Noname manuscript No.
(will be inserted by the editor)

Interpreting Neural Networks Prediction for a Single Instance via Random Forest Feature Contributions

Anna Palczewska · Urszula Markowska-Kaczmar · Daniel Neagu

Received: date / Accepted: date

Abstract Artificial Neural Networks are well-known black-box classifiers that generally suffer from their non-transparent structures, making it hard for data scientists to understand the mechanisms behind a particular prediction. Such knowledge is necessary for many domains for trained models used for forecasting. Rule extraction or variable importance are two main approaches allowing Neural Network model interpretation, but both interpret on a general level. So far, there are no known methods for interpreting artificial Neural Networks model prediction on an instance level. This paper is an attempt to solve this problem. We propose a pedagogical approach (borrowed from rule extraction) to estimate the influence between instance variables and the predicted outcome. We used the Feature Contributions method calculated from the Random Forest model that was trained to mimic the Artificial Neural Networks classification as close as possible. Feature contributions are numerical values that are further interpreted by domain experts to reveal some phenomena about a particular instance or model behaviour. We assume that we can trust the Feature Contributions results when both predictions are the same, i.e. Neural Network and Feature Contributions give the same results. The experimental results show that this highly depends on the level the Neural Network is trained because the error is then propagated to the Random Forest model. For good trained ANNs we can trust interpretation based on Feature Contributions on average in 80%.

Keywords Model interpretation · Artificial Neural Network · Feature Contributions

1 Introduction

Artificial Neural Networks, shortly ANNs, are widely accepted as machine learning tools to learn complex relationships from data. Currently, they see the resurrection and general adoption for classification and prediction problems in many data-rich areas. Their pattern-matching and learning capabilities allowed them to address many problems that were difficult or impossible to solve by other computational methods. Unfortunately, they lack transparency. The ANNs structure makes it impossible to predict or know the exact flow of data. It is hard to see how the network arrives at a particular conclusion due to the complexity of the network architecture. This is why ANN is often called a black-box model [21].

Interpretation of the "black box" models has become a crucial element for many modellers. Scientists who work with data want to benefit from applying non-linear models for a vast amount of data but also, they would like to understand why model make a particular decision, what are key variables, what is their correlation and whether they can explain the modelling outcome. External validation and estimation of model applicability domain are not enough in areas such as biology, medicine or chemistry. The

A. Palczewska
School of Computing, Creative Technologies & Engineering, Leeds Beckett University, UK
E-mail: a.palczewska@leedsbeckett.ac.uk

U. Markowska-Kaczmar
Department of Computational Intelligence, Wroclaw University of Science and Technology, Poland
E-mail: Urszula.Markowska-Kaczmar@pwr.edu.pl

D. Neagu
Faculty of Engineering and Informatics, University of Bradford, UK
E-mail: d.neagu@bradford.ac.uk

mechanistic interpretation of the model (why the model makes a particular decision) is very important [31], but for non-linear models extraction of such knowledge is difficult to achieve. To interpret ANN models two approaches are available: methods based on rule extraction and on variable importance.

Rule extraction methods that attempt to interpret trained Neural Networks, or opaque models, have a long track record in machine learning and its applications. The definition of the problem can be found in [14]. The taxonomy of rule extraction from Neural Networks distinguishes the following: decompositional (local methods), pedagogical (global methods) [7] and eclectic methods. Local methods go deeply into a Neural Network structure. They try to describe each neuron behaviour in the form of rule based on the strengths of connection weights of the neuron. Then, a set of rules interpreting the whole Neural Network is obtained by the concatenation of rules describing particular neurons [28]. The pedagogical approaches treat the model as an oracle. Inputs and outputs are matched to each other, and rules are extracted using machine learning approaches [6,12]. The eclectic methods are based on both decompositional and pedagogical approaches [4,30]. The main disadvantage of this approach is a limited interpretation of a model for data with a large number of variables. Models built for datasets that contain thousands of variables (e.g., codes DNA, chemical compounds or binary data) are not readily interpretable by rules.

While extracted rules from trained ANNs allow a global description of model behaviour, estimation of input variable importance for ANN models explains the relative contribution of each variable to the prediction result. This importance also varies with the designed network architecture and with the initial random weights used to train the ANN. In [23] authors presented the interquartile range (IQR) method to rank variables based on their importance. This relative importance is defined as an average of an interquartile range of each of the network weights from an input node to hidden nodes for all hidden units for a given input node. This method was used to rank variables but does not explain the influence of a variable on predicted value. Additionally, this method is general for the ANN model and does not distinguish the variable importance between instances for which the model was applied. There are some methods based on partial derivatives in ANN sensitivity analysis to calculate variable contribution/importance [9,20]. The analysis is based on the joint contribution of every possible pairwise combination of variables. In [21] the relative importance of variable, calculated using various existing methods, was averaged to handle the instability problem of variable importance. The variable importance is nicely applicable to datasets with a large number of input variables as quite often the importance factor is used for feature selection. The variables with the most significant importance are further used to build more accurate models [5,32].

There is a lack of similar methods that allow interpretation of Neural Networks on an instance level. Assessment of input variable importance would be useful for researchers in biology, chemistry, medicine and many other areas who work with ANN models already trained to their specific classification task. All methods for extracting feature/variable contributions are on a model level which is not sufficient for more detailed analysis. Unfortunately, the structure of the Neural Network does not allow extraction of such information, because it is distributed in the network. An output of each artificial neuron is calculated by a non-linear function of the sum of its inputs. The weight for a particular connection is adjusted with a dependency to the other neurons in the network. Applying the pedagogical approach with rule extraction to address this problem, we can extract a rule from a set of rules for which conditions are fulfilled for a new instance. In such case, we may end up with complex relations of a large number of variables that again would be difficult to interpret. Ideally, we would like to have numerical values for each variable representing the influence on a prediction. Also, the rule extraction on a single instance level does not distinguish between two instances that belong to the same class. In this case, they will share the same conditions in a rule. As an example, let us consider two toxic chemicals with similar structures. We would like to know which part of structures are the most toxic by extracting contributions of chemical substructures toward the toxicity. The substructures are input variables into a model. Applying the rules approach we could find that they share the same conditions that classify them toxic, but when we look at variable contributions we may see differences in substructures toxicity. Another example of need interpretation of model on instance levels comes from personalised healthcare. Let us consider a case from the healthcare-prediction of risk that someone can get cancer. General model interpretation can show all important features for a model but for specific two patients, there may be different factors leading to getting cancer (each patient can have a different subset of variables that are important for a model to make a decision). Having such knowledge can support medical decisions to provide different treatment for these patients.

Feature Contributions was proposed by Kuzmin et al. in 2011 [17]. It was designed to extract Feature Contribution for Random Forest models for regression problems. Feature contributions were defined as numerical values, computed separately for each instance/record. Kuzmin's method has been extended to

Random Forest classification models in [22]. This method was used further in work [18] where authors compare the chemical interpretability of the predictions, using scoring schemes for assessing heat map images of substructural contributions. Feature contributions allow us to extract a relationship between a particular feature value and a model's decision. For each instance, we calculate how much a given variable/feature contributed towards the predicted outcome. We can see which features have a positive/negative impact on a predicted value, and which of them have a stronger influence. They also may be normalised and transformed to represent a scale in a particular domain (e.g., the toxicity of chemicals, the presence of mutated genes in DNA corresponding to particular cancer). As an example, in [17], Feature Contributions were positively tested on a Quantitative Structure-Activity (QSAR) model for chemical compounds. The results showed a relationship between the presence of the chemical structure and toxicity, and they provided valuable information for the design of new compounds. In [18] authors compare various models performance (Random Forest, support vector machines, support vector regression, and partial least squares) regarding their predictivity as well as the chemical interpretability of the predictions, using novel scoring schemes for assessing heat map images of substructural contributions. Authors extracted Feature Contributions for instances used to predict mutagenicity and applied the heat map to colour chemical substructures according to the toxicity. Feature contributions have been also used in [16] to interpret Random Forest model predicting element concentrations, loss of ignition and pH in the soils of south-west England using high-resolution remote sensing and geophysical survey data. Also in [33], authors used Feature Contributions to elucidate the contribution of individual molecular descriptors to the predicted potency. Feature contributions provided easily interpretable suggestions of critical structural properties for potent permeation enhancers such as segregation of hydrophilic and lipophilic domains.

