Comparing Explanations between Random Forests and Artificial Neural Networks

Lee Harris[1] and Marek Grzes[1]

Abstract— **The decisions made by machines are increasingly comparable in predictive performance to those made by humans, but these decision making processes are often concealed as black boxes. Additional techniques are required to extract understanding, and one such category are *explanation methods*. This research compares the explanations of two popular forms of artificial intelligence; neural networks and random forests. Researchers in either field often have divided opinions on *transparency*, and comparing explanations may discover similar ground truths between models. Similarity can help to encourage trust in predictive accuracy alongside transparent structure and unite the respective research fields. This research explores a variety of simulated and real-world datasets that ensure fair applicability to both learning algorithms. A new heuristic explanation method that extends an existing technique is introduced, and our results show that this is somewhat similar to the other methods examined whilst also offering an alternative perspective towards least-important features.**

I. Introduction

Machine Learning (ML) is ubiquitous, and the presence of perceivably intelligent machines is commonplace in much of society [1], [2]. We delegate large amounts of responsibilities to machines, and understanding why and how decisions are made is crucial to mitigating problem behaviour.

ML models (e.g. decision trees, neural networks) are used to provide simple interfaces to highly-complex problems [3] (e.g. abstraction of the very complex mammal brain through graphs). Clear understanding of a model contributes towards this goal, but this human-centric design is often a secondary objective behind predictive performance. Understanding can improve efficiency, error correction and accuracy, along with enabling trust [4], [5]. It is unlikely that a patient preparing for serious life-critical surgery would consent to an abstruse operation, and workplace standards for human are often incredibly comprehensive; is it then permissible to excuse justification of machine actions? Allowing machines to make decisions about humans is a contentious area, and there is much work on identifying and resolving machine discrimination and fairness bugs [2], [6].

Some applications of ML may be content with unexplained decisions, maybe as a solution or algorithm is difficult to describe [4], and a leading retort against *transparency* is that it reveals "trade secrets" (e.g. a business model). However, the need for understandable and contestable decision making is now a legal requirement (GDPR) for all businesses that interact with humans [7]. Safety and security are paramount to accountable, fair, and ethical ML research [8].

There is clearly a need to understand machine decisions, yet, it is often perceived [5] that understandable representations are structure dependant. ML algorithms are typically categorised as either White Box (WB) or Black Box (BB), and this has previously determined whether a learning algorithm can be understood. WB algorithms produce decisions according to transparent structure, examples are classification rules [9], shallow decision trees [4], and linear models [5]. These can be decomposed into disjunctive predicates. Alternatively, BBs are complex structures with many parameters and settings that are often incomprehensible to humans. A popular algorithm in this category is the neural network, and variants of these have achieved record accuracy on complex ML tasks; such as object recognition [10] and natural language processing [11].

A. *Our Work*

A common interpretation of the 'best' algorithm to make decisions about humans often resolves to whichever can be contested and defended [7], but this may not be the most predictive (e.g. Random Forest (RF) application to granting loans [12] when Neural Networks (NNs) may have better predictive accuracy [13]). This work explores similarity between explanations of RFs and NNs, and if these are equivalent, perhaps predictive accuracy can become a larger contributor when choosing which algorithm to apply to new data. Formally, the learning algorithms that we explore are examples of BBs, but it can be argued that RFs are "grey box" models. Individual decision trees often have defined structure, but it is unclear which models are prioritised when collectively aggregating decisions.

As highlighted by [5], the definition of 'interpretability' is rarely explicit. From their research, we interpret this as "*useful information of any kind.*". This recognises that ML can produce more than just predictive accuracy (e.g. causal associations), and that other information or metrics may be the most important. This definition can be split into two distinct components: interpretation and explanation. Informally, the interpretation of a model collectively assesses the structure and parameters to determine which features are important across all data (general knowledge of the domain), while explanations identify how the model responds to a particular input. Interpretation and explanation may accompany each other, but importance scores between these are unlikely to correlate in all situations. Formal definitions of these terms are given in [14].

