Boston's one hidden layers of eight nodes

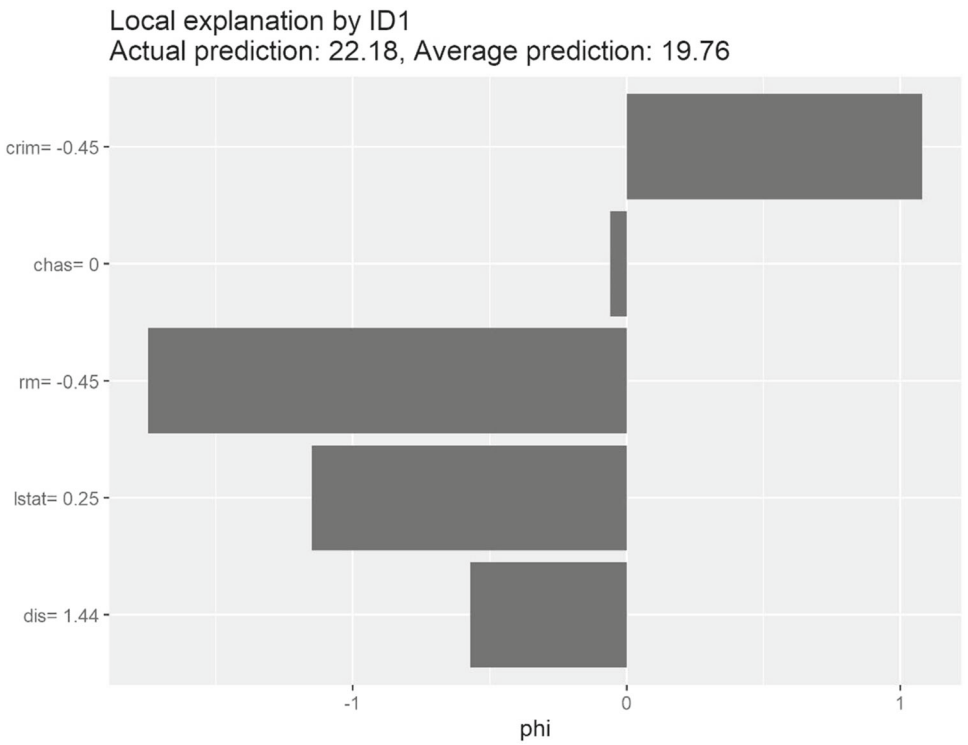


Fig. 16 SHAP effect of one hidden layer's INNA eight nodes from the Boston

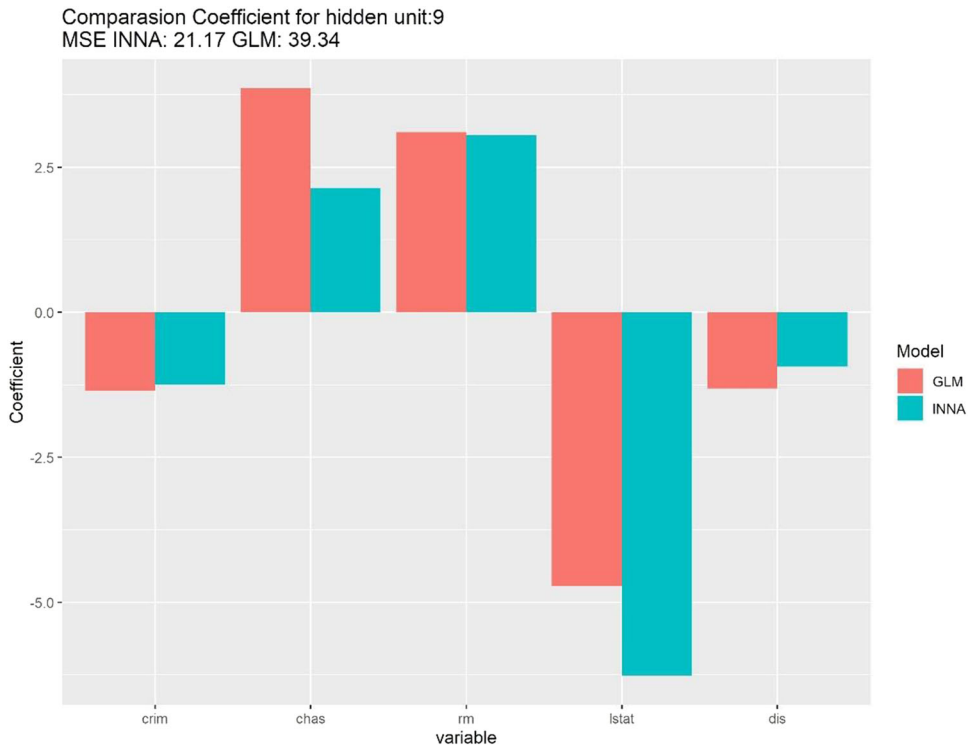


Fig. 17 Comparison of INNA and GLM coefficients for Boston's one hidden layers of nine nodes

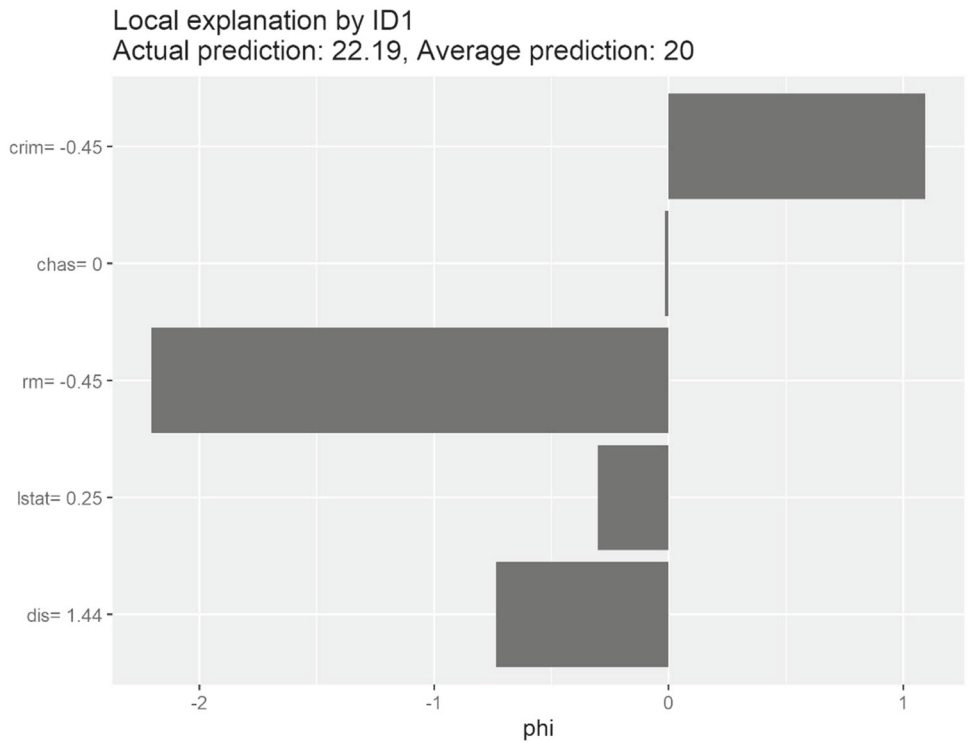


Fig. 18 SHAP effect of one hidden layer's INNA nine nodes from the Boston

Appendix 3

See Figs. 19, 20, 21, 22, 23, 24, 25, 26, 27, 28.

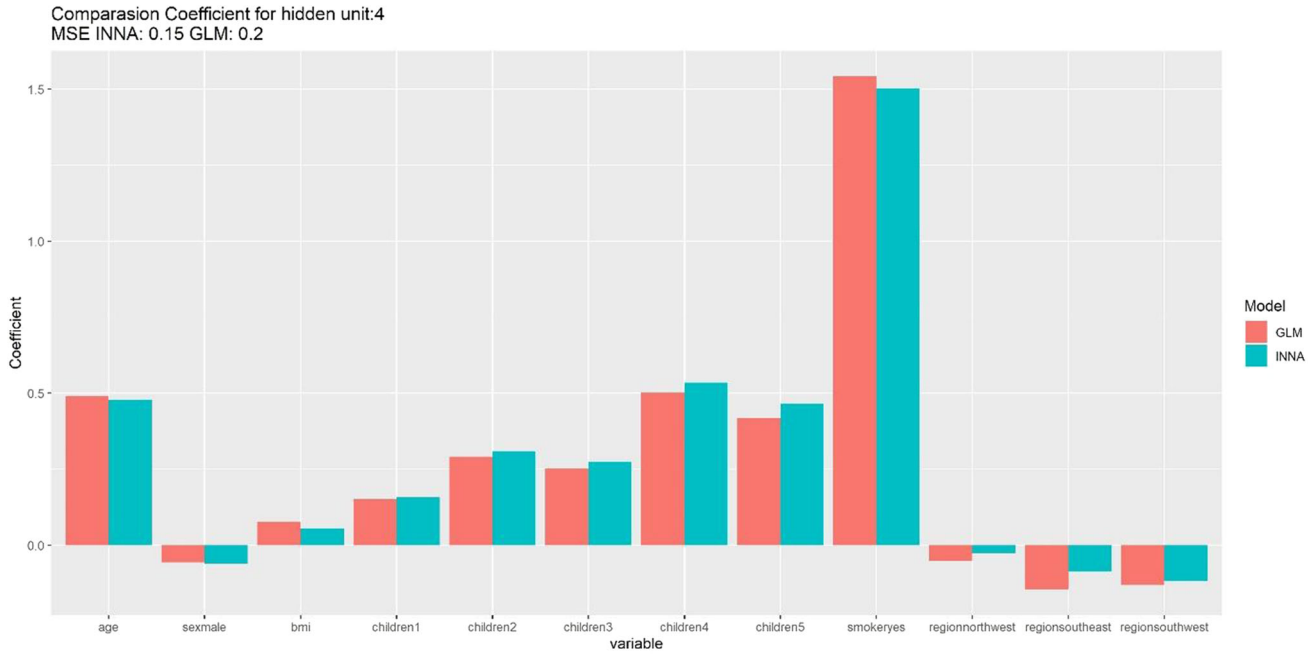


Fig. 19 Comparison of INNA and GLM coefficients for ACME's one hidden layers of four nodes

