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

Anna Palczewska[1][0000−0002−6196−9582] and Urszula Markowska-Kaczmar[2][0000−0001−7606−3057]

[1] School of Built Environment, Engineering & Computing, Leeds Beckett University, UK a.palczewska@leedsbeckett.ac.uk
[2] Wroclaw University of Science and Technology,Wroclaw, Poland
urszula.markowska-kaczmar@pwr.edu.pl

Abstract. In this paper, we are focusing on the problem of interpreting Neural Networks on the instance level. The proposed approach uses the Feature Contributions, numerical values that domain experts further interpret to reveal some phenomena about a particular instance or model behaviour. In our method, Feature Contributions are calculated from the Random Forest model trained to mimic the Artificial Neural Network's classification as close as possible. 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 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 in interpretation based on Feature Contributions on average in 80%.

Keywords: Model interpretation· Artificial Neural Network · feature contributions

1 Introduction

Neural Networks (NNs) are widely accepted machine learning technique to learn complex relationships for classification and prediction problems. Their pattern-matching and learning capabilities allowed them to address many difficult problems, impossible to solve by other computational methods. Unfortunately, they lack transparency. It is hard to see how the network arrives at a particular conclusion due to the network architecture's complexity. Therefore, ANN (Artificial Neural Network) is often called a black-box model [12]. The interpretation of the model (why the model makes a particular decision) is important [20], but for non-linear models' extraction of such knowledge is difficult to achieve.

There are two approaches to ANN models interpretation: methods based on rule extraction and variable importance.

Rule extraction methods, that try 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 [5]. The taxonomy of rule

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_12

extraction from neural networks distinguishes the following: decompositional (local methods), pedagogical (global methods) [3] and eclectic methods. 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.

Estimation of variable importance for ANN models explains the relative contribution of each variable to the prediction result. In [14] authors presented the interquartile range (IQR) method to rank variables based on their importance. This method was used to rank variables but does not explain the influence of a variable on predicted value. In [4, 6, 11] methods, based on partial derivatives in ANN sensitivity analysis were proposed to calculate variable importance. In [12] the relative importance of variable, calculated using various methods, was averaged to handle the instability problem of variable importance.

The variable importance is applicable to datasets with a large number of input variables as feature selection method. The variables with the most significant importance are further used to build more accurate models [21]. The need for interpretation and difficulties connected with this problem grow when we consider deep networks. A survey paper [2] and two latest methods [22] used flip points to explain the boundary between two classes and [6] proposed enhanced integrated gradients. Using principal component analysis (PCA) and rank-revealing QR factorization (RR-QR), the set of directions from each training input variable to its closest flip point provides explanations of how a trained neural network processes a dataset.

In some cases, we would like to interpret the model behaviour on the instance level. As an example, let us consider two toxic chemicals (class toxic) with similar structures. We would like to know which part of the structures are the most toxic by extracting contributions of chemical substructures toward the toxicity. 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. In [18] authors presented a method for colouring molecule using a heat map for interpretation of support vector machine models. Another method called Feature Contributions was proposed by Kuzmin et al. in 2011 [8]. It was designed to extract feature contributions for random forest models for regression problems. It has been extended to random forest classification models in [13] and used in work [10], where authors compared the predictions' chemical interpretability based on scoring schemes for assessing heat map images of substructural contributions. Another example of feature contribution was presented in [17] where authors propose the novel explanation technique LIME (Local Interpretable Model-agnostic Explanations) that approximates an interpretable model locally. Also, in [9], authors presented SHAP (SHapley Additive exPlanations) values allowing interpretation of predictive models based on a game theory approach.

Feature contributions are numerical values that allow extraction of a relationship between a particular feature value and a model's decision. For each

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_12

instance, we calculate how much a given variable/feature contributed to the predicted outcome. We can see which features have a positive/negative impact on a predicted value and which of them have a more decisive influence.

There are no methods (that analyse a network structure) to interpret neural networks prediction on an instance level. Currently, in the era of the application of deep neural network in almost all areas with large data availability, the interpretation of feature influence on the model decision would be beneficial for many decision-making models. Many methods for extracting feature/variable contributions are on a model level that 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.