This paper focuses on explainability, but this is not well defined in discourse, and different areas of research refer

[1]The School of Computing, The University of Kent, Canterbury, Kent, United Kingdom

to explanability differently (attribution techniques for NNs [14] vs case-wise iterpretations [15] for RFs). We generalise *explanation method* to mean any process used to generate an explanation, where an explanation assigns quantitative importance scores (a.k.a *relevance*) to each feature of a particular data instance.

Consider an object detection task in computer vision as an example of why explanations are important. Interpretation would identify which pixels are most active across all images, or those most active for a particular class, but explanation will identify which pixels contribute most for a specific input. If a prediction is incorrect, explanation can identify (provide *evidence*) which pixels contributed most to the outputs, and a human can visually interpret this to uncover model or input flaws (see Fig. 1 for explanation examples). Such insights help with debugging and corrections [16].

To our knowledge, no other research has evaluated the explanations of RFs and NNs to conclude whether the same features are identified as important by both. Our research is not an exhaustive comparison of all possible model variations and parameters, but we explore several existing explainability approaches, and contribute a heuristic extension of an existing algorithm.

In summary, the main contributions of this paper are:

- The first paper to compare explanations between neural networks and random forests
- Replication and extension of findings and methodology from previous work
- A new heuristic extension of an existing method
- Advice on which algorithms and parameters are most explanatory for domains with particular characteristics

II. Algorithms

A. Neural Networks

These are implemented through a graph structure containing layers of units, where the internal representation of inputs is transformed at later depths. Each unit in a network is labelled with the product of activations and weights in the proceeding layer applied to a non-linear activation function, and there are many different model variations. The shallow neural networks used in this research consist of fully-connected layers of units.

Sections II-A.1 & II-A.2 define the explanation methods which we use in this paper to explain decisions of NNs.

1) Sensitivity Analysis: Sensitivity evaluates the impact of each input on network output. Explanations are unbounded, and relevance scores may be unproportionate and scattered, however, Sensitivity Analysis (SA) has successfully explained several problems [14]. In this paper we define the sensitivity of an input feature X_i as the absolute summation of each network *output* partially differentiated with respect to X_i.

Our implementation of SA is:

$$S_i(X) = \sqrt{\sum_{k=1}^{|o|} \left(\frac{\partial o_k}{\partial X_i}\right)^2}, \tag{1}$$

where $S_i(X)$ is the sensitivity score, and o_k is the set of network outputs.

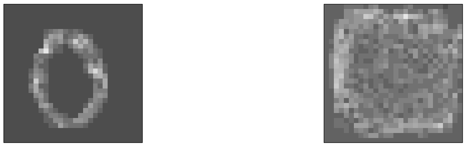


Fig. 1: Results of our LRP (left) and SA (right) implementations over a '0' from the MNIST [18] dataset.

2) Layerwise Relevance Propagation: Given an existing network and a data example X, Layerwise Relevance Propagation (LRP) [17] discovers the contribution of each input feature X_i by backpropagating output activations. The algorithm was originally applied to object recognision to identify which pixels in an image are being "looked at" by the network, but research in the area supports application to a range of different network variations and problems [14].

Our research uses a consistent [10] parameter variant of LRP called the *z-rule* that only propagates through connections with positive weights. The algorithm is implemented according to [14] and can be expressed as:

$$R_j = \sum_k \frac{a_j \max(0, w_{jk})}{\sum_{j'} a_{j'} \max(0, w_{j'k})} R_k, \tag{2}$$

where R is the relevance score, j is the current unit, k is a unit in the next layer (towards output), a is the activation and w is the weight between two units. Given a mixed-value vector, the max function specifies that every negative value becomes 0.