In this paper, we provide an attempt to the interpretation of a trained Neural Network model on an instance level. To achieve a solution we propose the use of the Neural Network as an oracle within a pedagogical approach [31] that is used for ANN rule extraction. This assumption means the Neural Network produces the examples of inputs and corresponding outputs for training the Random Forest (RF) model. The RF model as an ensemble of decision trees is more accurate than a single decision tree, does not overfit and has a less redistributed error as shown in [3] and more recently in [1, 5, 24]. RF allows calculating Feature Contributions (FC) using the method described in [22]. Feature Contributions show what is the influence of a single input variable for a given instance on the final classification. Because the FCs are acquired from RF that mimics the activity of ANN, we have to check whether we can trust the result offered by FCs. We use Feature Contributions to build a classifier. If for a new input vector the ANN responds with the same class as FCs, this is an indication that we can trust the interpretation delivered by FCs. The side benefit of the RF model is that it enables easy rule extraction that describes relationships between inputs and outputs of a trained Neural Network. It is worth to underline that rules do not focus on a single instance. We also show in an Iris dataset case study how, using Feature Contributions, one can verify the conditions in the premise part of extracted rules.

This paper is organised as follows. Section 2 describes the proposed method for the ANN model interpretation. It provides the formal problem statement and includes the definition of a Random Forest model, Feature Contributions, and their analysis. Section 3 describes the experimental study and discusses the obtained results. Section 4 concludes our work.

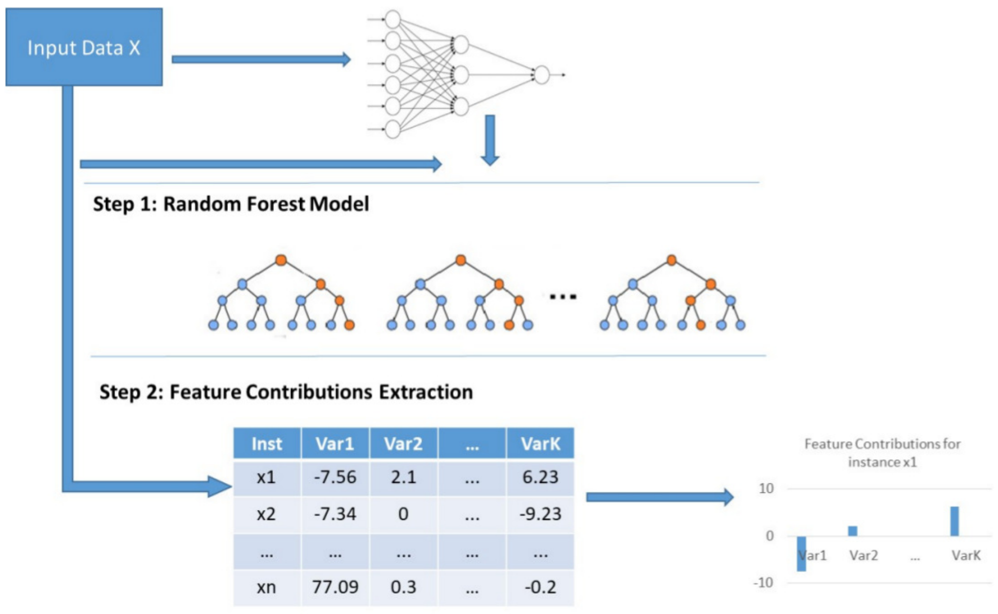
2 Methodology

In this section, we recall the definition for Feature Contributions, and we describe how the Feature Contributions can be used to interpret Neural Network.

2.1 Random Forest Feature Contributions

Firstly, we recall the definition of the Feature Contributions presented in [17,22]. Feature Contributions calculated for a given instance represent the influence (negative or positive) of each feature (input variable) on a predicted target. They are computed in two steps. Firstly, local increments are calculated for each node in the forest's trees using the trees training datasets:

$$LI_{fc} = \begin{cases} Y^{c}_{mean} - Y^{p}_{mean}, & \text{if the split in the parent is performed} \\ & \text{over the feature } f, \\ 0, & \text{otherwise,} \end{cases}$$


Input Data X
Step 1: Random Forest Model
Step 2: Feature Contributions Extraction
Feature Contributions for instance x1
10
0
-10
Var1
Var2
...
VarK

Inst	Var1	Var2	...	VarK
x1	-7.56	2.1	...	6.23
x2	-7.34	0	...	-9.23
...	...	...	...	...
xn	77.09	0.3	...	-0.2

Fig. 1: A schema for the ANN model interpretation method via Random Forest model and Feature Contributions

where Y_{mean} is a fraction of the training instances in a given node c (c - is a child node and p - is a parent node) belonging to a selected class (for details see [22]) or an average over the instances within the node for regression models. A local increment for feature f represents the change of the probability of being in a given class between the child node and its parent node in a tree.

Secondly, for any instance and a variable f these local increments are summarised on tree paths:

$$FC_{if} = \frac{1}{ntree} \sum_{k=1}^{ntree} \sum_{l=1}^{knode} LI_{if_{kl}}, \tag{1}$$

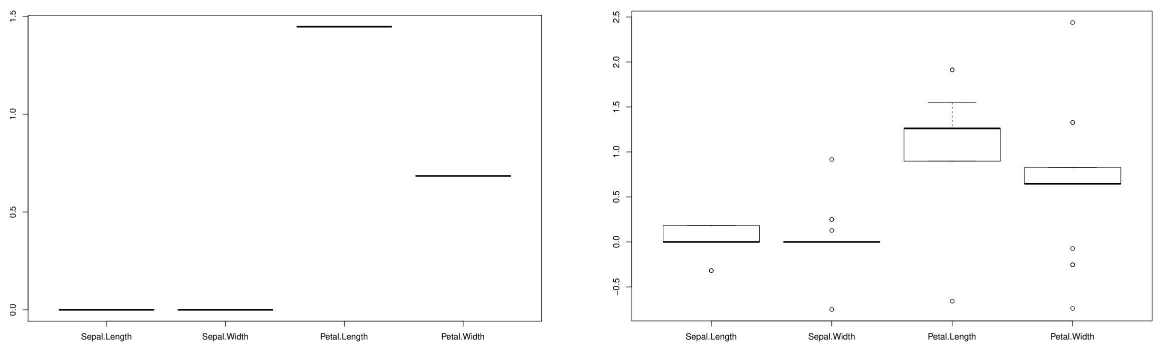
where the value $LI_{if_{kl}}$ is a local increment for the instance i, feature f in k tree and its l node. The values $ntree$ and $knode$ represent the number of trees in the forest and the number of nodes from the k tree which split over a feature f, respectively. In other words the Feature Contributions vector FC_i for an instance $\mathbf{x}_i$ represents the contribution of each feature towards/against the predicted outcome.

2.2 ANN Model Interpretation for a Single Instance

Let us assume that the ANN model trained for a specific classification problem is given. Our goal is to develop a new method of interpreting the ANN model on a single instance level. Feature Contributions offer such ability but they are calculated from the Random Forest model and we can not apply it directly to the ANN model. Our idea is to train a Random Forest model to mimic the behaviour of ANN (similarly to the solution applied in pedagogical approach in the rule extraction), then to calculate Feature Contributions, thus giving an opportunity of ANN interpretation.