In this paper, we address the problem of interpretation of neural networks on an instance level. To achieve a solution, we propose the use of the neural network as an oracle within a pedagogical approach (similar to rule extraction). This oracle could be any opaque model. Within that approach, we use a Random Forest (RF) model together with its Feature Contribution (FC) method described in [13]. In the presented research, we assume that the FCs are acquired from RF mimicking 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 the ANN responds with the same class as FCs for a new input vector, then this is an indication that we can trust the interpretation delivered by FCs.

The 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

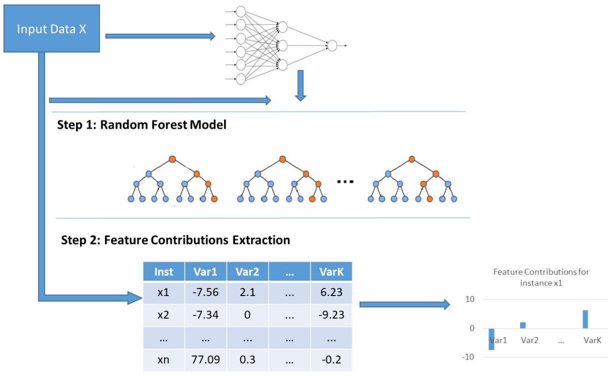
Although extraction of feature contributions is not new, as we are borrowing from existing methods, feature contributions in the context of ANN model interpretation require the development of some methodology. In this section, we recall the definition for feature contributions, and we describe how the feature contributions can be used to interpret neural networks.

2.1 ANN Model Interpretation for a Single Instance

We assume that the ANN model trained for a specific classification problem is given. Our idea is to train the Random Forest model to mimic the behaviour of ANN then to calculate Feature Contributions.

The workflow of the ANN model interpretation is presented in Figure 1. In step 1 we build Random Forest (RF) model using input data $\mathbf{x}$ and output $\mathbf{y} \in Y_{RF}$ produced by the ANN model. Thus, the training data set for RF is composed of pairs $<\mathbf{x}, \mathbf{y}>$, where $\mathbf{x} \in D_{RF}$ and $\mathbf{y} \in Y_{RF}$. In step 2, we extract Feature Contributions from a Random Forest model. When, for a new input

DOI: 10.1007/978-3-030-77964-1_12


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

instance vector $\mathbf{x}_{new}$ we want to interpret the ANN classification result, we calculate Feature Contributions (FCs) for this instance. They show the influence (negative or positive) of each feature (input variable) on a predicted class. To assess whether we can trust the result we perform classification based on FCs, and the evaluation is positive if the class predicted by ANN and by FCs are the same.

2.2 Random Forest Feature Contributions

Firstly, we recall the definition of the feature contributions proposed in [8, 13]. 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:

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

where Y_{mean} is a fraction of the training instances in a given node c, where c - is a child node and p - is a parent node, belonging to a selected class (for details see [13]) 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 summed on tree paths:

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

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 $\mathbf{x}_i$ in $D_{New}$ do
 5:     calculate feature contribution $FC_i$ for an instance $\mathbf{x}_i$)
 6:     for each class $c$ in datasets classes $C$ do
 7:         Calculate Euclidean distance between feature contributions $FC_i$ for the instance $\mathbf{x}_i$ 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 $\mathbf{x}_i$ then
11:         $p_i = 1$
12:     else
13:         $p_i = 0$
14:     end if
15: end for
```

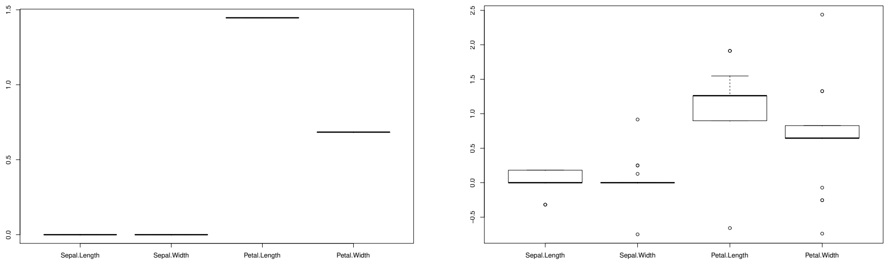
where the value $LI_{i_{f_{kl}}}$ is a local increment for the instance i, feature f in k tree and its l node. The values *ntree* and *knode* represent the number of trees in the forest and the number of nodes from the k tree, which split over a feature f, respectively.

Feature Contribution values estimate a contribution of feature values to the difference between the actual prediction and the mean prediction for the current set of feature values. As reported in [13], $Y' = Y^r + \sum_j FC_j$ where Y' denotes a predicted value and Y^r averages of Y_{mean} overall root nodes in the forest with the assumption of unanimity (all elements in trees nodes belonging to the same class). The magnitude represents how strongly the feature contributes and the sign represents the direction (such as toward the model decision or against).

2.3 Interpretation of ANN Prediction Based on Feature Contributions

Once feature contributions are extracted from a Random Forest model they can be interpreted by domain expert reviling the model decision process. As these values were calculated within the pedagogical approach we need to assess the certainty of such interpretation. This procedure is shown in Algorithm 1. As the input, the algorithm requires trained ANN, datasets D_{RF}, Y_{RF} for RF training.

To test if we can trust the interpretation of ANN prediction for a new instance $\mathbf{x}_{new}$ we use a distance between feature contributions of the new data and feature contributions *representatives* for the Random Forest training dataset (line 2-3 in Algorithm 1). As described in [13] we can consider two feature contri-



(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.

bution representatives: median and cluster centroids, computed for each class separately. To calculate representatives, we used all instances from the random forest training dataset that were correctly classified. 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.
  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 (eq.2) is computed for all class representative's medians (line 5-9 in Algorithm 1), 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 eq. (1), $nvar$ is a number of features (variables) in the input vector and m_{fl} is a feature contributions median 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).
  Many 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 feature contribution that differs from the majority group created. The group with the smallest variance is called a core cluster [13] and its centre is used as the class *representative*. If clusters have the same variance (e.g equal to zero) we can have more than one representative for a class. For each class, the best number of clusters is obtained using the elbow method [15].
  Training instances $\mathbf{x} \in D_{RF}$ are assigned to these clusters. To classify a new instance $\mathbf{x}^i_{new}$, the Euclidean distance (eq. 3) is calculated to all cluster

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	5.48	-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

centres and the smallest distance is selected (line 5-9):

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

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 $\mathbf{x}_{new}^i$ belongs to and defines a class for the new instance.

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

If the class assessed by the use of feature contributions 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 possible ($p_i = 0$) (lines 10-14 in Algorithm 1).

3 Experimental Study

The experimental research goal is to test whether the feature contributions method can be used to interpret a trained ANN model. In this research, we focused on the shallow ANN, but it could also be a more complex model. The process of training ANN for a given training dataset, developing the Random Forest model, extracting Feature Contributions, identifying the model FCs representatives, and testing ANN model reliability using FCs was repeated fifty times. The averaged results are presented in this section.

3.1 Datasets

Eight datasets from the UCL Machine Learning Dataset Repository [1] were used. We selected the datasets that were often used as benchmark sets in rule extractions for ANN models [7]: Breast Cancer Wisconsin (Diagnostic) Dataset (BCWD), COX2 [19], German Credit Scoring, IRIS, SEEDS, Teaching Assistant Evaluation (TEACHING), WAVEFORM, Database Generator (Version 1), WINE.