3) Implementation: Figure 1 depicts our implementations of LRP and SA over the MNIST [18] dataset. This is an object recognition task, where an NN must identify handwritten digits between 0 and 9. The underlying NN that explains these images uses similar hyperparameters to those in S3 of Sec. III-C (a single hidden layer, 40 HU, 100 epochs, 32 batch size, no biases, ReLu and Softmax activation functions). Here we explain the input image, *0*. LRP very clearly explains this, while the explanation of SA is very scattered. Our results are consistent with those of other research [14].

B. Random Forests

Decision Trees (DTs) split on important features through recursive-partitioning that aims to maximise the class-purity of each partition. Predictive performance of these can be improved through aggregation of several DTs into a forest ensemble, but this masks the transparent structure. This paper focuses on the Random Forest (RF) [19] and Conditional Inference Forest (CF) [20] algorithms. The term *forest* is generalised in this paper to mean either RF or CF depending on context, and we focus on ML classification.

Forests classify data through majority voting, where the majority contains all DTs that classify a data example according to the most frequent prediction of all trees. There are various DT hyperparameters that control tree structure, but the key difference between implementations is how subtrees are created. This is important for our experiments as this can dictate which features appear in a DT. The main hyperparameters for forests are *ntree* (the number of ensemble members) and *mtry* (the number of random features to trial over at each DT) as these effect individual DT structure and feature frequency.

Research by [21] has shown that interpretability of forests may fluctuate relative to the underlying ensemble members, as some DTs (e.g. CART [19]) favour splitting on numerical or nominal features with many categories. Their research goes on to show that forests should be unbiased when the objective is interpretability. Our research explores RFs and CFs to test both biased and unbiased splitting.

The different types of forests can be explained through the explanation methods below in Sec. II-B.1 & II-B.2.

1) Intervention in Prediction Measure: This algorithm is similar to the simple selection frequency method described by [21], but only certain features are counted, opposed to all features across all trees. The Intervention in Prediction Measure (IPM) [15] records the likelihood of each feature appearing as a splitter in the applicable path of each ensemble tree, and this frequency is averaged over all trees. The explanation score assigned to each feature is the average path-likelihood across the entire forest. IPM abbreviates explanation over RFs, while explanations over CFs are referred to as CIPM throughout this paper.

2) Adjusted IPM: The method that we propose in this paper extends IPM by ensuring that paths which contribute to a prediction (those in the winning majority) decide explanation. This algorithm is partially class discriminative [22] as feature likelihood alters in response to prediction, but majority and minority DTs may have similar structure. The current paper abbreviates AIPM as application of Adjusted IPM over RFs, and ACIPM as application to CFs.

III. Methodology

In this section, we give details about our experiment methodology, describing what and how we tested.

A. Tools

The Conditional (CF) and Random (RF) Forests are respectively implemented in R by the *cforest* [20] and *randomForest* [19] packages, and these are available on the CRAN. Support for NNs and manipulation of these uses the high-level Keras[1] library and low-level Tensorflow[2] library.

The similarity between explanation methods is measured using Spearman's ρ and Kendall's τ correlation metrics.

The original GlaucomaMVF dataset is obtainable from the *ipred* [23] package in R, all other datasets can be obtained from the UCI[3] repository, and these are referenced in Tab. III.

B. Method

We evaluate the explanation methods introduced in Sec. II over simulated and real data. Simulated data is explained in Sec. III-D.

By evaluating explanations, we are assessing:

- Whether competing models empirically produce similar explanations
- Whether features correlate between explanations
- The confidence of a model, and the resulting explanation preciseness
- Under what conditions correlation between models is weak
- Where applicable, which explanation method correctly captures the true explanation of features

For each dataset, we begin our comparisons by randomly sampling 100 unique (without replacement) data instances. Each explanation method independently generates explanations for each of these using an adaptation of leave-one-out cross validation; where one instance is selected to be explained, and all others in the dataset are used to construct a model. The explanations generated by this process have differing confidence values, so we rank each feature in each explanation to ensure fair comparison.