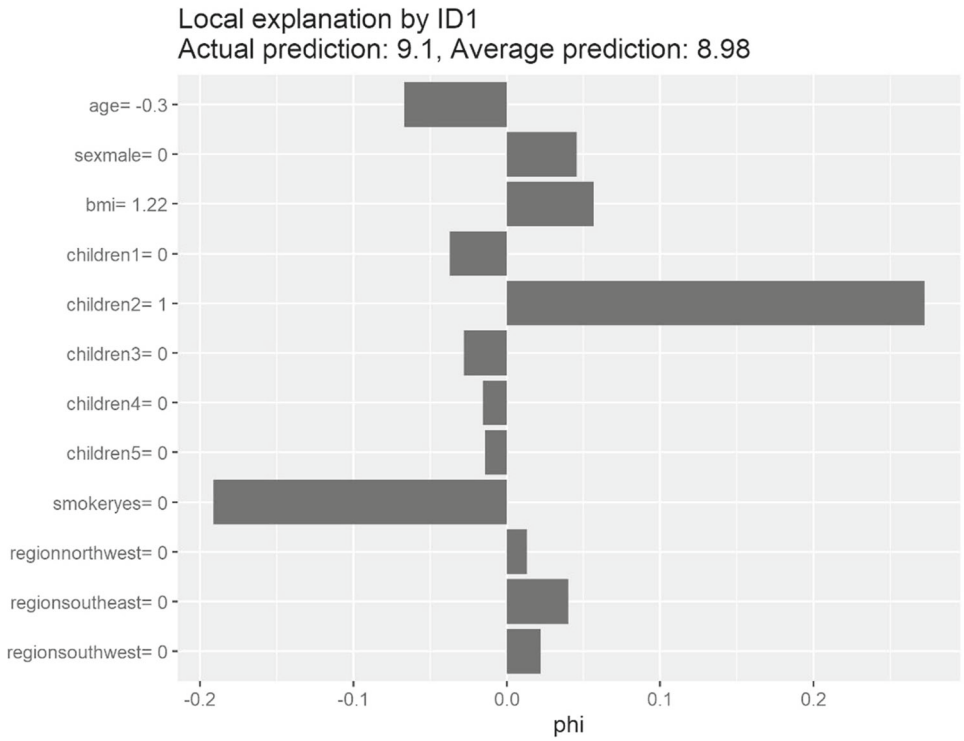


Fig. 20 SHAP effect of one hidden layer's INNA four nodes from the ACME

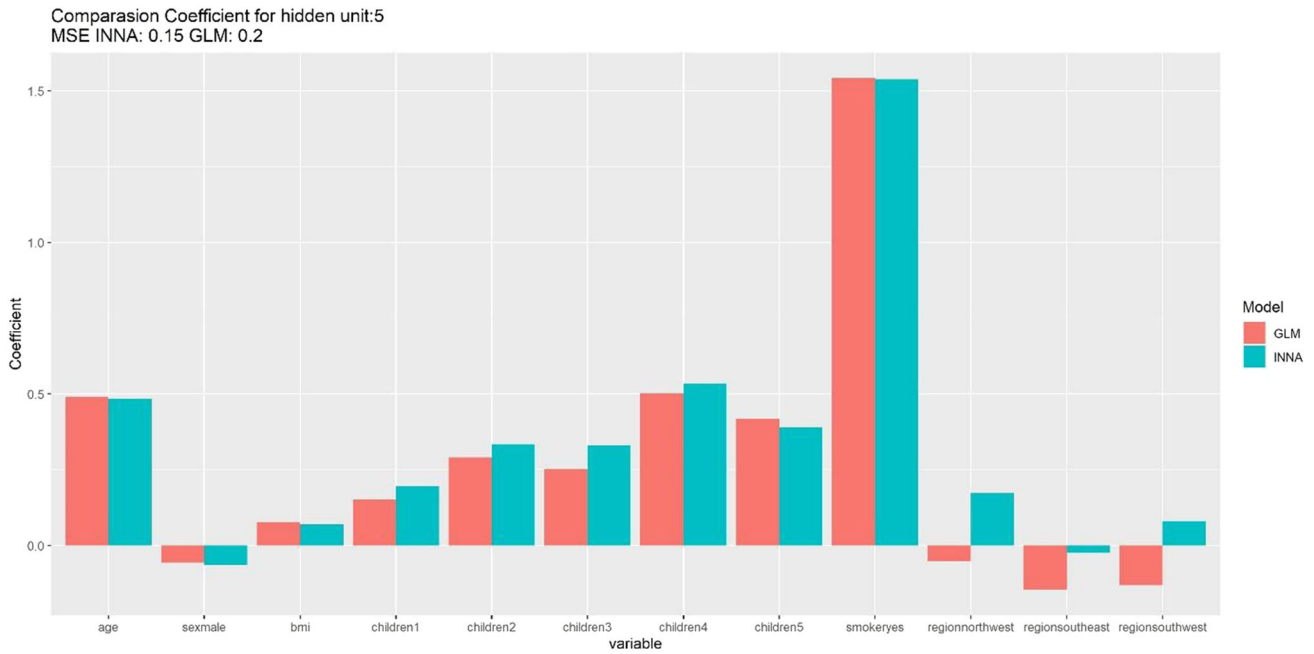


Fig. 21 Comparison of INNA and GLM coefficients for ACME's one hidden layers of five nodes

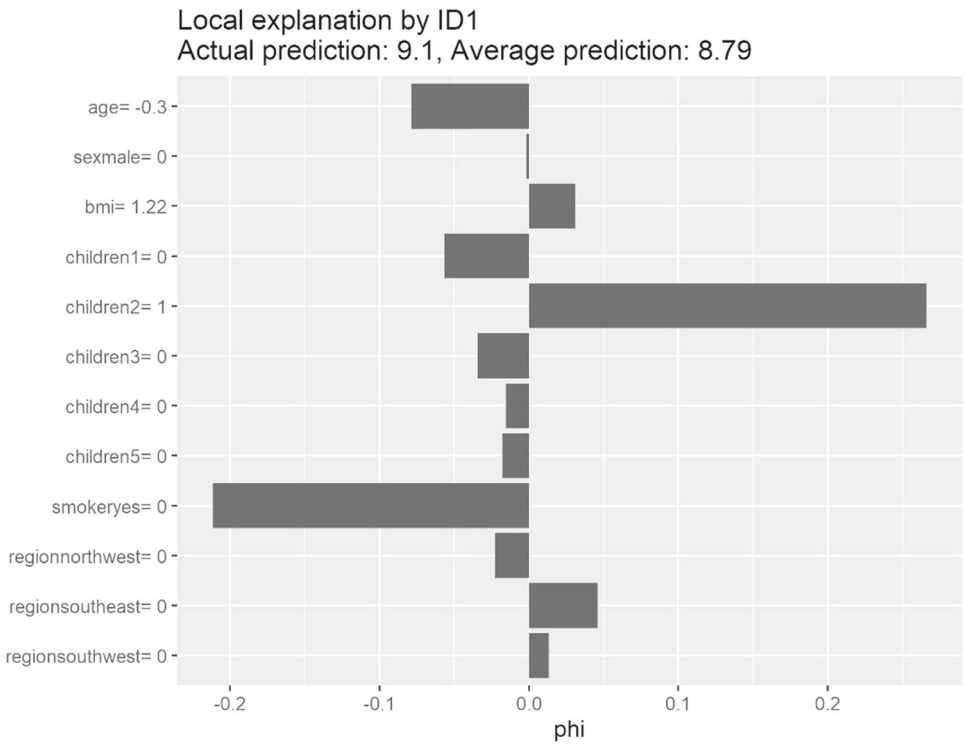


Fig. 22 SHAP effect of one hidden layer's INNA five nodes from the ACME

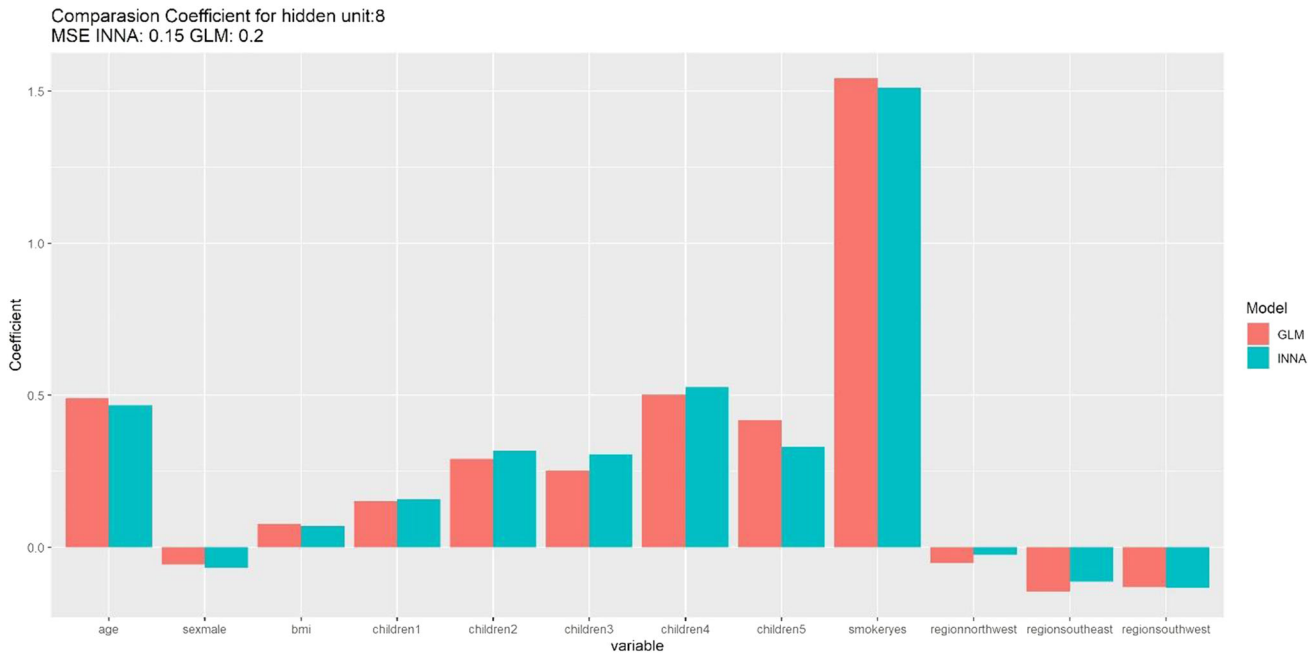


Fig. 23 Comparison of INNA and GLM coefficients for ACME's one hidden layers of eight nodes

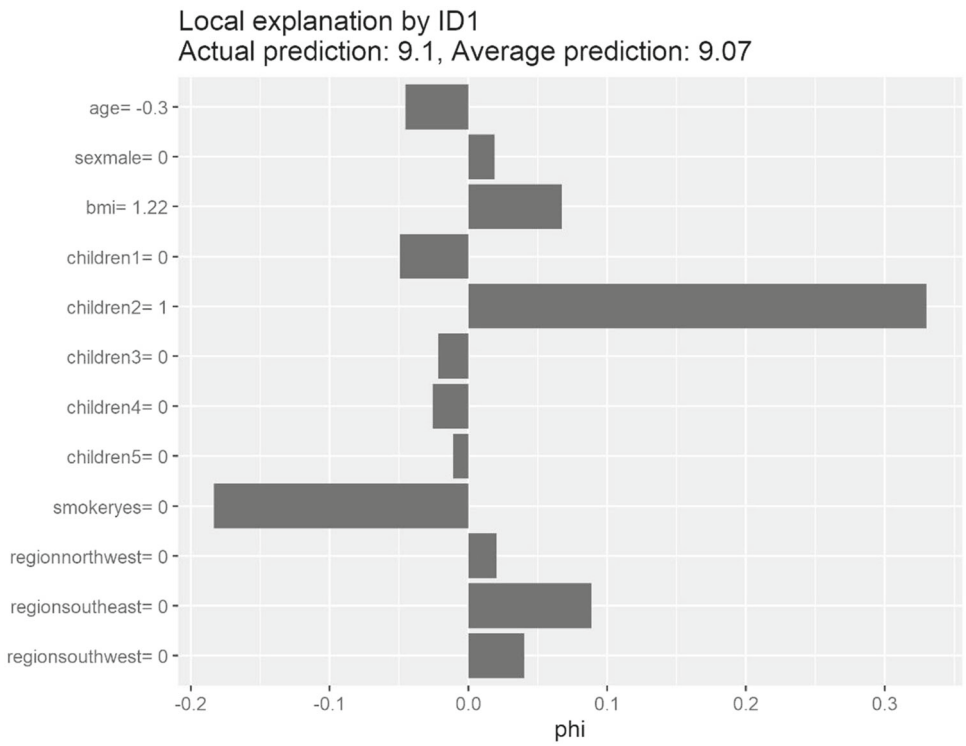


Fig. 24 SHAP effect of one hidden layer's INNA eight nodes from the ACME

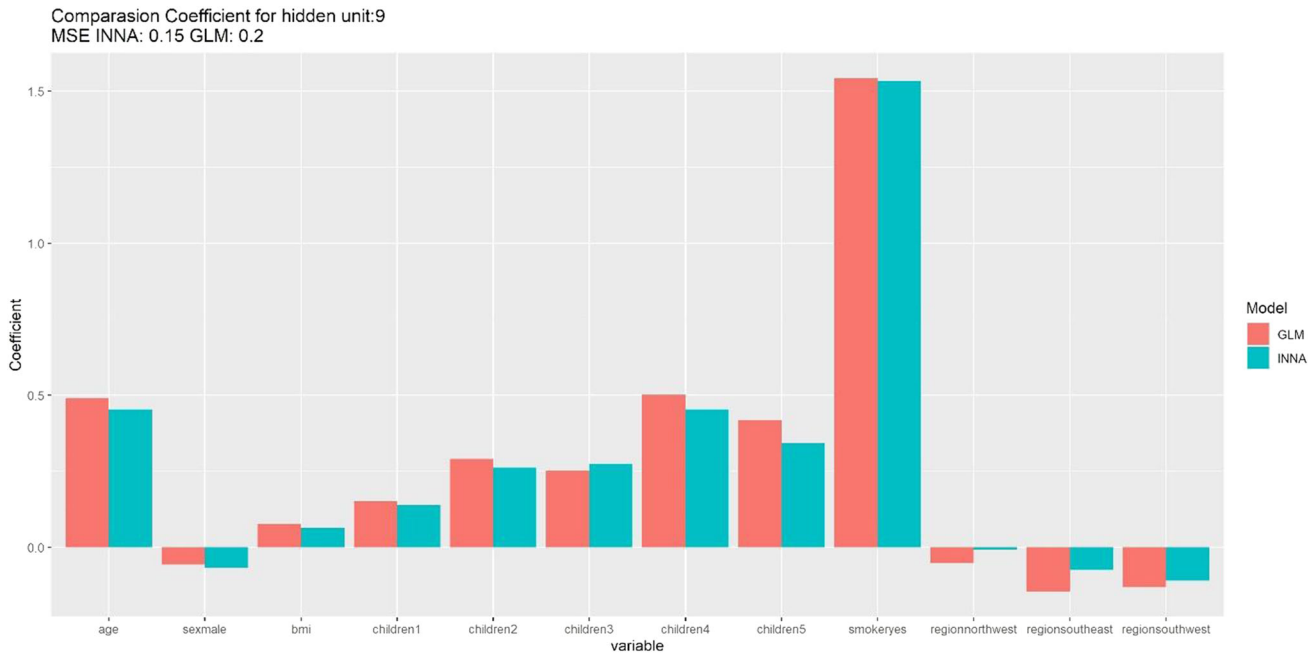


Fig. 25 Comparison of INNA and GLM coefficients for ACME's one hidden layers of nine nodes