The workflow of the ANN model interpretation is presented in Figure 1. In step 1, we build the Random Forest model using input data $\mathbf{x}$ and output $\mathbf{y}$ of the ANN model. In step 2, we extract Feature Contributions from the Random Forest model using Kuzmin/Palczewska approach [17,22] for any record from $\mathbf{x}$ or for any new record $\mathbf{x}_{new}$ for which model was used to predict an outcome. Large positive values define stronger Feature Contribution for the predicted class whereas negative values define contributions towards the other class or classes.

Once we have calculated Feature Contributions we need to assess the certainty of the ANN model interpretation. In this case, we perform classification based on FCs representatives extracted from the Random Forest training dataset, including records for which Random Forest agreed with Neural Network on prediction and with the original output. The evaluation is positive if the class predicted by ANN and by FCs are the same.



(a) Setosa (b) Virginica

Fig. 2: An example of Feature Contribution variations. Box plots of Feature Contributions for two classes of IRIS dataset [1]. The axes x and y represent the IRIS features and values of their contributions, respectively.

2.2.1 Classification by Feature Contributions

Firstly, we calculate FCs for RF training dataset. Then for each class, we select records for which predictions from RF agree with predictions from ANN and their original output variable. In the next step, we determine the FC representatives. As described in [22] we can consider two Feature Contribution representative's types: median and cluster centroids, computed for each class separately. Then:

– if there is no variation within Feature Contributions which means that all values are distributed around the FC mean (see for example Figure 2a) then as Feature Contributions representative we use a median. It is a better estimator than mean as mean can be biased by outliers.
  To classify a new i-th instance $\mathbf{x}^i_{new}$ based on its Feature Contributions, we calculate Feature Contributions first. Then, the Euclidean distance d_E is computed for all class representative's medians, and minimal distance is selected:

$$d^i_E = \min_l \sqrt{\sum_{f=1}^{nvar} (FC_{if} - m_{fl})^2}, \tag{2}$$

  where FC_{if} is calculated using Equation (1), $nvar$ is a number of features (variables) in the input vector and m_{fl} is a Feature Contributions median (representative of a class) of f-th feature and l-th class. The smallest distance indicates the class of the new data i predicted by the Feature Contributions method.
– otherwise, there is a variation within Feature Contributions (see Figure 2b for $Virginica$ class as an example). A majority of instances have values close to FC mean, and there are few elements with different values. These few elements can produce a small group with another pattern of Feature Contributions different from these created by the majority group. The group with the minimum variance is called a core cluster [22] and its centre is used as the class representative. If clusters have the same variance (e.g equal to zero) we can have more then one representative for a class. For each class, the optimum number of clusters is obtained using the elbow method [11,24]. Selected training instances are assigned to these clusters. To classify a new instance $\mathbf{x}^i_{new}$, the Euclidean distance is calculated to all cluster centres and the smallest distance is selected:

$$d^i_E = \min_l \sqrt{\sum_{f=1}^{nvar} (FC_{if} - c_{fj_l})^2} \tag{3}$$

  where c_{fj_l} is a centroid of a cluster j of class l, $nvar$ is the number of variables f. The smallest distance indicates which cluster a new data $\mathbf{x}^i_{new}$ belongs to and defines a class for the new instance. It is worth noticing that other distance metrics can also be considered, depending on the distributional properties of data. We do not study this problem in the paper leaving it for the future research.
  To illustrate how to use centroids as representatives, let us consider the example of *Virginica* class in detail. Table 1 shows examples of patterns in Feature Contributions for the Virginica class from the IRIS dataset. There were 42 elements in the training dataset for the Random Forest model that were

Table 1: The number of instances in Feature Contributions groups for Virginica class of IRIS dataset

S.Length	S.Width	P.Length	P.Width	Count
0	0	1.26	0.65	25
0.18	0	0.9	0.83	9
-0.32	0	0.9	1.3	2
0	0	1.08	0.83	1
0.18	0.92	548	-0.74	1
0	-0.75	1.91	-0.25	1
0.18	0.25	1.55	-0.07	1
0	0.25	1.91	-0.25	1
0	0.3	-0.66	2.43	1

Algorithm 1 The method (in pseudocode) of ANN interpretation using Feature Contributions

```
Require: ANN, $D_{RF}, Y_{RF}$ and $D_{New}, Y_{New}$
 1: Train a Random Forest model RF on $D_{RF}$, $Y_{RF}$ datasets
 2: Calculate Feature Contributions FC from the trained RF model
 3: Find the class representative $FC^c_{rep}$ for Feature Contributions (medians or cluster centres)
 4: for each instance $\mathbf{x}^i_{new}$ in $D_{New}$ do
 5:     calculate Feature Contribution $FC_i$ for an instance $\mathbf{x}^i_{new}$
 6:     for each class $c$ in datasets classes $C$ do
 7:         Calculate Euclidean distance between Feature Contributions $FC_i$ for the instance $\mathbf{x}^i_{new}$ and class representative:
            $d_E(FC_i, FC^c_{rep})$
 8:     end for
 9:     Select the class $c$ for which the distance is minimal.
10:     if class $c$ is equal to the predicted ANN model class $\mathbf{y}_i$ for the instance $\mathbf{x}^i_{new}$ then
11:         $p_i = 1$
12:     else
13:         $p_i = 0$
14:     end if
15: end for
```

correctly classified. We can notice that there are two main groups with cardinality 25 and 9 elements. The clusters that have minimum variance become core clusters and core clusters are further used to evaluate whether we can trust in the interpretation of ANN offered by FC.

2.3 The Method of ANN Interpretability in Detail

The idea of the method is focused on training the Random Forest on the data produced by the ANN model (the ANN works as an oracle) and then, on the Feature Contributions extraction for the trained Random Forest model.

The general description of the method is shown in Algorithm 1. Let us assume that a trained ANN model and the input training datasets D_{RF} for Random Forest are given. Based on the set D_{RF} ANN produces corresponding output dataset Y_{RF} for training RF model. In the algorithm, we assume to process more than one new instance $\mathbf{x}_{new}$ for which we want to obtain the interpretability of a Neural Network output. These new instances create a set called D_{new}. The corresponding outputs are generated by ANN. They create Y_{new} dataset.

Then (line 1), we train a Random Forest model. During training, the Random Forest learns to map the input $\mathbf{x}$ into the output $\mathbf{y}$. It approximates the Neural Network behaviour. Each tree is grown using the classification and regression tree (CART) method [3].

Next (line 2) the FCs are calculated. They reflect the features influence on the ANN prediction. Feature Contributions are further used to analyse input data to search for the most powerful features within classes or among entire dataset.

Further (line 3), for each class the FC representatives are calculated. We use the centre of the FC cluster or FC median for a given class as describe in Section 2.2.1. The representatives are determined from the instances of the Random Forest training dataset for which the RF prediction agrees with the ANN model prediction and the original value of the output variable.

Next, FC is calculated also for the new input $\mathbf{x}_{new}$ (line 4). Then, the smallest distance between FC for the new input and the representative assigns the class for the new input data (lines 5-9). If the class is the same as the class predicted by the ANN, we trust the FCs interpretation result ($p_i = 1$), in another case, it is not true ($p_i = 0$) (lines 10-14).

3 Experimental Study

In this section, we will test whether the Feature Contributions method can be used to interpret a trained ANN model. We will use a selected number of datasets from UCI benchmarks datasets (described in the next section) that were also used in papers regarding the development of ANN models and rule extraction.