Average feature ranks are produced from the 100 explanations, and this is recorded for each explanation method. This metric is most applicable when we know the important features, as with the simulation data, and this averaging is used in [15]. Real data may not have consistent important features, so we also generate correlation heatmaps. Each correlation score is again averaged over the 100 explanations.

C. Datasets

Multi-layered NNs can learn complex representations, and this is why they perform well in domains where input features are *un-informative* and do not reveal information about the final prediction. On the contrary, SIFT features [24] aggregate individual pixels in computer vision into *informative* regions or shapes. RFs are usually applied to informative features (e.g. presence of a disease) and these are better correlated (prehaps unintentionally [25]) with the prediction. Prior explainability work focuses on informative features [15], [21], and simulated data can embed high-level associations between features. We thus believe that comparing informative features is not a limiting factor.

Features values are numeric, and scaled between 0 and 1. This is for fair applicability to NNs, RFs and each explanation measure, and this scaling is common in live deployment [26]. We also remove instances with missing values from the real datasets. These are detailed in Tab. III.

D. Simulated Data

Simulated data can assess how the explanation methods respond to different dimensionality and structure, while similar papers exploring the techniques used here also experimented over simulated data [15], [21], [27]. Previous experiments

[1] https://keras.rstudio.com
[2] https://tensorflow.rstudio.com
[3] https://archive.ics.uci.edu/ml/datasets.html

Scenario	Instances	Features
S1	300	6
S2	3000	12
S3	1500	30

TABLE I: The dimensionality of each scenario.

	RF		CF		NN	
Scenario	ntree	mtry	ntree	mtry	HU	Epochs
S1	500	4	300	4	4	750
S2	450	8	250	6	9	700
S3	400	13	200	9	25	650

TABLE II: The non-default hyperparameters across experiments. *ntree* is the amount of trees in the ensemble, *mtry* is the number of random features to try in each tree and *HU* is the number of hidden units.

have only evaluated nominal features, and testing over numerical values increases the applicability and validity of these explanation methods [27].

Knowing and being able to manipulate class labels and feature values of simulated data ensure that we know the target result of each explanation. We explore three sizes of simulation scenario (Tab. I), and each of these contains four different datasets (problems).

Each dataset being explored is as follows:

Baseline: every feature value and class value is uniformly sampled between 0 and 1, therefore explanatory features do not (intentionally) exist.

One important feature: as in [15], the class value is decided by a single feature (feature 1) and explanations should highlight this. If the first feature value is greater than 0.5, the class becomes 0, otherwise it becomes 1.

Co-Importance: these datasets test explanations over co-importance. If the combined sum of features 1 and 2 is less than or equal to 1, or more than 1.5, the class of an instance is labelled as 0, otherwise it becomes 1.

Feature importance relative to feature index: with a set of features f, the feature at index i exerts $\frac{i}{|f|}$ influence. The class assigned to an instance X is 1 if the weighted sum of each feature is greater than 0.5, otherwise it is 0. This is expressed as:

$$C(X) = \begin{cases} 1, & \text{if} \sum_{i=1}^{|f|} \left(\frac{i}{|f|} X_i\right) > 0.5 \\ 0, & \text{otherwise} \end{cases}, \tag{3}$$

where $C(X)$ is the class of X, and X_i is a feature value. Feature importances increase progressively, and this should be visible in the explanation results.

Some properties of the datasets are inhereited from other research, notably [15], [21], [27]. The values of ignored features are sampled from a uniform distribution between 0 and 1, and the class label of each instance may flip according to a noise probability. These values are 0.1, 0.2 and 0.3. Noise allows us to compare simulated explanation correctness and correlation in response to predictive accuracy.

Non-default hyperparameters are listed in Tab. II.