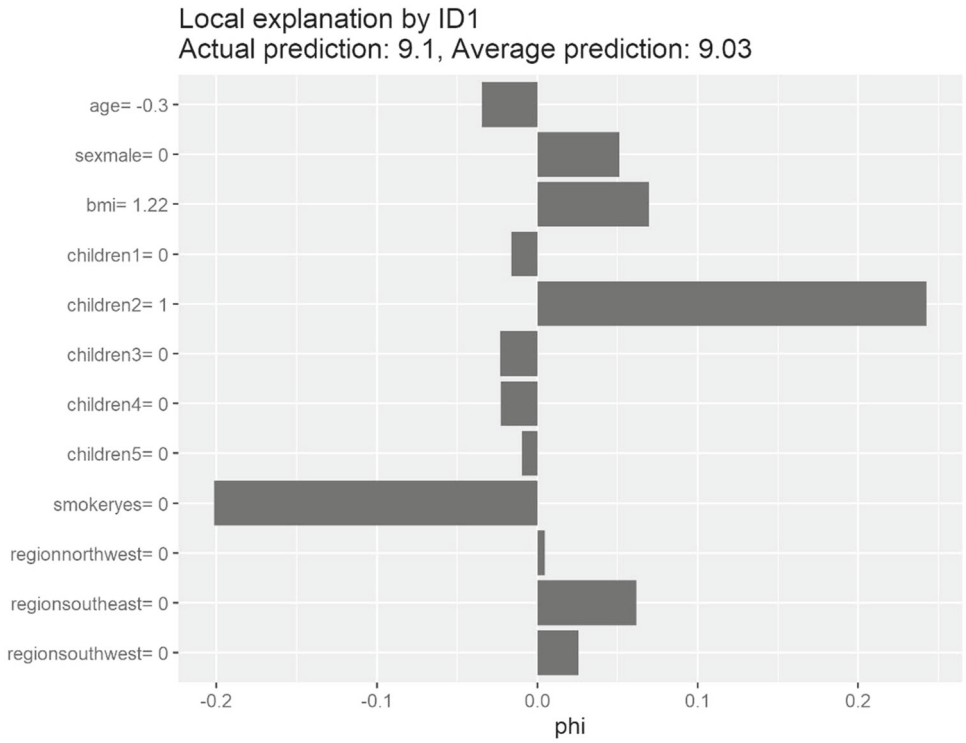


Fig. 26 SHAP effect of one hidden layer's INNA nine nodes from the ACME

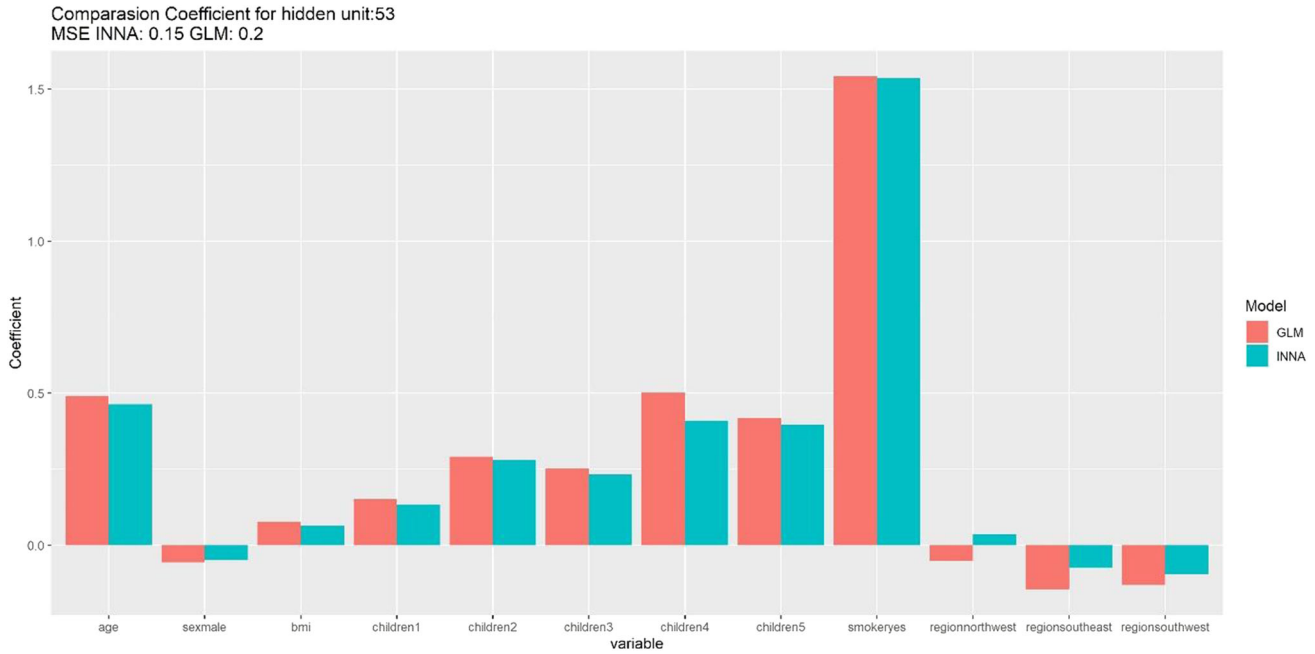


Fig. 27 Comparison of INNA and GLM coefficients for ACME's two hidden layers of five and three nodes

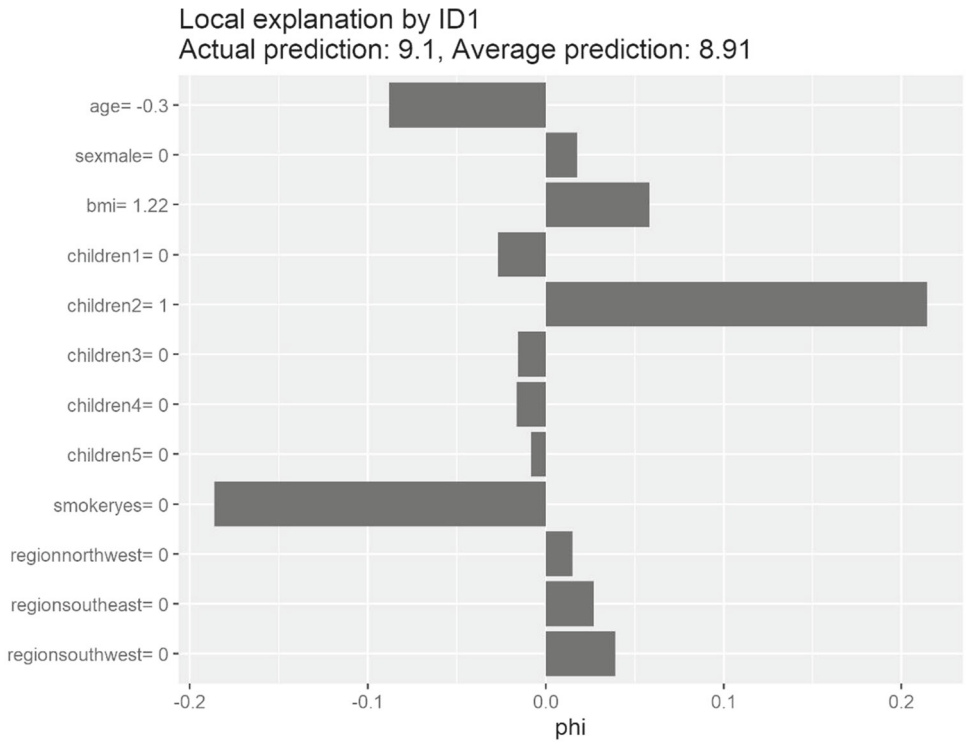


Fig. 28 SHAP effect of two hidden layer's INNA five and three nodes from the ACME

Appendix 4

See Figs. 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52.

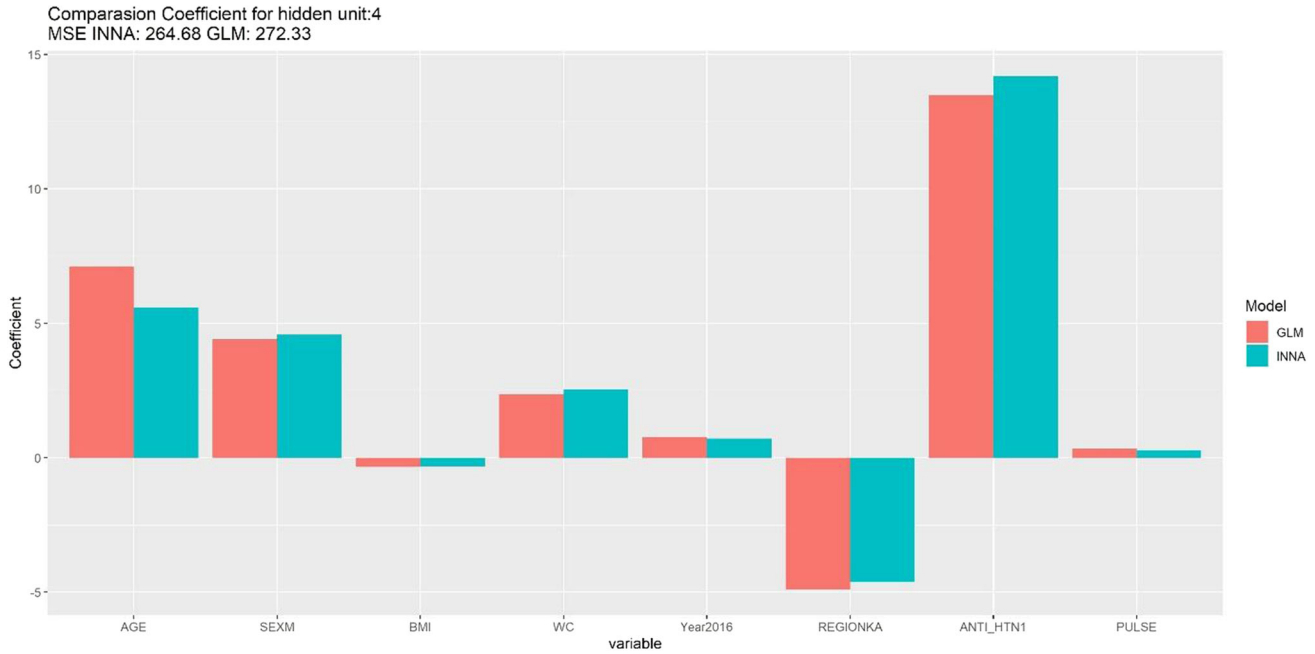


Fig. 29 SBP Comparison of four nodes from the Bukavu one hidden layer's INNA and GLM coefficients

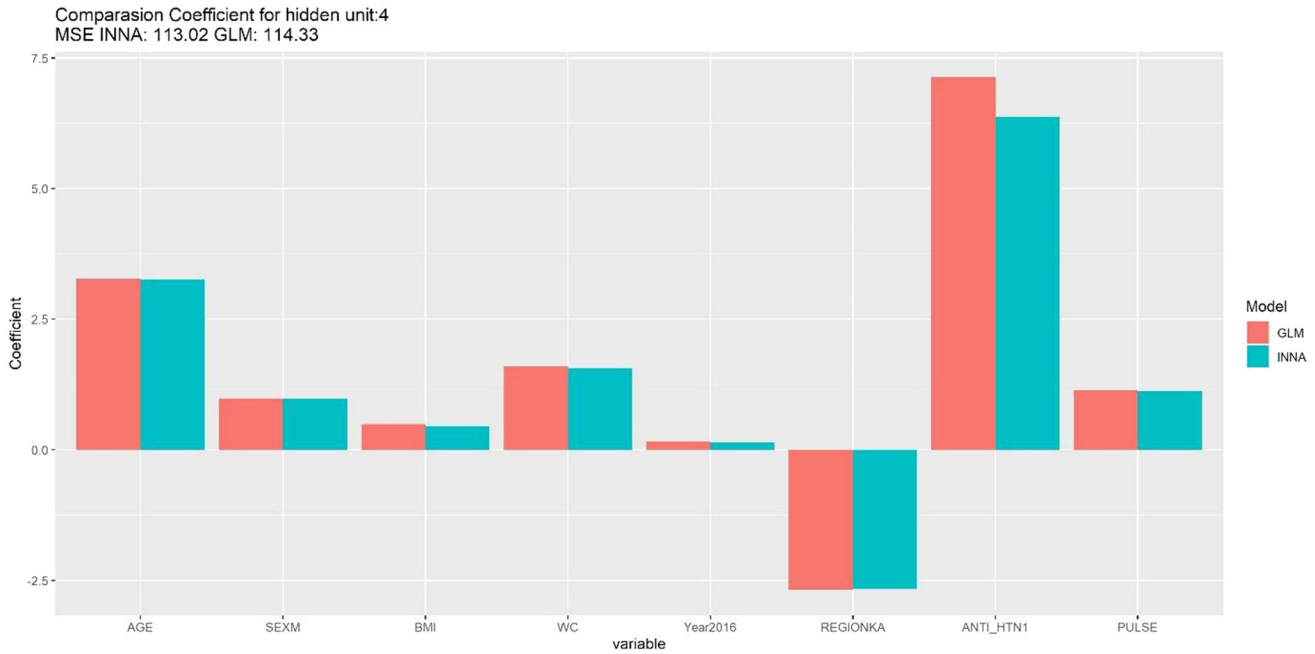


Fig. 30 DBP Comparison of four nodes from the Bukavu one hidden layer's INNA and GLM coefficients