3.1 Datasets

In the experiments, we used eight datasets from the UCL Machine Learning Dataset Repository [1]. We selected the datasets that were often used as benchmark sets in rule extractions for ANN models [15]:

1. Breast Cancer Wisconsin (Diagnostic) Dataset (BCWD). Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe the characteristics of the cell nuclei present in the image.
2. COX2 [29]. A set of 467 cyclooxygenase-2 (COX-2) inhibitors have been assembled from the published work of a single research group, within vitro activities against human recombinant enzyme expressed as IC50 values ranging from 1 nM to > 100 uM. To classify the data, a cutoff of $2^{2.5}$ was used to determine activity.
3. German Credit Scoring. The dataset classifies people described by a set of attributes as good or bad credit risks. This file has been edited, and several indicator variables added to make it suitable for algorithms which cannot cope with categorical variables. Several attributes that are ordered categorical (such as attribute 17) have been coded as an integer.
4. IRIS. The dataset contains three classes of 50 instances each, where each class refers to a type of iris plant.
5. SEEDS. Measurements of geometrical properties of kernels belonging to three different varieties of wheat. A soft X-ray technique and GRAINS package were used to construct all seven, real-valued attributes.
6. Teaching Assistant Evaluation (TEACHING). The data consist of evaluations of teaching performance over three regular semesters and two summer semesters of 151 teaching assistant (TA) assignments at the Statistics Department of the University of Wisconsin-Madison. The scores were divided into 3 roughly equal-sized categories (”low”, ”medium”, and ”high”) to form the class variable.
7. WAVEFORM Database Generator (Version 1). CART book’s waveform domains.
8. WINE. Results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

3.2 Experimental Procedure

To satisfy the assumption of the method, we trained the Neural Network first and then we generated the dataset D_{RF} that is used to train RF. In order to train the network, all files chosen for experiments were divided into three files: training D_{train}, Y_{train}, testing D_{test}, Y_{test} and validating D_{new},Y_{new} datasets. The network was trained using training dataset D_{train}, and it was tested on dataset D_{train}.

When the training the network was completed, we joined $D_{train} \cup D_{test}$. It creates D_{RF} input dataset that is further used for training Random Forest model. Next, we delivered instances from D_{RF} to the ANN and for each input vector $\boldsymbol{x} \in D_{RF}$ the network produced a response that was included in Y_{RF}. In the experiment, ANN interpretation is tested on instances from D_{new}.

3.3 Training the Artificial Neural Network Model

For the ANN model, the multi-layer perceptron (MLP) network has been used. MLP is a fully connected feed-forward network and the most common network architecture in use. Training is performed by the backpropagation method [13]. In this study, we trained the MLP model using the *MLP* function from the RSNNS package in R [27]. We used the default settings for the *MLP* model. We only set a parameter *size* (describing the number of hidden neurons) to be equal to an average of sum of input and output variables, learning coefficient - *learnFuncParams* equals 0.1 and the maximal number of iteration equals

Table 2: Characteristics of datasets and average accuracy (ACC) of ANN over 50 runs of the of ANN model development procedure. The columns represent: the number of instances in the dataset (Inst), the number of instances for the training and testing dataset for the ANN model ($\#D_{train}$, $\#D_{test}$), the number of instances for a validating dataset ($\#D_{new}$), the number of dataset's attributes and classes (#Attr, #Class), an average accuracy for the training dataset (ACC_{train}) and accuracy for the testing dataset ACC_{train} for the ANN model

.

Name	#Inst	$\#D_{train}$	$\#D_{test}$	$\#D_{new}$	#Attr	#Class	ACC_{train}	ACC_{test}
BCWD	683	409	137	137	9	2	0.981	0.966
COX2	190	114	38	38	255	2	1.000	0.677
German_CS	1000	600	200	200	20	2	0.878	0.737
IRIS	150	90	30	30	4	3	0.958	0.941
SEEDS	210	126	42	42	7	3	0.953	0.916
TEACHING	151	90	30	31	5	3	0.576	0.491
WAVEFORM	5000	2998	1000	1002	21	3	0.904	0.856
WINE	178	105	36	37	13	3	1.000	0.980

Table 3: Average AUC for the ANN model for all datasets.

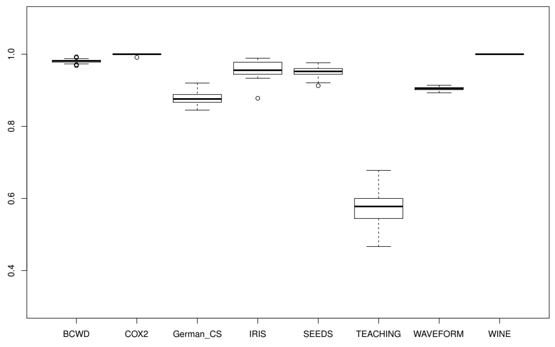
Data Set	IRIS	BCWD	German_CS	COX2	WINE	TEACHING	SEEDS	WAVEFORM
D_{train}	0.977	0.984	0.846	0.999	1	0.745	0.955	0.899
D_{test}	0.983	0.962	0.691	0.705	0.9755	0.677	0.875	0.883
D_{new}	0.987	0.963	0.662	0.666	0.989	0.644	0.849	0.841

50. The ANN model had only one hidden layer. The number of ANN's output neurons was equal to the number of classes in a given dataset because we used 1 of n encoding for the output layer. We did not focus on the *MLP* model accuracy, so we did not optimize the model parameters to get the most accurate model (the model accuracy was not the subject of this study).

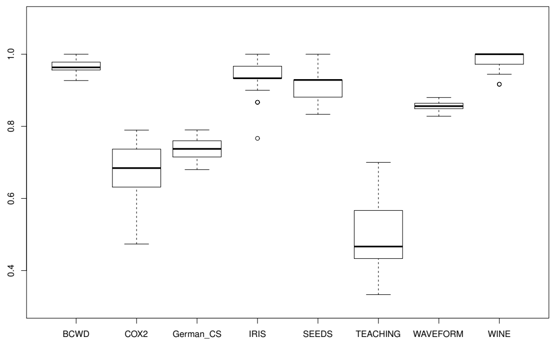
Table 2 presents the averaged results from building the *MLP* model. First four columns show the cardinality of each dataset and the split for training (D_{train}), testing (D_{test}) and validating (D_{new}) datasets. Testing D_{test} and D_{new} datasets were randomly selected taking 20% of data for both datasets. To have an equally represented set of elements in each class this selection was conducted for each class separately. The fifth and sixth columns in the table represent the number of attributes and classes for each dataset, respectively. The last two columns show the averaged accuracies for the *MLP* model for training and testing datasets obtained from the repeated approach of the model development process. This means that for each dataset we repeated the procedure fifty times by splitting the dataset and generating the ANN model. Figures 3a and 3b present boxplots for all generated models for each dataset-case for training and testing datasets.

One can see that the *MLP* model gives good results for BCWD, IRIS, SEEDS, WAVEFORM and WINE with averaged accuracies around 0.9. Surprisingly, the model has high accuracy for the COX2 training dataset 1, but much worse results for testing dataset 0.7. This result may suggest that *mlp* model was overtrained for this dataset. The TEACHING dataset got the worst results. It resulted from the way of features encoding for ANN. All attributes in this dataset are categorical, and we use their numeric representation without conversion to binary sub-vectors. A similar situation is with German Credit Scoring dataset. In this case, the results show the higher accuracy of a model, but still lower in comparison to other datasets. An explanation for this regards to how Neural Networks work. ANNs process numbers, sum up weighting signals expressed as numbers and then calculate the value of the activation function. Therefore it is not a good idea to encode categorical variable on one neuron. Between categorical values, there is no order between numbers. This is why we need as many neurons as many values have a categorical variable to encode such variables. Each neuron can take only 0 or 1 representing the given value. We did not encode the input variable into vectors as we wanted to generate some various "good" and "bad" models to demonstrate their interpretability.

To evaluate the ANN classification quality, we calculated the Area Under ROC Curve (AUC) for ball datasets. They are presented in Table 3. In literature [2], the model with the AUC measure greater than 0.7 is considered to be a good model. A multi-class AUC is computed as an average AUC as defined by [10] and implemented in the multiROC package in R [19]. The AUC values for all datasets show that predictivity of the ANN model for GERMAN, COX2 and TEACHING dataset is worse than for the other datasets but yet still good enough to consider the ANN model suitable for all datasets.