Dataset	Instances	Features	Classes	MV	Ref
GlaucomaMVF	170	66	2	17	[23]
Abalone	4177	8	29	0	[29]
Diabetic Retinopathy	1151	20	2	0	[30]
White Wine Quality	4898	12	7	0	[26]
Website Phishing	1353	10	3	0	[31]

TABLE III: Statistical summary of the real datasets.

E. Model Structure

This research uses NNs with a single hidden layer. This is because we focus on data with *informative* features, and to avoid gradient problems (shattering, vanishing or exploding [14]) that would corrupt the findings of sensitivity analysis. Our decision is further reinforced by [28] who states that any network can be represented by a single hidden layer containing many hidden units.

Unit biases are not present in the NN. The LRP algorithm does not specify behaviour for these, and these are not included in similar research using this technique [14], [22]. Empirical evaluations, which are omitted for brevity, showed that neither inclusion of biases nor additional hidden layers significantly altered accuracy on our domains. Other NN implementation details include that the ReLu activation function is applied to each Hidden Unit (HU), the softmax function is applied to each output unit, the batch size of each update is 32, and RMSProp is used as the loss function. The default number of HUs for the real data are $\frac{2}{3}$ of the number of features.

As mentioned in Sec. II-B, we use bootstrap sampling without replacement when constructing RFs to avoid bias. This method of sampling has been shown to be a fairer assessment of feature importance if features have a varying number of nominal categories or split on numerical attributes [21]. By incorporating this in both tree ensembles we can reduce bias that would skew comparisons.

IV. Results

We now explore several simulation and real-world datasets. The x-axis displays each explanation measure, and the y-axis represents either Average feature Rank (AR) or noise. The AR is scaled between 0 and 1 in order to highlight importance, and darker bar shades indicate higher ranks.

A. Simulated data

Simulated data is paramount in transparency research because the ground truth is known [21], we therefore present results for the simulated datasets defined in Sec. III-D first.

1) Dataset 1—Baseline: Figure 2 depicts each explanation method on scenarios of different size and noise. We would expect to see the top of the bars aligned if all features contributed equally to decisions, but high variance in bar height across all forest results indicates that these algorithms incorrectly favour some features. On the contrary, we can see that LRP and SA are clearly optimal in scenarios 1 & 2 across all noise levels since their bars have similar height.

As we can clearly see in Fig. 2-S1, the original RF explanation methods (IPM and CIPM) incorrectly identify

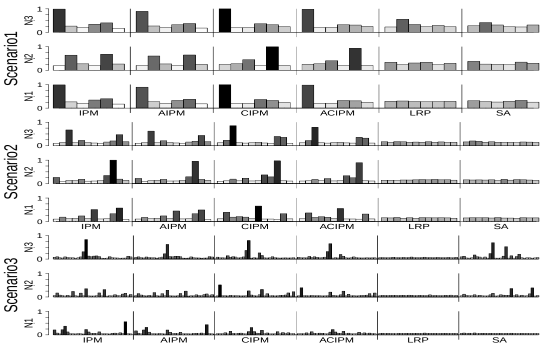


Fig. 2: Explanation results of dataset 1 over the three simulated scenarios (external y-axis). Each scenario contains results of each noise level (internal y-axis). The x-axis for each graph displays the 6 explanation methods, and each sub-sub-graph displays the importance of each feature according to the explanation measure, the given scenario and the given noise. Bar height represents importance, and darker bars are the most important.

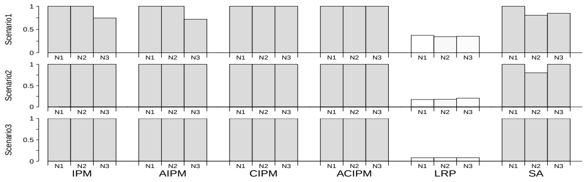


Fig. 3: Feature Rankings for Simulation Dataset 2.