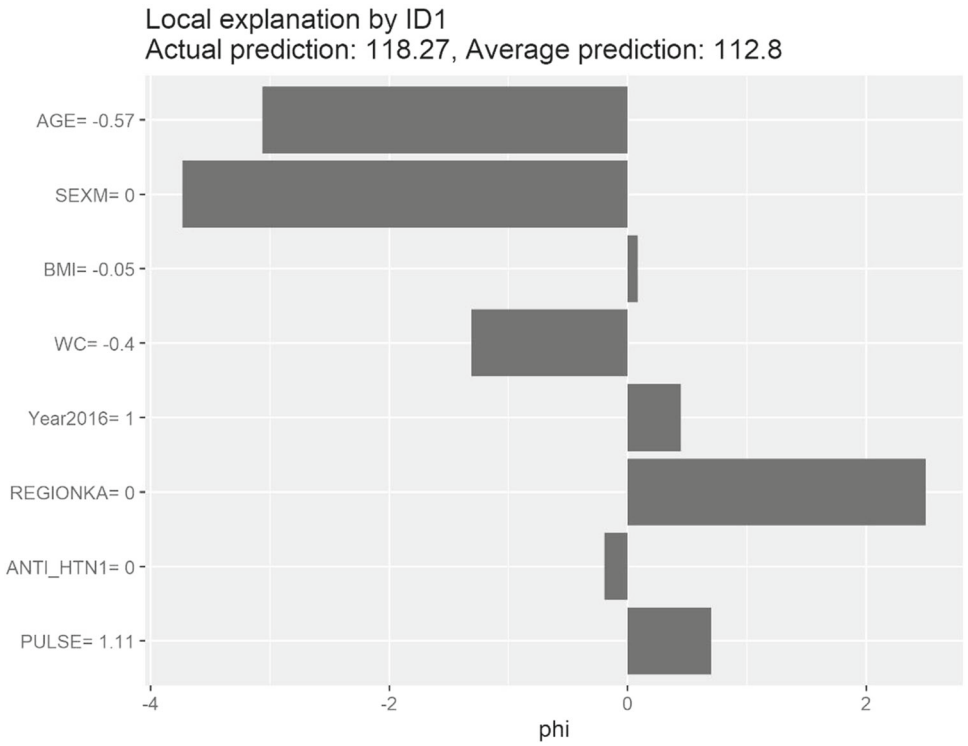


Fig. 31 SHAP effect of one hidden layer's INNA four nodes from the Bukavu SBP

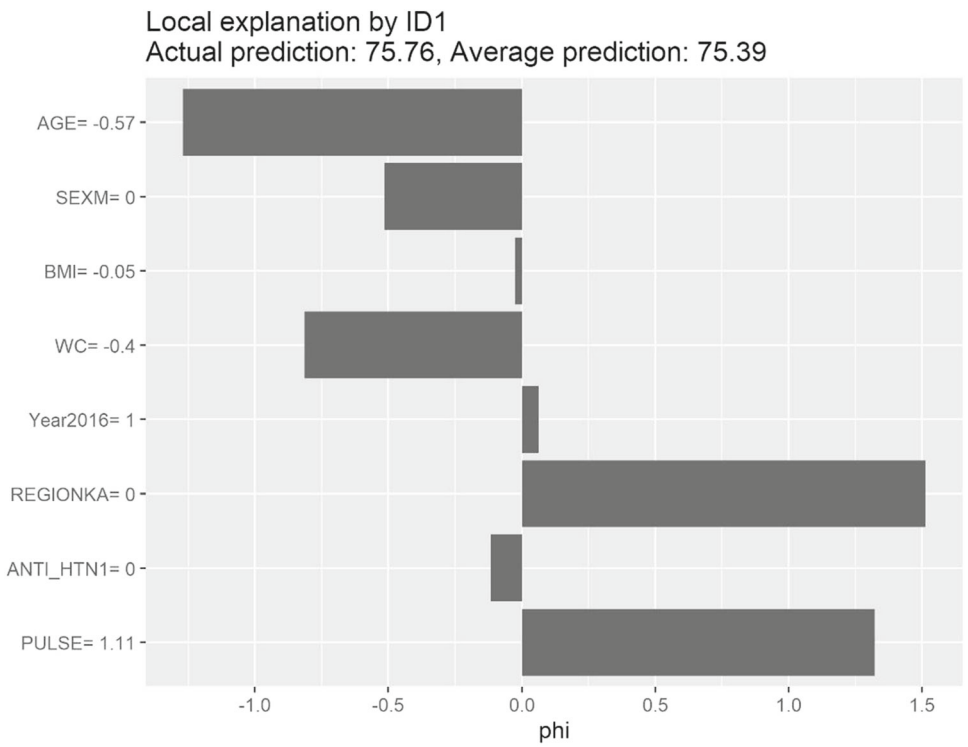


Fig. 32 SHAP effect of one hidden layer's INNA four nodes from the Bukavu DBP

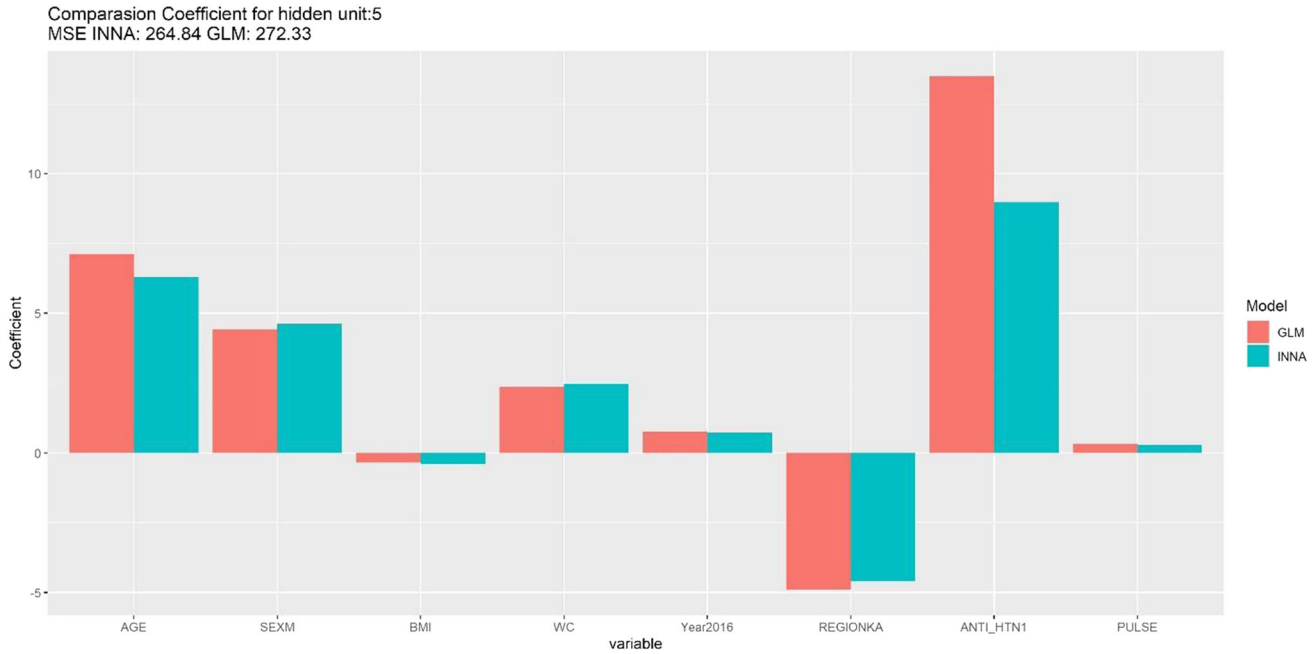


Fig. 33 SBP Comparison of five nodes from the Bukavu one hidden layer's INNA and GLM coefficients

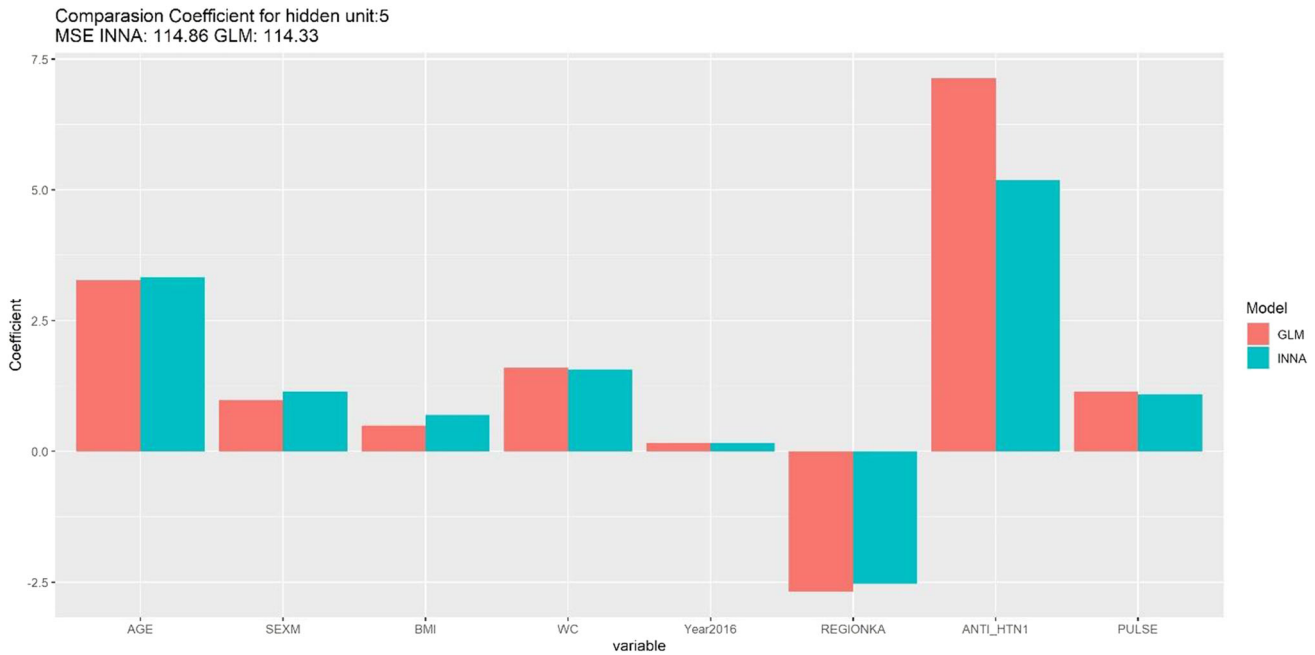


Fig. 34 DBP Comparison of five nodes from the Bukavu one hidden layer's INNA and GLM coefficients