(a) Training dataset D_{train}



(b) Testing dataset D_{test}

Fig. 3: The box plot shows accuracies of fifty ANN models for all selected datasets for a) training dataset D_{train} and b) testing dataset D_{test}, separately.

Table 4: Average accuracy for ANN (ANN_{new}) and RF (RF_{new}) models for validation dataset D_{new} and for RF training D_{RF} dataset (RF_{train} column)

Name	$\#D_{new}$	ANN_{new}	RF_{train}	RF_{new}
BCWD	137	0.970	0.996	0.976
COX2	38	0.676	0.947	0.766
German_CS	200	0.736	0.965	0.814
IRIS	30	0.946	0.996	0.967
SEEDS	42	0.932	0.986	0.922
TEACHING	31	0.533	0.980	0.934
WAVEFORM	1002	0.859	0.973	0.841
WINE	37	0.982	0.990	0.941

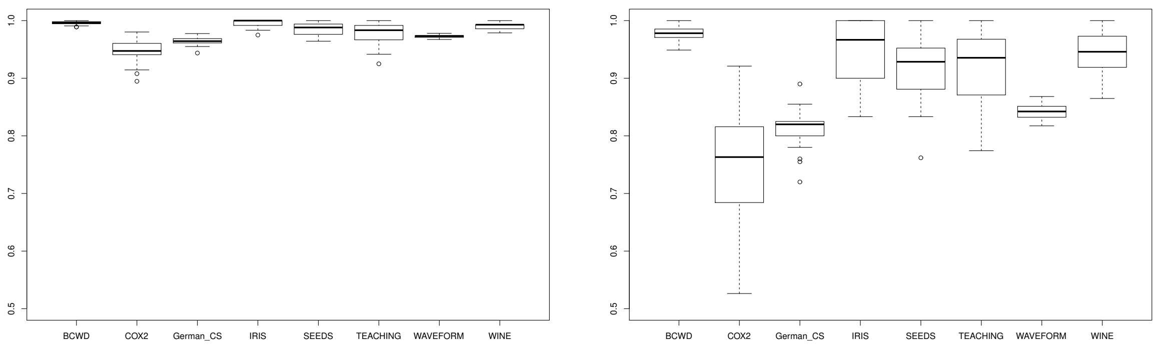
3.4 Training Random Forest Model and Calculating Feature Contributions

Random Forest model was trained on a combined dataset D_{train} and D_{test} called D_{RF} and Y_{RF} - an output of ANN for D_{RF} as described in Section 2.3. We used *randomForest* [25] package in R. The number of trees was set to the number of input variable for each dataset separately. The reason lies in avoiding the overfitting for datasets like IRIS with a small number of variables. We used default settings for this method. We set the parameter *replace*=False to avoid selection with a replacement for training trees. We also keep information on records which were used to train a tree in a forest by setting the parameter *keep.inbag*=True. This is needed to calculate Feature Contributions.

Table 4 shows the averaged results for Random Forest model and for the *MLP* model. Column $\#D_{new}$ informs how many instances contains the D_{new} dataset. The averaged accuracy of *MLP* model for D_{new} is included in the column (ANN_{new}). The column RF_{new} describes average accuracy for the Random Forest model. The table shows also the average accuracy of the Random Forest model for training data (column RF_{train} achieved on the D_{RF} dataset. Models do not predict well for COX2 and TEACHING datasets. The averaged accuracy of the Random Forest model for COX2 is the lowest among all datasets, whereas for the other datasets is in the range of 0.95 which means that RF model mimics well the ANN model's behaviour. Figures 4a and 4b show the summary of accuracies for Random Forest models in the repeated runs of the algorithm for each dataset.

To test how well the Random Forest model mimics the ANN model, we calculated the average Area Under Curve for each RF model. Table 5 presents AUC for each dataset. The higher AUC value closed to one, the less noise/error was introduced by the Random Forest model, and the better interpretability of the ANN model we can expect. As the ground truth, the instances from Y_{RF} and Y_{new} were considered, respectively.

It can be noticed that the AUC values for the Random Forest model are in some cases higher than ANN's AUC values (see Table 3) but RF model was trained with a greater number of instances - D_{RF} is a combination of D_{train} and D_{test} used for the ANN model. Similarly to ANN also for RF, the AUC values were calculated using *multiROC* package [19] in R, where the AUC for multiclass is calculated as an average of the pairwise AUCs.



(a) Training dataset D_{RF} (b) Validation dataset D_{new}

Fig. 4: Box plots for RF models for all datasets for a) RF training dataset D_{RF} and b) validation dataset D_{new}. The y axis represents the Random Forest model accuracies.

Table 5: Average AUC for the RF model calculated for each dataset

	IRIS	BCWD	German_CS	COX2	WINE	TEACHING	SEEDS	WAVEFORM
D_{RF}	0.995	0.996	0.846	0.938	0.987	0.976	0.985	0.970
D_{new}	0.956	0.976	0.719	0.743	0.953	0.903	0.909	0.836

3.5 Certainty Assessment of ANN Interpretation

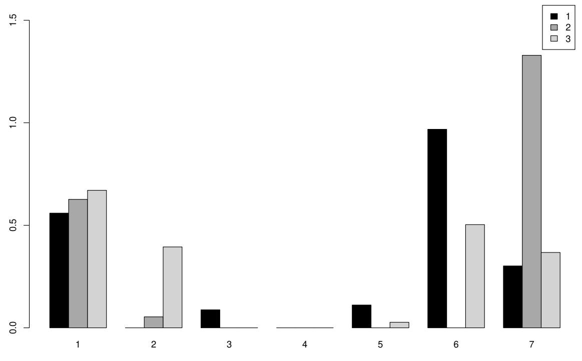
Feature Contributions calculated for the instance $\mathbf{x}_{new}$ give information about the relation between predicted class and input features for an RF model. Because the RF model only mimics the ANN model we are interested in evaluating how much we can rely on this interpretation. In other words, we would like to assess the certainty of obtained interpretation. To decide if the extracted Feature Contributions for an instance $\mathbf{x}_{new}$ give certain interpretation we test them against ANN model prediction for this instance. The verification of the ANN model prediction is based on the comparison of the classification of $\mathbf{x}_{new}$ data made with the ANN model and the class found by the Feature Contribution analysis. If the prediction from FC agrees with the prediction from ANN for an instance $\mathbf{x}_{new}$, we say that interpretation is *certain* for this instance. If the predicted class from ANN agrees with FC prediction and with the original class for this instance, we say that prediction is *correct*.

Following the Algorithm 1 we calculated Feature Contributions for instances from the Random Forest training dataset D_{RF} and the validation dataset D_{new}. We used the rfFC R package [26]. We selected these instances from D_{RF} for which predictions from RF and ANN models agree with the original value of the output variable. Then for each class, we calculated Feature Contributions medians. In the second step, we applied *k-means* to cluster Feature Contributions within each class. For each Feature Contribution subset with non zero variance, the number of clusters was assessed using the *MClust* R package. This method uses the Bayesian Information Criterion (BIC) - a model selection criterion to search for the best number of clusters [8]. The number of clusters was selected based on the optimal number of clusters between 1 and 10. Finally, we extract the Feature Contributions representatives for each class. In Figure 5 we present medians representatives of Feature Contributions for a few datasets and all classes (for each dataset). Contributions can be positive as well as negative values and representatives differ between classes.

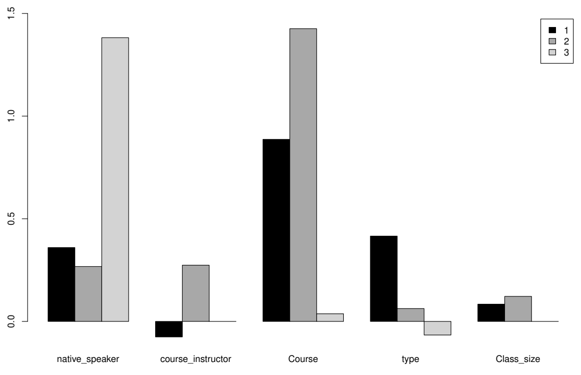
Next, Feature Contributions for each $\mathbf{x}_{new}$ instance from D_{new} were calculated. Then, we calculate distances between representatives and Feature Contributions for instances from D_{new} using equations (2) and (3). In following subsections, we demonstrate how the method works for a few instances from the Iris dataset and we present the aggregated results for repeated runs of the algorithm.