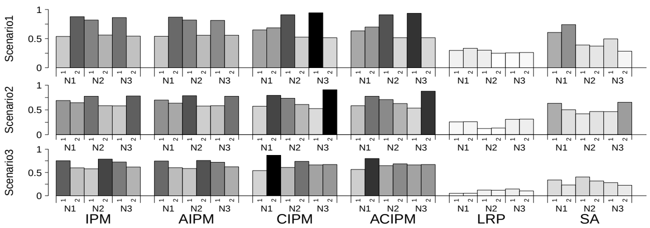


Fig. 4: Feature Rankings for Simulation Dataset 3.

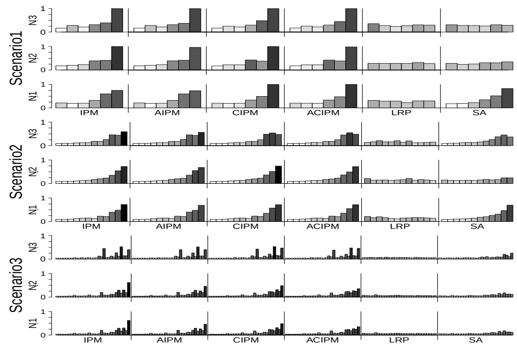


Fig. 5: Feature Rankings for Simulation Dataset 4.

specific features as the most important. One intuition for this behaviour and the fact that noisy features are favoured is that RFs must always have a root node. A slightly important feature (which is only slightly more informative) is attributed a disproportionate amount of importance; especially if the DT is a decision stump (one splitter). LRP can distribute the rank uniformly across all irrelevant features, whereas DT must prioritise a feature(s) as important in order to create nodes and partition data. Due to their nature, RFs are somewhat "pushed" to make some features important, whereas LRP can remain uncertain and does not have to commit to strong opinions. This is an interesting finding as it was not captured in previous research [15].

2) Dataset 2—One Important Feature: Figure 3 shows the AR of feature 1 over the six explanation methods. There are three bars per method, and the smallest amount of noise is shown at the left-most bar of each explanation result. The bar reaches 1 when the feature is the most important.

We can see that the forest explanation methods assign feature 1 the most importance, in contrast, LRP consistently struggles over all three scenarios and it was not able to confidently identify the importance of feature 1. This result can be partially explained using arguments in Sec. IV-A.1 where we state that DTs must always commit to a feature(s), whereas LRP has flexibility to remain neutral. The fact that DTs must have a root node is clearly useful in this problem.

Sensitivity analysis identifies the feature perfectly in scenario 3, and almost as assuringly in scenarios 1 and 2. More features and data instances appear to increase confidence. The ability for it to confidently choose the best feature diminishes as the feature-to-instance ratio decreases, but it is otherwise capable of identifying the same features as the forest explanations. For the reasons mentioned above, forests also appear to be the most robust against moderate levels of noise.

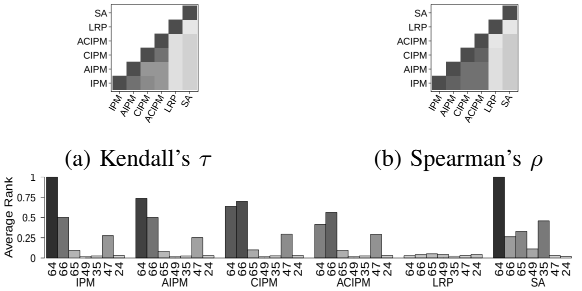
3) Dataset 3—Co-Importance: Figure 4 depicts the importance given to features 1 and 2 across all simulated scenarios. All methods correctly determine that features 1 and 2 are the most relevant. We control the noise level for each method, and this leads to 6 bars and a lengthened graph. LRP is the least confident, and it only just narrowly explains correctly. LRP is uncertain which features are the best, and this is a trend throughout the simulation results. We explore possible resolutions to this in Sec. IV-C. Forest methods are the most confident, potentially because they implement a kind of internal and hard feature selection, however, the fully-connected NNs consider all features, and as long as classification quality (e.g. loss and predictive accuracy) is satisfactory, these are not encouraged to perform extreme alterations to weights and potentially regress.