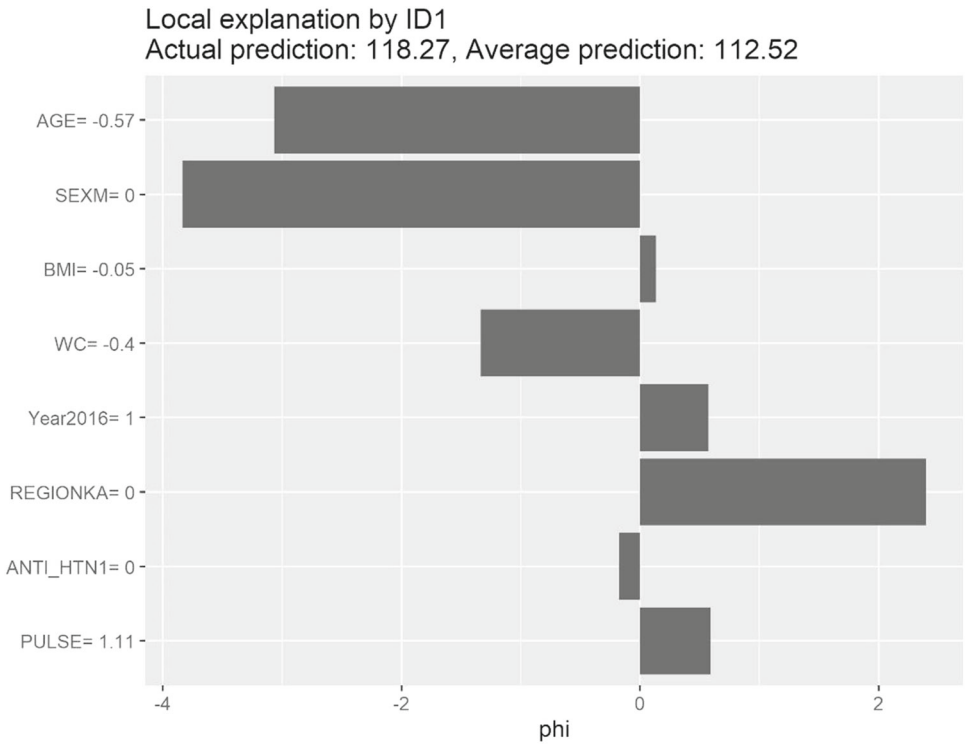


Fig. 35 SHAP effect of one hidden layer's INNA five nodes from the Bukavu SBP

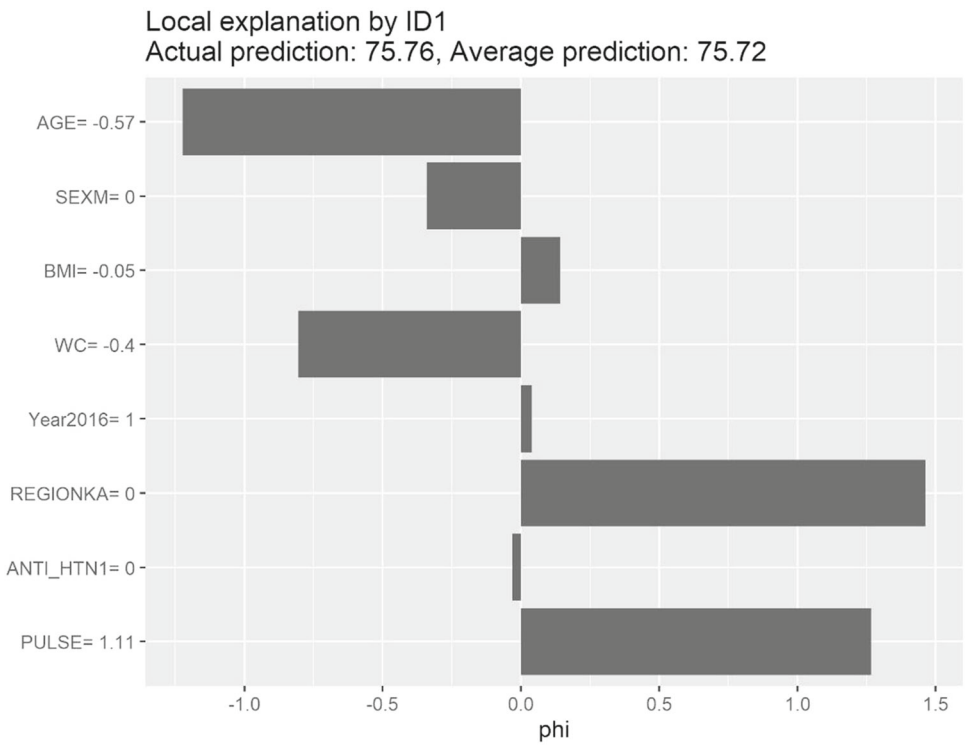


Fig. 36 SHAP effect of one hidden layer's INNA five nodes from the Bukavu DBP

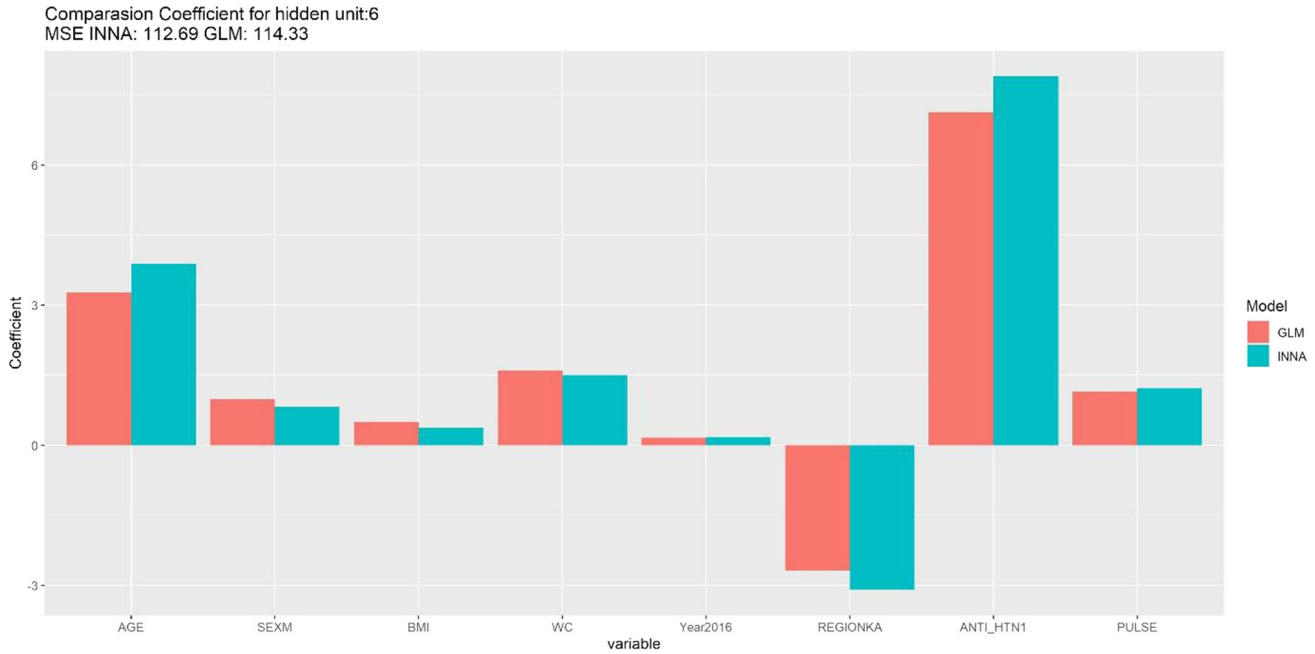


Fig. 37 DBP Comparison of six nodes from the Bukavu one hidden layer's INNA and GLM coefficients

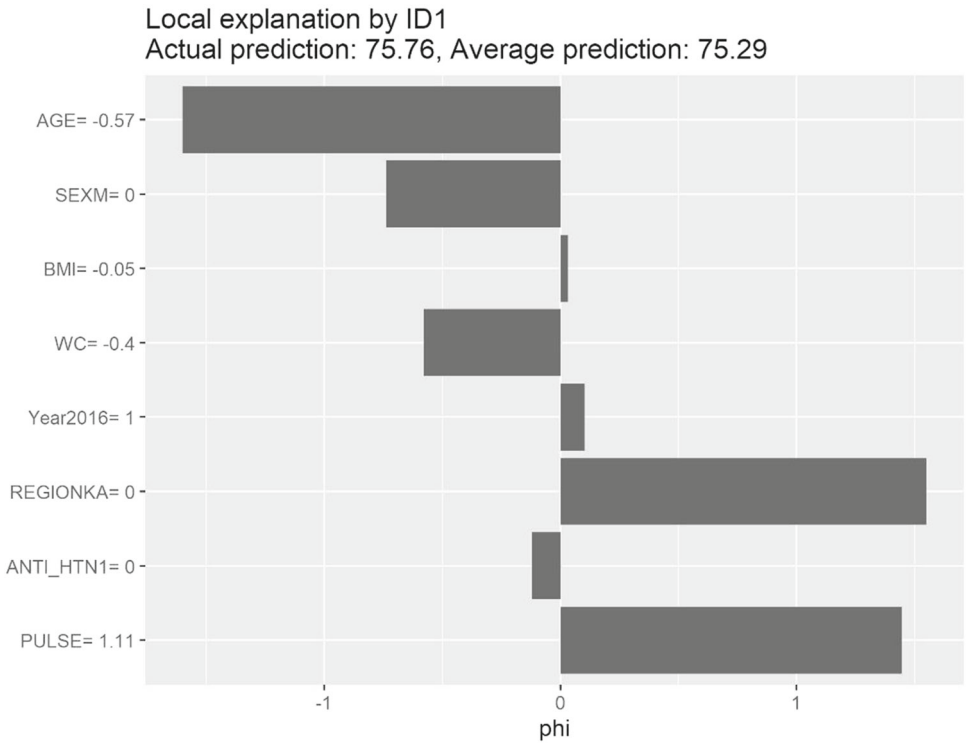


Fig. 38 SHAP effect of one hidden layer's INNA six nodes from the Bukavu DBP

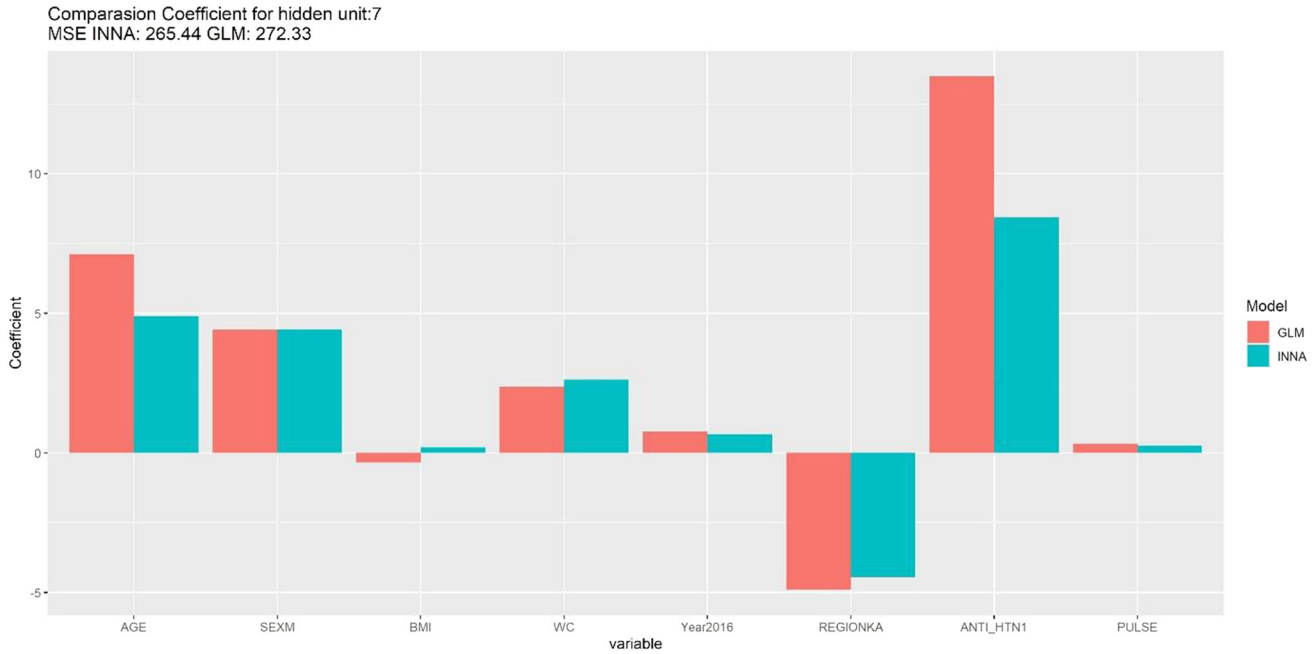


Fig. 39 SBP Comparison of seven nodes from the Bukavu one hidden layer's INNA and GLM coefficients

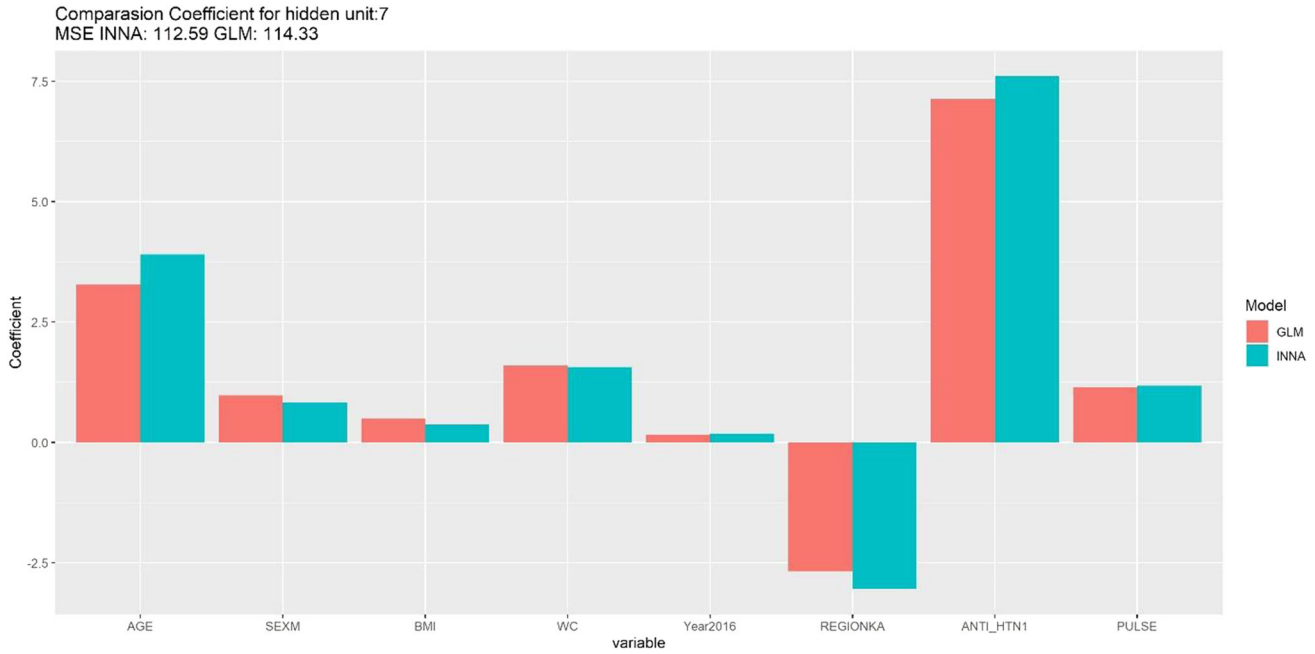


Fig. 40 DBP Comparison of seven nodes from the Bukavu one hidden layer's INNA and GLM coefficients