3.5.1 Iris Dataset Example

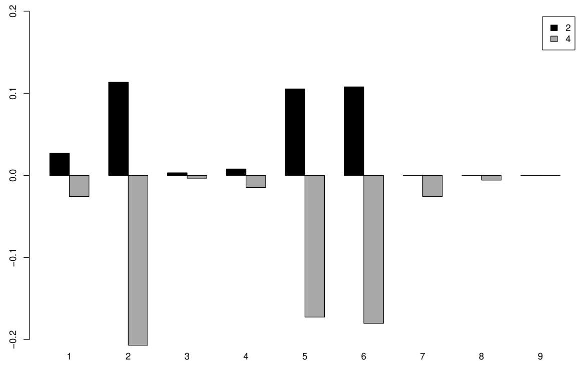
As an example, let us consider a few instances $\mathbf{x}_{new}$ for a single method run from Iris dataset. Table 6 shows the original data for Iris instances and prediction from the ANN model. The ANN model's accuracy was 0.93. Three exemplary instances were selected from the D_{new} dataset. There are two instances from the Virginica class (second and third). The ANN model classified the third instance as Versicolor. The Random Forest model accuracy was 0.997 for D_{new}. Feature contributions were extracted



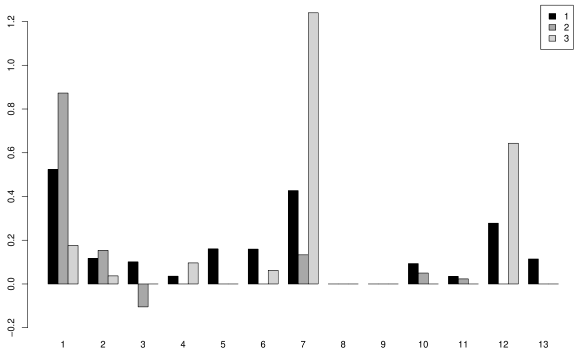
(a) Feature contributions medians for SEED dataset. The variable numbers represent: area, perimeter, compactness, length of kernel, width of kernel, asymmetry coefficient and length of kernel groove, respectively.



(b) Feature contributions medians for TEACHING



(c) Feature contributions medians for BCWD. The variable numbers represent: Clum Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, Mitoses, respectively.



(d) Feature contributions medians for WINE. The variable numbers represent: Alcohol, Malicacid, Ash, Alcalinityofash, Magnesium, Totalphenols, Flavanoids, Nonflavanoidphenols, Proanthocyanins, Colorintensity, Hue, Diluted Wines, Proline, respectively.

Fig. 5: Example of Feature Contributions median for selected datasets

from the RF model for each of these instances (see the bottom part of the Table 6). One can notice that in the first instance the fourth variable is the most important for predicting this instance with Setosa class. The second instance has the third variable as the most important and also the first and the forth variables contribute to predicting class Versicolor. The third instance originally being in the same class as the second instance, has the forth variable the most significant and also the second one. But for this instance, the ANN predicts wrongly.

Table 6: Instances (Data) from D_{new} Iris dataset, their ANN prediction and Feature Contributions (FC) calculated for each instance.

	No	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Class	ANNPrediction
	1	4.6	3.4	1.4	0.3	Setosa	Setosa
Data	2	7.2	3.6	6.1	2.5	Virginica	Virginica
	3	6.3	2.8	5.1	1.5	Virginica	Versicolor
	1	0	0	0.2193	0.4386		
FC	2	0.04318	0	0.3243	0.2816		
	3	-0.0256	0.11	-0.1278	0.4031		

As the model *representative* for Feature Contributions, we have taken median calculated on the base of D_{RF} training dataset for each class. In the next step, we calculate distances between each instance and the class representatives. The representatives for each class are shown in Table 7. For Setosa class, the third and the forth variables are the most significant and the fourth one has much stronger contributions. For Versicolor class, the third variable contributes the most but also the second and the forth features are important. Similarly, for Virginica class the third variable contributes the most and also the first and the third are important - having values greater than zero.

Table 7: Median representatives for RF model's training dataset D_{RF} for IRIS dataset.

Class	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Setosa	0	0	0.2193	0.4386
Versicolor	0	0.02667	0.4132	0.2531
Virginica	0.04318	0	0.3243	0.2816

To verify the certainty of the ANN interpretation we calculate distances between Feature Contributions of each instance and all class representatives (see Formula 2). The distances are presented in Table 8. The minimum distance defines the class for a considered instance. The first two instances have Feature Contributions equal to their class representatives. This is possible when many elements follow the same path in the forest's trees. In this case, the class assigned based on a distance agrees with the ANN model predictions. In the case of the third instance, the distances to all class representatives are big. Following the algorithm, the class assigned by using Feature Contributions is Setosa which differs from the ANN prediction. In this case, we do not trust the ANN interpretation.

Table 8: Distances to the median representatives calculated for instances from Table 6.

Nr	Setosa	Versicolor	Virginica	FC_Class	Response p
1	0	0.2697	0.1937	Setosa	1
2	0.1937	0.1063	0	Virginica	1
3	0.3667	0.5681	0.4858	Setosa	0

The procedure to calculate distances to the cluster representatives is similar, so we do not present it in this section. The only difference is that we can have more than one representatives for a class. It is related to the core clusters as it was described in Section 2.1. In the next section, we will evaluate the ANN interpretation method for all datasets.

3.5.2 Interpretability Method Evaluation for all Datasets

In this section, we repeat the procedure described in Algorithm 1 for all eight chosen datasets. Table 9 shows averaged results from repeated runs of the method for each dataset. The values were rounded to the nearest integer. The first column in this table shows the number of elements in the new dataset D_{new}. The second (Med_Certain) column shows the number of instances that were marked certain with the median approach. The third (Med_Correct) column shows how many instances were correctly classified by the median approach concerning the original class value. The last two columns show the number of interpretations that were marked as certain based on the clustering approach (Clust_Certain) and the number of correctly classified instances in respect to the original class (Clust_Correct). Also, Table 10 presents detailed results from the certainty assessment of interpretability method. For each dataset, columns represent a number of instances for which ANN interpretation was marked as certain and uncertain for both median and clustering methods. In rows, we have ANN prediction expressed by a number of instances that were classified wrongly by the ANN model.

The aggregated results confirm that the presented method is suitable to interpret ANN model for new data. For ANN models with good predictive accuracy such as for IRIS, BCWD, WINE, SEEDS, the certainty of ANN interpretation is greater than 80%. This means that Feature Contributions represents the true importance for the ANN model. For bad bad models (see TEACHING and German_CS datasets), the certainty is greater than 60%. It is worth noticing that models for these two datasets had a low predictive accuracy (see Figure 3). This demonstrates that the proposed approach of assessment of the ANN model interpretability can filter instances with correct ANN prediction and with certain Feature Contribution values. Also, the results show that the use of clustering seems to work better than the use of the median approach.

Table 9: Number of elements from the D_{new} dataset marked as a correctly predicted by ANN model via median and clustering methods in respect to their original class label

Name	$\#D_{new}$	Med_Certain	Med_Correct vs Orig	Clust_Certain	Clust_Correct
BCWD	137	130 (94,8%)	127	133 (97%)	130
COX2	38	28 (73,6%)	21	29 (76,3%)	23
German_CS	200	161 (80,5%)	127	180 (90%)	145
IRIS	30	27 (90%)	26	28 (93,3%)	27
SEEDS	42	35 (83,3%)	33	39 (92,8%)	37
TEACHING	31	22 (70,0%)	19	27 (87%)	22
WAVEFORM	1002	674 (67,2%)	585	745 (74,3%)	663
WINE	37	31 (83,7,4%)	29	34 (91,8%)	33

Table 10: Certain/Uncertain vs correct/non correct prediction for elements of D_{new} dataset.