3.2 Training the Artificial Neural Network Model

The multi-layer perceptron (MLP) network has been used as ANN model. Training is performed by the backpropagation method. We used the default settings for the *MLP* model from the RSNNS package in R. We only set a parameter *size* (describing the number of hidden neurons) to be equal to the averaged sum of input and output variables, learning coefficient - *learnFuncParams* equals 0.1 and the maximal number of iteration equals 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* models for training and testing datasets obtained from the repeated procedure of 50 runs, each time splitting the dataset and generating a new ANN model.

In the training *MLP* procedure, we do not focus on high-quality results, therefore one can see that the *MLP* model gives for some datasets (BCWD, IRIS, SEEDS, WAVEFORM and WINE) high averaged accuracy around 0.9, but for some (TEACHING and COX2 dataset) they are less satisfying.

3.3 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 2.1. We used *randomForest* 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

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_12

Table 2: Characteristics of datasets and average accuracy (ACC) of ANN over 50 runs of the 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), 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 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), AUC for ANN and RF models for D_{new}

Name	$\#D_{new}$	ANN_{new}	$AUC_{ANN_{new}}$	RF_{train}	RF_{new}	$AUC_{RF_{new}}$
BCWD	137	0.97	0.96	0.99	0.98	0.97
COX2	38	0.68	0.67	0.95	0.77	0.74
German_CS	200	0.74	0.66	0.96	0.81	0.72
IRIS	30	0.95	0.94	0.99	0.97	0.96
SEEDS	42	0.93	0.84	0.99	0.92	0.90
TEACHING	31	0.64	0.53	0.98	0.93	0.90
WAVEFORM	1002	0.86	0.84	0.97	0.84	0.83
WINE	37	0.98	0.98	0.99	0.94	0.94

replacement for training trees. We also keep information on records that were used to train a tree in a forest by setting the parameter *keep.inbag*=True. This is needed to calculate Feature Contributions.

Table 3 shows the averaged results for Random Forest models and for the *MLP* models. 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 models. The table also shows the average accuracy of the Random Forest models for training data (column RF_{train}) achieved on the D_{RF} 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 3 presents averaged AUCs for D_{new}. The higher the 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.

3.4 Certainty Assessment of ANN Interpretation

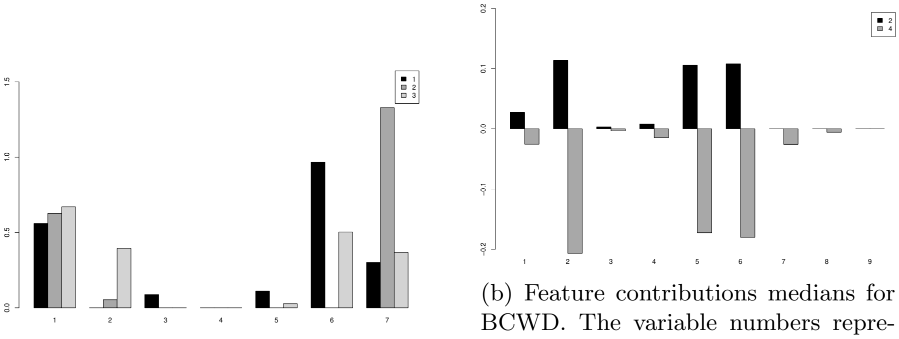
Feature Contributions calculated for the instance $\mathbf{x}_{new}$ give information about the relation between predicted class and input features for an RF model. Because the RF model only mimics the ANN model we are interested in evaluating how much we can rely on this interpretation. To decide whether 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 [16]. 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. Finally, we extract the Feature Contributions representatives for each class. In Figure 3 we present medians representatives of Feature Contributions for two datasets and all classes (for each dataset). Contributions can be positive as well as negative values and representatives differ between classes.

Having the Feature Contributions representatives for each class, we calculate Feature Contributions for each $\mathbf{x}_{new}$ instance from D_{new}. Then, to find the class for the new instance, we compute distances between representatives and Feature Contributions for instances from D_{new} using eq. (2) and (3). The smallest distance assigns the class.

Interpretability Method Evaluation for all Datasets In this section, we repeat the procedure described in Algorithm 1 for all eight chosen datasets. Table 4 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

DOI: 10.1007/978-3-030-77964-1_12



(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 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.

Fig. 3: Example of Feature Contributions median for selected datasets

as certain based on the clustering approach (Clust_Certain) and the number of correctly classified instances for the original class (Clust_Correct). Also, Table 5 presents detailed results from the certainty assessment of the interpretability method. For each dataset, columns represent 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 instances that were classified wrongly by the ANN model.

The aggregated results confirm that the presented method is suitable to interpret the 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 represent the true importance of the ANN model. For weak models (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. This shows 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.

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 shallow ANN models as the example of a non-transparent model. This approach offers interpretation

DOI: 10.1007/978-3-030-77964-1_12

Table 4: 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