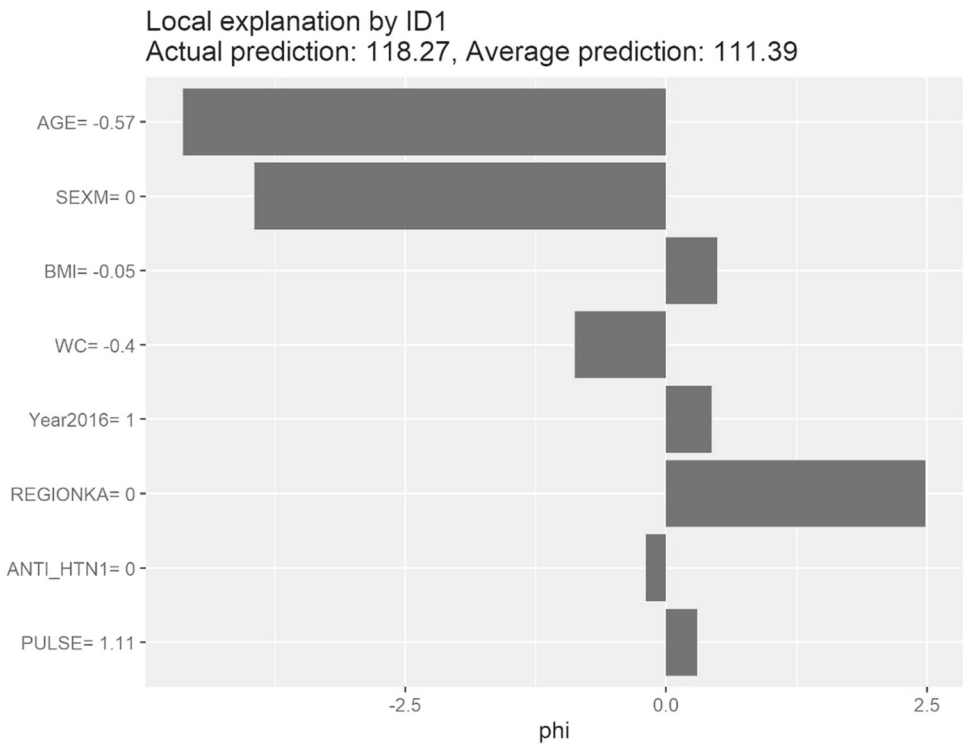


Fig. 41 SHAP effect of one hidden layer's INNA seven nodes from the Bukavu SBP

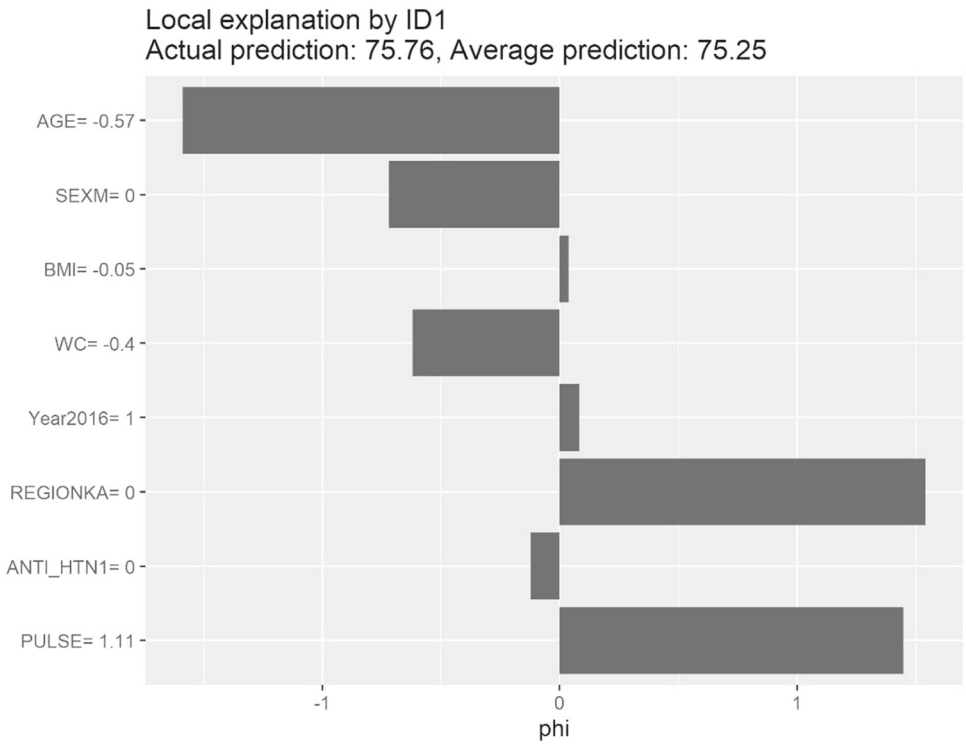


Fig. 42 SHAP effect of one hidden layer's INNA seven nodes from the Bukavu DBP

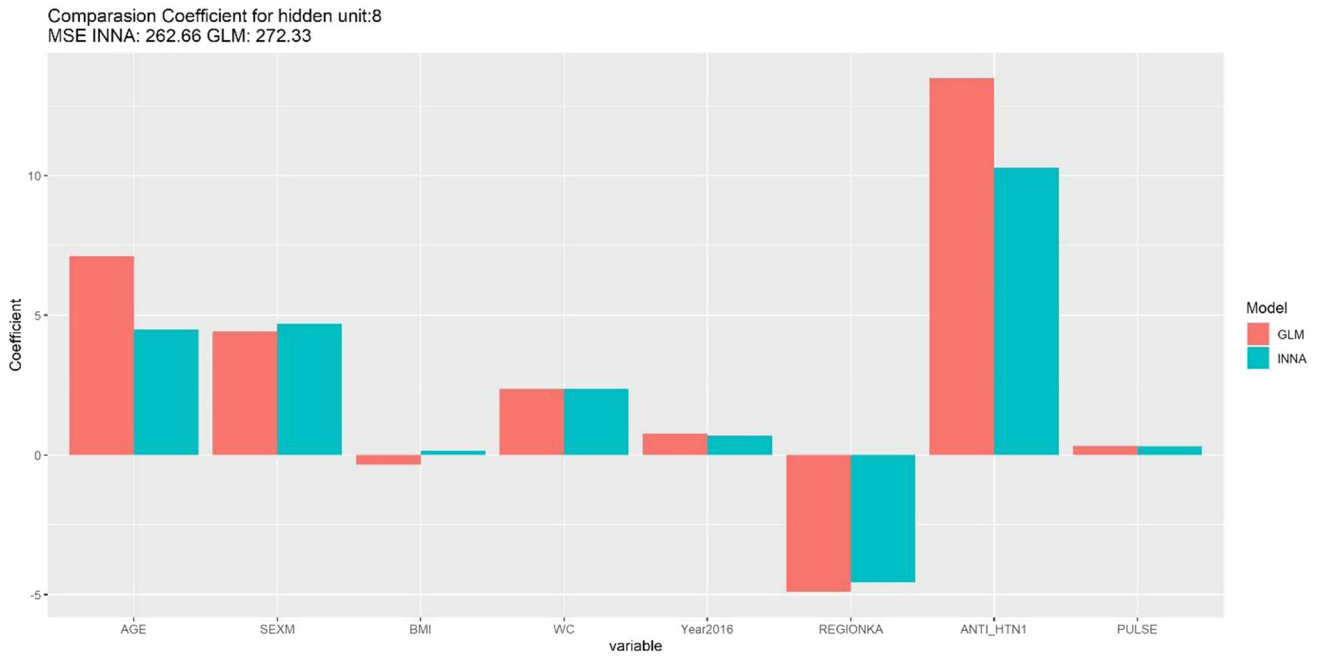


Fig. 43 SBP Comparison of eight nodes from the Bukavu one hidden layer's INNA and GLM coefficients

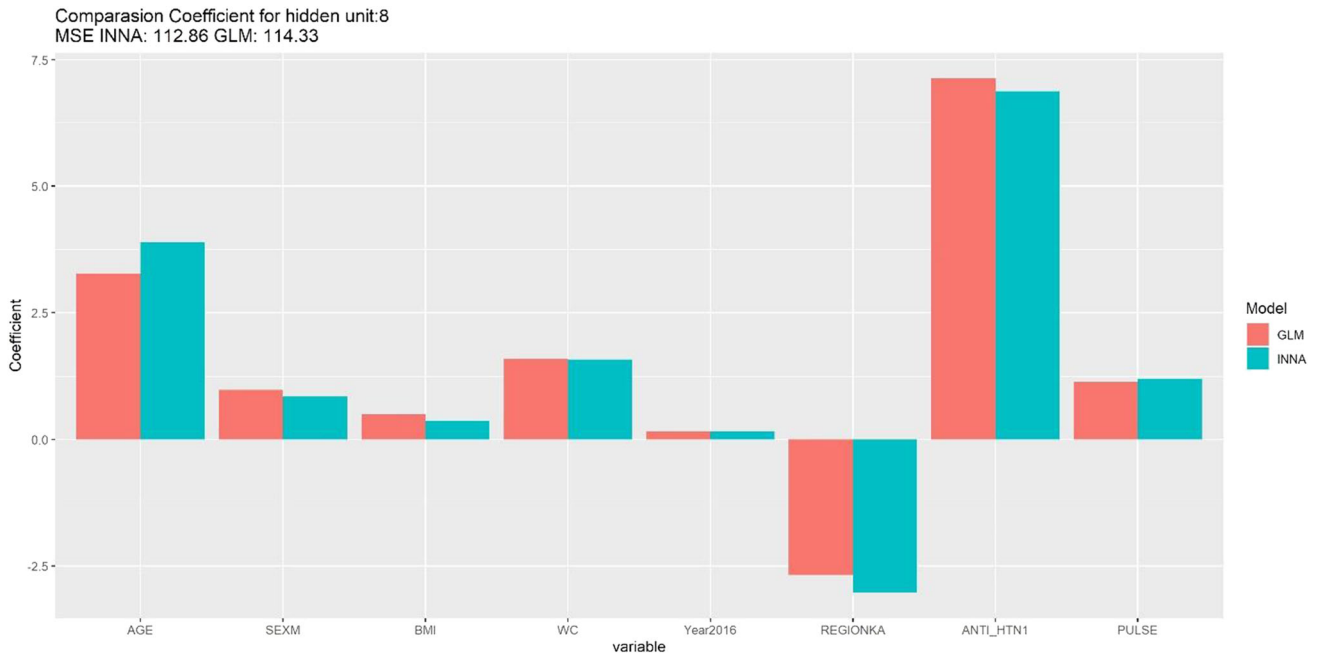


Fig. 44 DBP Comparison of eight nodes from the Bukavu one hidden layer's INNA and GLM coefficients