Name	#Valid	ANN Pred	Median		Cluster	
			Certain	Uncertain	Certain	Uncertain
BCWD	137	correct	127	2	130	1
		non correct	3	5	3	3
COX2	38	correct	21	3	23	3
		non correct	7	7	6	6
German_CS	200	correct	127	32	145	19
		non correct	34	7	35	1
IRIS	30	correct	26	2	27	2
		non correct	1	1	1	0
SEEDS	42	correct	33	4	37	2
		non correct	2	3	2	1
TEACHING	31	correct	19	1	22	0
		non correct	3	8	5	4
WAVEFORM	1002	correct	585	107	663	98
		non correct	89	221	82	159
WINE	37	correct	29	4	33	2
		non correct	2	2	1	2

3.6 Rule Extraction vs Feature Contributions

In this section, we illustrate the IRIS dataset case study how Feature Contributions can support interpretation offered by rules extracted from the Random Forest model. The rules give us a general description of the model decision-making process. Table 11 presents rules extracted from the forest with three trees for IRIS dataset. Rules $1-5$, $6-11$ and $12-16$ are extracted from the first, the second and the third tree respectively. Variables $1, 2, 3, 4$ correspond to Sepal.length, Sepal.width, Petal.length and Petal.width. After pruning the number of rules decreased (see Table 12), still some of them may partially overlap, but we are not investigating such a case in this paper.

The other question we raised in this paper is whether Feature Contributions can support or strengthen the general description of a model behaviour included in the set of rules. Feature contributions present relationships between single instance variables and their model prediction. The generalisation of Feature Contributions may be useful to visualise which variable conditions are the strongest in the rule if we want to apply rule pruning. The bars in Figure 6. present the median of Feature Contributions, calculated for the Random Forest training dataset D_{RF}, for each class for the IRIS dataset. Petal.width and Petal.length are significant variables for all classes. For the Setosa class, the Petal.length has the biggest contribution. Let us look at the rules presented in Table 12. We can see that rule 12 has Petal.length in its premise to classify instances in the Setosa class. For the other two classes the Petal.width variable is the most significant. We can also find that Versicolor is also described by Sepal.length at rule 2. The median plot supports this showing the small contribution for this variable. The similar situation exists with Virginica class. Petal.Width is the most important feature and its premise is present in all rules for this class. Petal.Length as the second important feature for this class is present in rules 11 and 14. Finally, Sepal.Width variable is present only in rule 14 and there is a small contribution for this feature for class Virginica. The drawback of this approach is that the relation between rules and Feature Contributions representatives is easily visible only if there is a small number of rules after pruning.

Table 11: Extracted rules for IRIS dataset.

1	$Petal.width <= 1.75$ & $Petal.width <= 0.8$	Setosa
2	$Sepal.length <= 6.65$ & $Petal.width <= 1.75$ & $Petal.width > 0.8$	Versicolor
3	$Sepal.length > 6.65$ & $Petal.length <= 4.95$ & $Petal.width <= 1.75$ & $Petal.width > 0.8$	Versicolor
4	$Sepal.length > 6.65$ & $Petal.length > 4.95$ & $Petal.width <= 1.75$ & $Petal.width > 0.8$	Virginica
5	$Petal.width > 1.75$	Virginica
6	$Petal.length <= 2.6$	Setosa
7	$Sepal.width <= 2.75$ & $Petal.length > 2.6$ & $Petal.length <= 4.95$	Versicolor
8	$Sepal.width > 2.75$ & $Petal.length > 2.6$ & $Petal.length <= 4.95$ &$Petal.width <= 1.7$	Versicolor
9	$Sepal.width > 2.75$ & $Petal.length > 2.6$ & $Petal.length <= 4.95$ & $Petal.width > 1.7$	Virginica
10	$Petal.length > 2.6$ & $Petal.length > 4.95$ & $Petal.width <= 1.55$	Versicolor
11	$Petal.length > 2.6$ & $Petal.length > 4.95$ & $Petal.width > 1.55$	Virginica
12	$Petal.length <= 2.45$	Setosa
13	$Petal.length > 2.45$ & $Petal.width <= 1.55$	Versicolor
14	$Sepal.width <= 3.35$ & $Petal.length > 2.45$ & $Petal.width > 1.55$	Virginica
15	$Sepal.width > 3.35$ & $Petal.length > 2.45$ & $Petal.length <= 4.95$ & $Petal.width > 1.55$	Versicolor
16	$Sepal.width > 3.35$ & $Petal.length > 2.45$ & $Petal.length > 4.95$ & $Petal.width > 1.55$	Virginica

Table 12: Rules after pruning using cross-frequency table.

2	$Sepal.length <= 6.65$ & $Petal.width <= 1.75$ & $Petal.width > 0.8$	Versicolor
5	$Petal.width > 1.75$	Virginica
11	$Petal.length > 2.6$ & $Petal.length > 4.95$ & $Petal.width > 1.55$	Virginica
12	$Petal.length <= 2.45$	Setosa
13	$Petal.length > 2.45$ & $Petal.width <= 1.55$	Versicolor
14	$Sepal.width <= 3.35$ & $Petal.length > 2.45$ & $Petal.width > 1.55$	Virginica

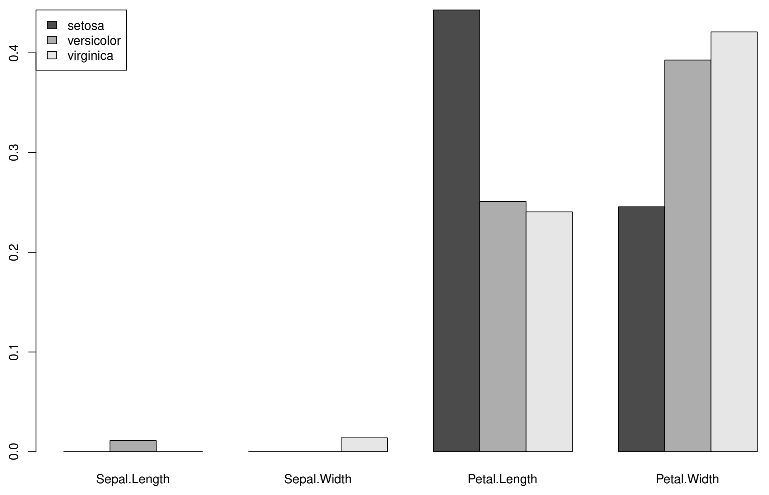


Fig. 6: Medians of Feature Contribution for each class of IRIS dataset

4 Conclusions

In this paper, we showed that Feature Contributions could be used to interpret an ANN model for a before unseen data (instance) to find relationships between instance variables and the predicted outcome. We used ANN models as the example of a non-transparent model which does not allow easy analysis of the network structure and its averaging functions of internal neurons. We built a forest of trees that with high accuracy emulates the behaviour of the ANN model.

Features contributions are numerical values, and their interpretation depends on the problem domain. In the paper, we cited real known problems where Feature Contributions were used for Random Forest models. We also applied Feature Contribution to interpret ANN models via pedagogical approach. To test the certainty of Feature Contribution for the ANN model interpretation, we proposed the procedure of identifying correctly predicted instances. The aggregated results showed that the interpretation of a classification model by the proposed approach is possible once we have good quality data and a highly accurate ANN model. Using a distance measure to representatives of known Feature Contributions for training instances we can decide when to trust the interpretation of ANN model. The representatives in this work were defined by a median or by cluster centres. The averaged results showed that for the best ANN models in 80% of new instances we were able to tell whether the interpretation was certain. The experiment was carried on eight datasets from the UCI Machine Learning repository.