4) Dataset 4—Relative Importance: This simulation dataset assigns features progressive importance. The results in Fig. 5 again show the average rank of each feature across all scenarios and noise.

One can see that forest methods identify the correct ranks across all experimental conditions; as illustrated by the "triangular" shape favoured by the bars. Some inconsistencies can be observed in the largest scenario (bottom), and we believe that this is due to a large number of features and a small 'mtry' parameter. A DT will split on the feature that best partitions instances, but consistently omitting the

Dataset	RF Acc	CF Acc	NN Acc
GlaucomaMVF	91.4 ± 0.007	89.7 ± 0.008	85.1 ± 0.017
Abalone	54.8 ± 0.003	56.0 ± 0.004	56.0 ± 0.003
Diabetic Retinopathy	68.1 ± 0.006	69.9 ± 0.007	72.6 ± 0.014
White Wine Quality	70.9 ± 0.003	62.4 ± 0.002	58.8 ± 0.005
Website Phishing	89.2 ± 0.002	89.0 ± 0.002	87.2 ± 0.005

TABLE IV: Testing accuracy across the test sets of each real dataset. The ± represents the standard deviation over the 100 models.



(a) Kendall's τ

(b) Spearman's ρ

(c) Feature Rankings

Fig. 6: Feature Rankings for the Glaucoma Dataset.

most important features can increase the frequency of less important features. This skews explanations that use feature frequency, and so 'mtry' may bias these explanations.

LRP finds this task particularly challenging. The "triangle" of importance is partially present when there is less noise and the feature-to-instance ratio is low, but this quickly starts to flatten. We believe that larger datasets may work better, and we explore this in Sec. IV-C.

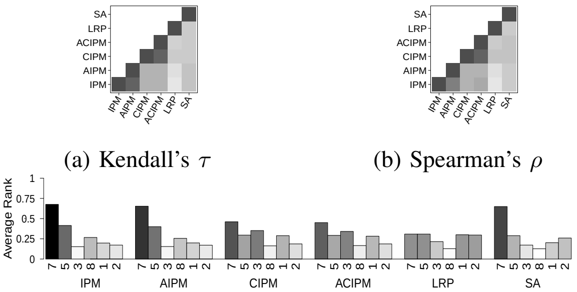
B. Real World Data

This section evaluates explanation over the datasets in Tab. IV. Each real dataset is accompanied by a graph displaying the six explanation methods and average feature rankings. These are kept consistent to enable comparisons, and the selection of features is constructed from the unique union of the top-4 AR across each explanation method.

We show correlation heatmaps between each pair of explanation methods, and this process is described in Sec. III-B. These figures are titled with the applicable correlation method, and explanation methods are labelled on each axis.

1) GlaucomaMVF: As we can see from Fig. 6c, features 64 and 66 were unanimously chosen among the best features across most (except LRP) explanation methods, showing that it is possible to produce similar explanations over many features [27]. IPM and SA are both certain that feature 64 is the most explanatory, while two of the four features are present across all methods. The performance accuracy of the base learners are listed in Tab. IV. Although the NN is less accurate than the RF and CF on this dataset, these findings identify that the same accuracy across models may not be necessary for explainability if there are informative features in the data.

Figures 6a & 6b show that the correlation between the forest methods is good, but there is little correlation between the forests and NN. Our interpretation is that the high feature count and the existence of few instances (170 in this dataset) effectively mean that only a few features are useful. The other features likely contributed very little information, and so these do not significantly impact classification, yet they skew the correlations. This made the extremely small explanation values undistinguishable from each other, and unimportant features were effectively ranked randomly between each explanation. We calculated correlations between the five best features of each explanation method (correlating only between the top features of each explanation method), and results were significantly better correlated. In light of this observation, we can argue that NNs and RFs are consistent with respect to the most important features on highly dimensional data.