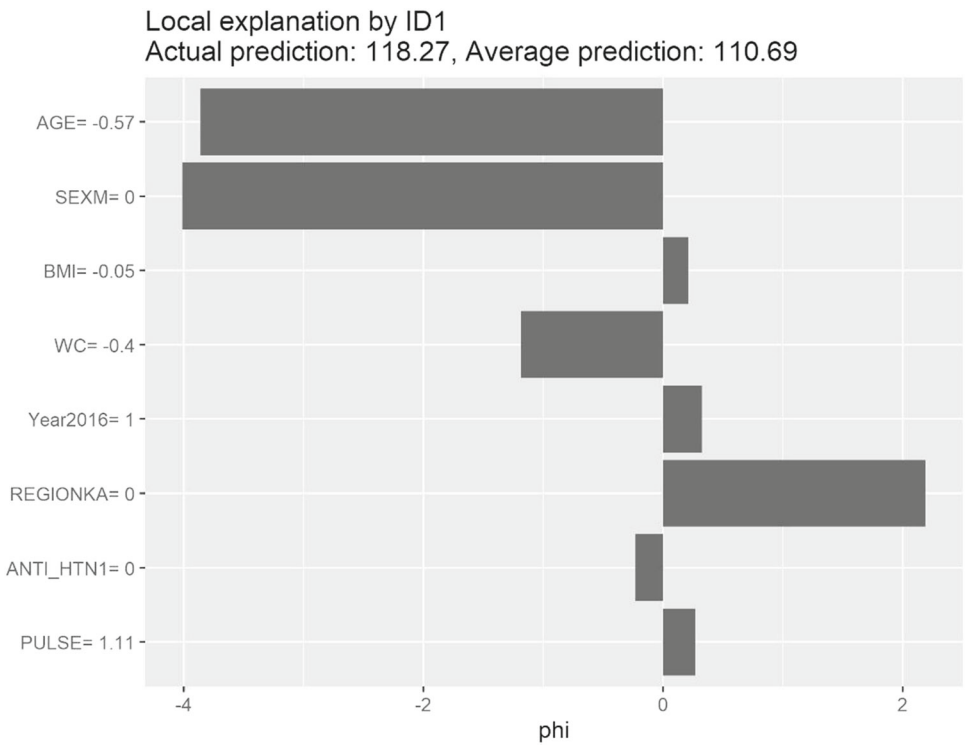


Fig. 45 SHAP effect of one hidden layer's INNA eight nodes from the Bukavu SBP

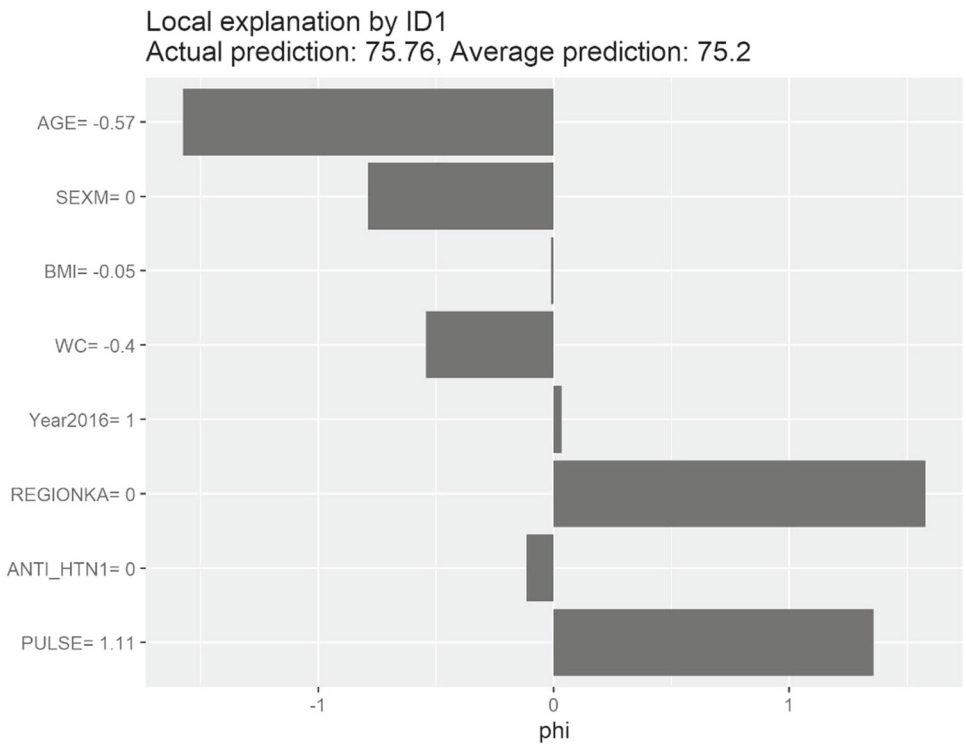


Fig. 46 SHAP effect of one hidden layer's INNA eight nodes from the Bukavu DBP

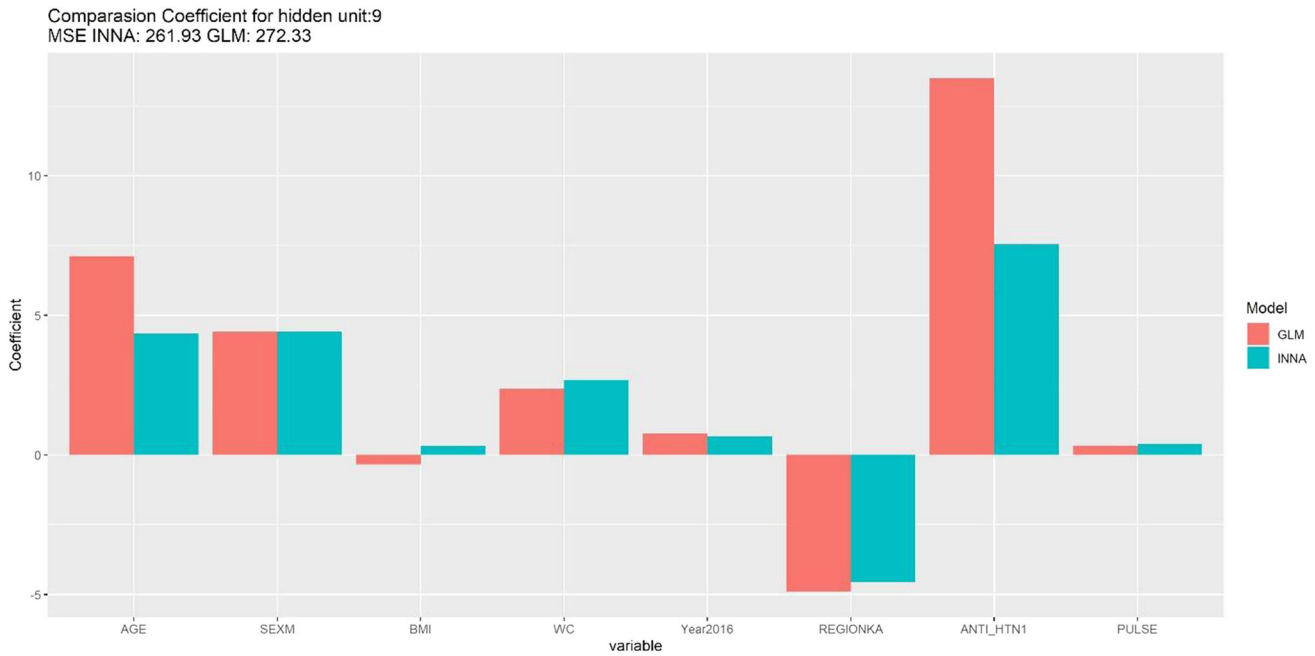


Fig. 47 SBP Comparison of nine nodes from the Bukavu one hidden layer's INNA and GLM coefficients

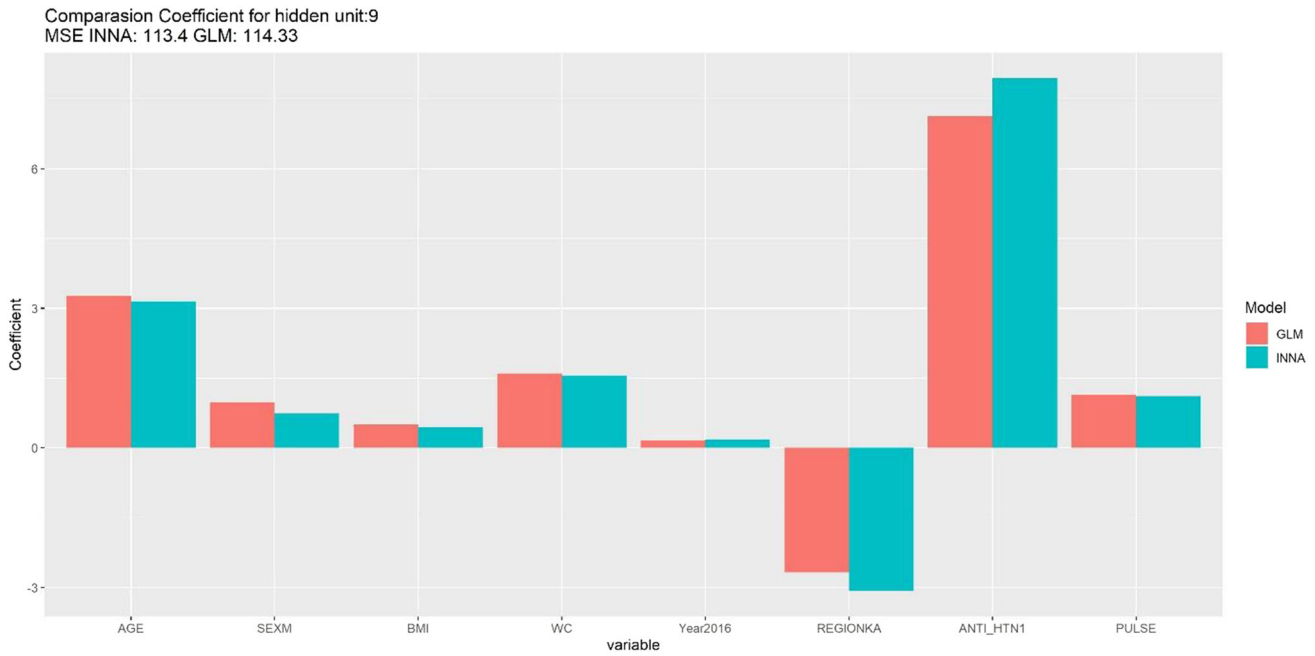


Fig. 48 DBP Comparison of nine nodes from the Bukavu one hidden layer's INNA and GLM coefficients