A study on the threshold level for the Euclidean distances used in median and clustering methods and its influence on the ability of ANN interpretation is the next step of our research in this area. Further research, focusing on the distance metrics choice will be an essential enhancement of the study presented here. We also plan to examine a dataset with hundreds of features as in real problems solved by chemists or biologists and conduct further experiments to fully evaluate the proposed method. Investigations of other Feature Contributions analysis methods that can lead to the reduction of false negatives can be interesting research challenges. Another interesting study can be related to the error propagation from ANN to Random Forest models.

5 Acknowledgement

This work was partially supported by the European Commission under the 7th Framework Programme, Coordination and Support Action, Grant Agreement Number 316097, ENGINE - European research centre of Network intelliGence for INnovation Enhancement.

6 Conflict of Interest

The authors declare that they have no conflict of interest.

References

1. Bache, K., Lichman, M.: UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences (2013). URL `http://archive.ics.uci.edu/ml/datasets`. [Accessed: 28 August 2016]
2. Bradley, A.P.: The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recognition **30**(7), 1145 – 1159 (1997). DOI https://doi.org/10.1016/S0031-3203(96)00142-2. URL `http://www.sciencedirect.com/science/article/pii/S0031320396001422`
3. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Statistics/Probability Series. Wadsworth Publishing Company, Belmont, California, U.S.A. (1984)
4. Craven, M., Shavlik, J.: Learning symbolic rules using artificial neural networks. In: Proceedings of tthe 10th International Conference on Machine Learning, pp. 73–80 (1993)
5. Díaz-Uriarte, R., Alvarez de Andrés, S.: Gene selection and classification of microarray data using random forest. BMC Bioinformatics **7**(1), 3 (2006). DOI 10.1186/1471-2105-7-3. URL `https://doi.org/10.1186/1471-2105-7-3`
6. Diederich, J., Saito, K., Nakano, R.: Medical diagnostic expert system based on pdp model. In: Proceedings of IEEE International Conference on Neural Networks, vol. 1, pp. 255–262 (1988)
7. de Fortuny, E., Martens, D.: Active learning-based pedagogical rule extraction. Neural Networks and Learning Systems, IEEE Transactions on **26**(11), 2664–2677 (2015)
8. Fraley, C., Raftery, A.E.: Model-based clustering, discriminant analysis, and density estimation. Journal of the American Statistical Association **97**(458), 611–631 (2002). ABI/INFORM Global
9. Gevrey, M., Dimopoulos, I., Lek, S.: Two-way interaction of input variables in the sensitivity analysis of neural network models. Ecological Modelling **195**(12), 43 – 50 (2006). Selected Papers from the Third Conference of the International Society for Ecological Informatics (ISEI), August 26–30, 2002, Grottaferrata, Rome, Italy
10. Hand, D.J., Till, R.J.: A simple generalisation of the area under the roc curve for multiple class classification problems. Machine Learning **45**(2), 171–186 (2001). DOI 10.1023/A:1010920819831. URL `https://doi.org/10.1023/A:1010920819831`
11. Hardy, A.: An examination of procedures for determining the number of clusters in a data set, pp. 178–185. Springer Berlin Heidelberg, Berlin, Heidelberg (1994)
12. Hayward, R., Ho-Stuart, C., Diederich, J., Pop, E.: Ruleneg: extracting rules from a trained ann by stepwise negation. QUT NRC technical report (1996)
13. Hertz, J., Palmer, R.G., Krogh, A.S.: Introduction to the Theory of Neural Computation, 1st edn. Perseus Publishing (1991)
14. Huysmans, J., Baesens, B., Vanthienen, J.: Using rule extraction to improve the comprehensibility of predictive models,. In: Research 0612, K.U.Leuven (2006)

15. Kamruzzaman, S.M., Islam, M.M.: An algorithm to extract rules from artificial neural networks for medical diagnosis problems. CoRR **abs/1009.4566** (2010)
16. Kirkwood, C., Cave, M., Beamish, D., Grebby, S., Ferreira, A.: A machine learning approach to geochemical mapping **167**, 49 – 61 (2016)
17. Kuz'min, V.E., Polishchuk, P.G., Artemenko, A.G., Andronati, S.A.: Interpretation of QSAR models based on random forest methods. Molecular Informatics **30**(6-7), 593–603 (2011)
18. Marchese Robinson, R.L., Palczewska, A., Palczewski, J., Kidley, N.: Comparison of the predictive performance and interpretability of random forest and linear models on benchmark data sets. Journal of Chemical Information and Modeling **57**(8), 1773–1792 (2017). DOI 10.1021/acs.jcim.6b00753. URL `http://dx.doi.org/10.1021/acs.jcim.6b00753`. PMID: 28715209
19. multiROC: Calculating and visualizing roc and pr curves across multi-class classifications. URL `https://cran.r-project.org/web/packages/multiROC/index.html`. [Accessed: 28 August 2018]
20. Olden, J.D., Joy, M.K., Death, R.G.: An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. Ecological Modelling **178**(34), 389 – 397 (2004)
21. de Oña, J., Garrido, C.: Extracting the contribution of independent variables in neural network models: a new approach to handle instability. Neural Computing and Applications **25**(3), 859–869 (2014)
22. Palczewska, A., Palczewski, J., Marchese Robinson, R., Neagu, D.: Interpreting random forest classification models using a feature contribution method. In: T. Bouabana-Tebibel, S.H. Rubin (eds.) Integration of Reusable Systems, *Advances in Intelligent Systems and Computing*, vol. 263, pp. 193–218. Springer International Publishing (2014)
23. Paliwal, M., Kumar, U.A.: Assessing the contribution of variables in feed forward neural network. Applied Soft Computing **11**(4), 3690 – 3696 (2011)
24. Qin, L.X., Self, S.G.: The clustering of regression models method with applications in gene expression data. Biometrics **62**(2), 526–533 (2006)
25. randomForest: Breiman and cutler's random forests for classification and regression. URL `https://cran.r-project.org/web/packages/randomForest/index.html`. [Accessed: 28 August 2018]
26. rfFC: Random forest feature contrubutions. URL `https://r-forge.r-project.org/R/?group_id=1725`. [Accessed: 28 August 2018]
27. RSNNS: Neural networks in r using the stuttgart neural network simulator (snns). URL `https://cran.r-project.org/web/packages/RSNNS/RSNNS.pdf`. [Accessed: 28 August 2016]
28. Setiono, R., Leow, W.: Fernn: An algorithm for fast extraction of rules from neural networks. Applied Intelligence, **12**(1-2), 15–25 (2000)
29. Sutherland, J., O'Brien, L., Weaver, D.: A comparison of methods for modeling quantitative structure activity relationships. Journal of Medicinal Chemistry **47**(22), 5541–5554 (2004). PMID: 15481990
30. Tickle A.B. Orlowski, M., J. Diederich, J.: Dedec:decision detection by rule extraction from neural networks. In: QUTNRC (1994)
31. Tropsha, A., Gramatica, P., Gombar, V.: The importance of being earnest: Validation is the absolute essential for successful application and interpretation of qspr models. Molecular Informatics **22**(1), 69–77 (2003)
32. Wang, T., Guan, S.U., Ma, J., Liu, F.: Linear feature sensibility for output partitioning in ordered neural incremental attribute learning. In: X. He, X. Gao, Y. Zhang, Z.H. Zhou, Z.Y. Liu, B. Fu, F. Hu, Z. Zhang (eds.) Intelligence Science and Big Data Engineering. Big Data and Machine Learning Techniques, pp. 373–383. Springer International Publishing, Cham (2015)
33. Welling, S.H., Clemmensen, L.K., Buckley, S.T., Hovgaard, L., Brockhoff, P.B., Refsgaard, H.H.: In silico modelling of permeation enhancement potency in caco-2 monolayers based on molecular descriptors and random forest. European Journal of Pharmaceutics and Biopharmaceutics **94**, 152 – 159 (2015)