for any opaque model and does not limit its architecture. The idea of method interpretation lies in building a forest of trees that with high accuracy emulates the behaviour of the opaque model and then Features Contributions calculation allow us interpretation on an instance level.

To test the certainty of Feature Contribution for the ANN model interpretation, we proposed the procedure for the classification of instances based on their feature contribution values. Using a distance measure between a new instance feature contribution and the model representatives Feature Contributions we can decide wherever to trust the interpretation of the ANN model. The representatives in this work were defined by a median or by cluster centres defined on the model training dataset. 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. Comparison of the proposed method with other available methods to test the agreement on the explained model decision will be the next interesting research problem to address.

References

1. Bache, K., Lichman, M.: UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences (2013), http://archive.ics.uci.edu/ml/datasets, [Accessed: 28 August 2016]
2. Feng-Lei Fan, Jinjun Xiong, M.L., Wang, G.: On interpretability of artificial neural networks: a survey (2020), https://arxiv.org/ftp/arxiv/papers/2001/2001.02522.pdf

Table 5: 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
		incorrect	3	5	3	3
COX2	38	correct	21	3	23	3
		incorrect	7	7	6	6
German_CS	200	correct	127	32	145	19
		incorrect	34	7	35	1
IRIS	30	correct	26	2	27	2
		incorrect	1	1	1	0
SEEDS	42	correct	33	4	37	2
		incorrect	2	3	2	1
TEACHING	31	correct	19	1	22	0
		incorrect	3	8	5	4
WAVEFORM	1002	correct	585	107	663	98
		incorrect	89	221	82	159
WINE	37	correct	29	4	33	2
		incorrect	2	2	1	2

3. de Fortuny, E., Martens, D.: Active learning-based pedagogical rule extraction. Neural Networks and Learning Systems, IEEE Transactions on **26**(11), 2664–2677 (Nov 2015)
4. Gevrey, M., Dimopoulos, I., Lek, S.: Two-way interaction of input variables in the sensitivity analysis of neural network models. Ecological Modelling **195**(1–2), 43 – 50 (2006), selected Papers from the Third Conference of the International Society for Ecological Informatics (ISEI), August 26–30, 2002, Grottaferrata, Rome, Italy
5. Huysmans, J., Baesens, B., Vanthienen, J.: Using rule extraction to improve the comprehensibility of predictive models,. In: Research 0612, K.U.Leuven (2006)
6. Jha, A., Singh, J.K.A.M.R.G.D., Barash, Y.: Enhanced integrated gradients: improving interpretability of deep learning models using splicing codes as a case study. Genome Biol **149**(21) (2020)
7. Kamruzzaman, S.M., Islam, M.M.: An algorithm to extract rules from artificial neural networks for medical diagnosis problems. CoRR **abs/1009.4566** (2010)
8. 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)
9. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), `https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf`
10. 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 Mod-

eling **57**(8), 1773–1792 (2017). https://doi.org/10.1021/acs.jcim.6b00753, `http://dx.doi.org/10.1021/acs.jcim.6b00753`, pMID: 28715209

11. 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**(3–4), 389 – 397 (2004)
12. 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)
13. Palczewska, A., Palczewski, J., Marchese Robinson, R., Neagu, D.: Interpreting random forest classification models using a feature contribution method. In: Bouabana-Tebibel, T., Rubin, S.H. (eds.) Integration of Reusable Systems, Advances in Intelligent Systems and Computing, vol. 263, pp. 193–218. Springer International Publishing (2014)
14. Paliwal, M., Kumar, U.A.: Assessing the contribution of variables in feed forward neural network. Applied Soft Computing **11**(4), 3690 – 3696 (2011)
15. Qin, L.X., Self, S.G.: The clustering of regression models method with applications in gene expression data. Biometrics **62**(2), 526–533 (2006)
16. rfFC: Random forest feature Contrubutions., `https://r-forge.r-project.org/R/?group_id=1725`, [Accessed: 28 August 2016]
17. Ribeiro, M.T., Singh, S., Guestrin, C.: ”why should i trust you?” explaining the predictions of any classifier. In: KDD 2016 San Francisco, CA, USA (2016)
18. Rosenbaum, L., Hinselmann, G., Jahn, A., Zell, A.: Interpreting linear support vector machine models with heat map molecule coloring. Journal of Cheminformatics **3**(1), 1–12 (2011)
19. 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
20. 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 (4 2003)
21. Wang, T., Guan, S.U., Ma, J., Liu, F.: Linear feature sensibility for output partitioning in ordered neural incremental attribute learning. In: He, X., Gao, X., Zhang, Y., Zhou, Z.H., Liu, Z.Y., Fu, B., Hu, F., Zhang, Z. (eds.) Intelligence Science and Big Data Engineering. Big Data and Machine Learning Techniques. pp. 373–383. Springer International Publishing, Cham (2015)
22. Yousefzadeh, R., O'Leary, D.P.: Interpreting neural networks using flip points (2019), `https://ArXiv,abs/1903.08789`