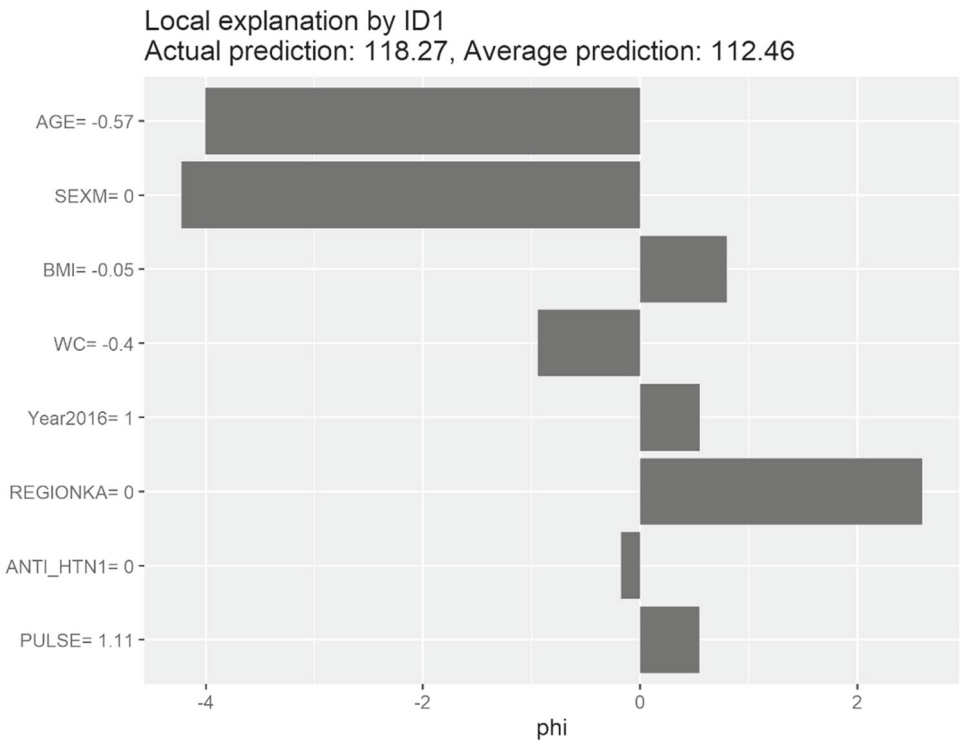


Fig. 49 SHAP effect of one hidden layer's INNA nine nodes from the Bukavu SBP

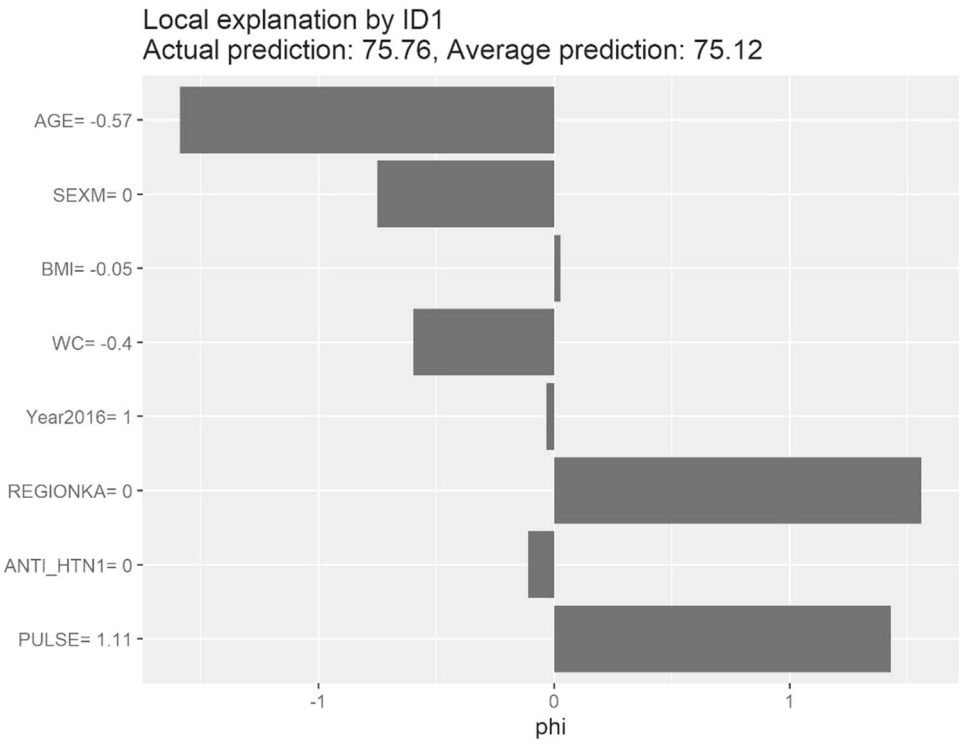


Fig. 50 SHAP effect of one hidden layer's INNA nine nodes from the Bukavu DBP

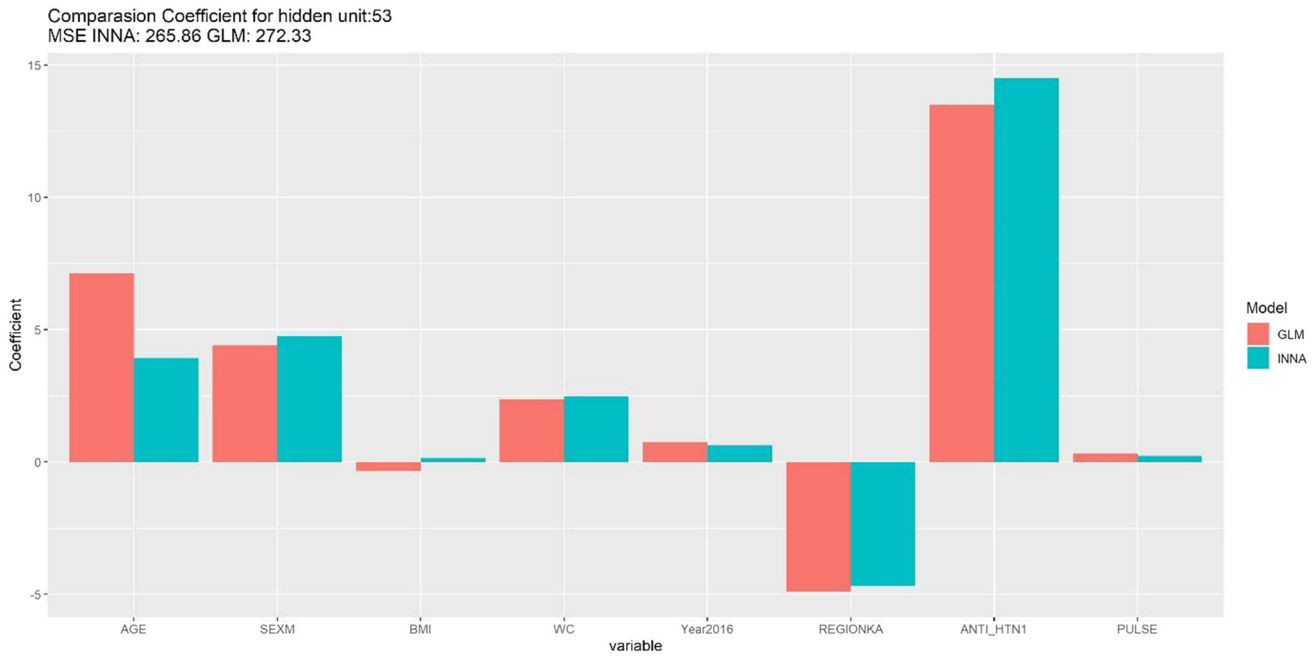


Fig. 51 SBP Comparison of five and three nodes from the Bukavu two hidden layer's INNA and GLM coefficients

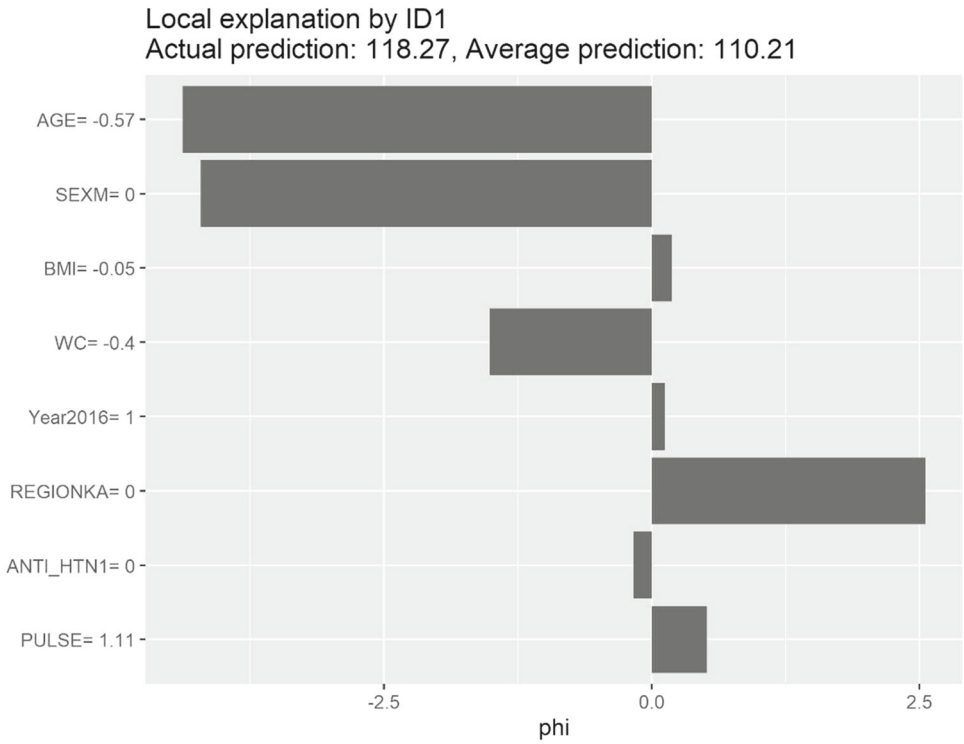


Fig. 52 SHAP effect of two hidden layer's INNA five and three nodes from the Bukavu SBP

References

1. Jordan, M.I., Mitchell, T.M.: Machine learning: trends, perspectives, and prospects. Science **349**(6245), 255–260 (2015)
2. Raita, Y., et al.: Emergency department triage prediction of clinical outcomes using machine learning models. Crit. Care **23**(1), 1–13 (2019)
3. Giuste, F.O., et al.: Early and fair COVID-19 outcome risk assessment using robust feature selection. Sci. Rep. **13**(1), 18981 (2023)
4. Yarkoni, T., Westfall, J.: Choosing prediction over explanation in psychology: lessons from machine learning. Perspect. Psychol. Sci. **12**(6), 1100–1122 (2017)
5. Guo, C.Y., Chou, Y.C.: A novel machine learning strategy for model selections—stepwise support vector machine (StepSVM). PLoS ONE **15**(8), e0238384 (2020). https://doi.org/10.1371/journal.pone.0238384.eCollection

6. Guo, C.Y., Yang, Y.C., Chen, Y.H.: The optimal machine learning-based missing data imputation for the cox proportional hazard model. Front. Public Health **9**, 680054 (2021)
7. Guo, C.Y., Lin, Y.J.: Random interaction forest (RIF)–a novel machine learning strategy accounting for feature interaction. IEEE Access **11**, 1806–1813 (2023)
8. Wang, Y., et al.: Generalized estimating equations boosting (GEEB) machine for correlated data. J. Big Data **11**, 20 (2024)
9. Gaonkar, B., et al.: Interpreting support vector machine models for multivariate group wise analysis in neuroimaging. Med. Image Anal. **24**(1), 190–204 (2015)
10. Zhao, X., et al.: iforest: Interpreting random forests via visual analytics. IEEE Trans. Vis. Comput. Graph. **25**(1), 407–416 (2018)
11. Belle, V., Papantonis, I.: Principles and practice of explainable machine learning. Front. Big Data 39 (2021)
12. Alwosheel, A., van Cranenburgh, S., Chorus, C.G.: Why did you predict that? Toward explainable artificial neural networks for travel demand analysis. Transp. Res. Part C Emerg. Technol. **128**, 103143 (2021)
13. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**(5), 359–366 (1989)
14. White, H.: Learning in artificial neural networks: a statistical perspective. Neural Comput. **1**(4), 425–464 (1989)
15. Huang, J.-C., et al.: Application and comparison of several machine learning algorithms and their integration models in regression problems. Neural Comput. Appl. **32**(10), 5461–5469 (2020)
16. Wiens, J., Shenoy, E.S.: Machine learning for healthcare: on the verge of a major shift in healthcare epidemiology. Clin. Infect. Dis. **66**(1), 149–153 (2018)
17. Ding, N. et al.: An artificial neural networks model for early predicting in-hospital mortality in acute pancreatitis in MIMIC-III. BioMed Res. Int. **2021** (2021)
18. Alhazmi, A., et al.: Application of artificial intelligence and machine learning for prediction of oral cancer risk. J. Oral Pathol. Med. **50**(5), 444–450 (2021)
19. Musunuri, B., et al.: Acute-on-chronic liver failure mortality prediction using an artificial neural network. Eng. Sci. **15**, 187–196 (2021)
20. Cardozo, G. et al.: Use of Machine learning and routine laboratory tests for diabetes mellitus screening. BioMed Res. Int. **2022** (2022)
21. Anton, N., et al.: Use of artificial neural networks to predict the progression of glaucoma in patients with sleep apnea. Appl. Sci. **12**(12), 6061 (2022)
22. Shanbehzadeh, M., Nopour, R., Kazemi-Arpanahi, H.: Developing an intelligent system for diagnosis of COVID-19 based on artificial neural network. Acta Med. Iran. **60**(3), 135 (2022)
23. Schulz, H., Behnke, S.: Deep learning. KI-Künstliche Intelligenz **26**(4), 357–363 (2012)
24. Ribeiro, M.T., Singh, S., Guestrin, C.: "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 (2016)
25. Dargan, S., et al.: A survey of deep learning and its applications: a new paradigm to machine learning. Arch. Comput. Methods Eng. **27**(4), 1071–1092 (2020)
26. Lundberg, S.M., Lee, S.-I.: A unified approach to interpreting model predictions. Adv. Neural Inf. Process. Syst. **30** (2017).
27. Cheng, X. et al.: Polynomial regression as an alternative to neural nets. arXiv preprint arXiv:1806.06850 (2018)
28. Lemhadri, I., Ruan, F., Tibshirani, R.: Lassonet: neural networks with feature sparsity. In: International Conference on Artificial Intelligence and Statistics. PMLR (2021)
29. Agarwal, R., et al.: Neural additive models: interpretable machine learning with neural nets. Adv. Neural. Inf. Process. Syst. **34**, 4699–4711 (2021)
30. Morala, P., et al.: Toward a mathematical framework to inform neural network modeling via polynomial regression. Neural Netw. **142**, 57–72 (2021)
31. Giuste, F., et al.: Explainable artificial intelligence methods in combating pandemics: a systematic review. IEEE Rev. Biomed. Eng. **16**, 5–21 (2023)
32. Rahmatinejad, Z., et al.: A comparative study of explainable ensemble learning and logistic regression for predicting in-hospital mortality in the emergency department. Sci. Rep. **14**(1), 3406 (2024)
33. Son, B., et al.: Improved patient mortality predictions in emergency departments with deep learning data-synthesis and ensemble models. Sci. Rep. **13**(1), 15031 (2023)
34. Meltzer, M.E., et al.: Venous thrombosis risk associated with plasma hypofibrinolysis is explained by elevated plasma levels of TAFI and PAI-1. Blood J. Am. Soc. Hematol. **116**(1), 113–121 (2010)
35. Rauber, F., et al.: Consumption of ultra-processed food products and its effects on children's lipid profiles: a longitudinal study. Nutr Metab Cardiovasc Dis **25**(1), 116–122 (2015)
36. Kirby, M., et al.: Total airway count on computed tomography and the risk of chronic obstructive pulmonary disease progression. Findings from a population-based study. Am. J. Respir. Crit. Med. **197**(1), 56–65 (2018)
37. McCullagh, P., Nelder, J.A.: Generalized Linear Models. Routledge (2019)
38. Harrison, D. Boston Housing (2019). Available from: https://www.kaggle.com/c/boston-housing.
39. Singh, H. Medical Insurance Payout (2021). Available from: https://www.kaggle.com/datasets/harshsingh2209/medical-insurance-payout.
40. Katchunga, P.B., et al.: The trend in blood pressure and hypertension prevalence in the general population of South Kivu between 2012 and 2016: results from two representative cross-sectional surveys—the Bukavu observational study. PLoS ONE **14**(8), e0219377 (2019)
41. R Core Team (2014). R: A language and environment for statistical computing. R Foundation for Statistical Computing, V., Austria. URL http://www.R-project.org/.