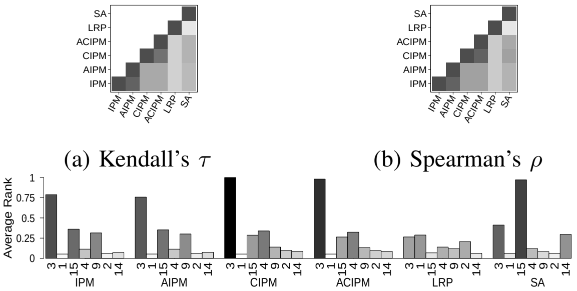


(a) Kendall's τ

(b) Spearman's ρ

(c) Feature Rankings

Fig. 7: Feature Rankings for the Abalone Dataset.



(a) Kendall's τ

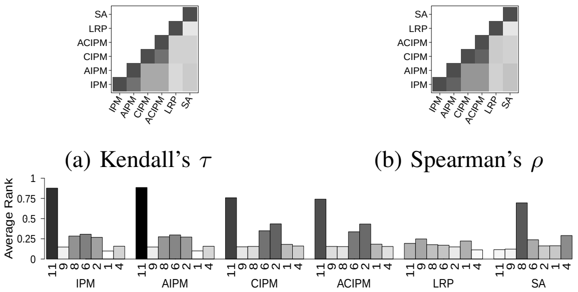
(b) Spearman's ρ

(c) Feature Rankings

Fig. 8: Feature Rankings for the Diabetic Retinopathy Dataset.

2) Abalone: Tab. IV shows that no learning algorithms have particularly high predictive accuracy on this dataset, but Fig. 7 shows that explanation methods agree on the majority of feature ranks. Feature 5 appears as the first, second or third most important feature in each explanation method, and all forest methods and SA confidently rank feature 7 as the most important. This highlights that the different learning algorithms can still agree to an extend regardless of predictive accuracy.

3) Diabetic Retinopathy: The correlation statistics in Fig. 8a & 8b are surprising, as SA shows high agreement with most other explanation methods, and different explanation methods show a weaker correlation between methods that should be intuitively similar. This indicates that the two

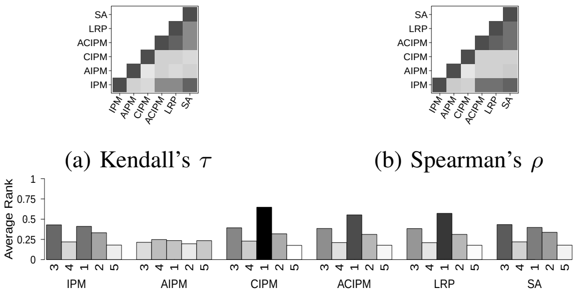


(a) Kendall's τ

(b) Spearman's ρ

(c) Feature Rankings

Fig. 9: Feature Rankings for the White Wine Dataset.



(a) Kendall's τ

(b) Spearman's ρ

(c) Feature Rankings

Fig. 10: Feature Rankings for the Website Phishing Dataset.

forests are prioritising features differently.

It should be noted that LRP poorly correlates with the other explanation methods. This is apparent as all others have respective agreement, and these rank features similarly, whereas LRP failed to find two of the three most important features agreed by all others. The fact that SA is consistent with the correlations of other methods indicates that the NN actually contain relevant explainability information, but LRP cannot find it as well as SA. This may be due to using a shallow NN, but it is quite interesting.

Figure 8c shows that features 3 and 15 appear among the most important features across most explanation methods, but this finding is not very confident. The feature-to-instance ratio of the dataset is quite low and this could explain when why there is a more scattered assignment of importance.

4) White Wine Quality: Figure 9 is the only dataset for which we have explicit domain knowledge, and professional opinions in the original paper [26] state that features 6 and 11 (bubbles and alcohol content) strongly influence each class of wine. We can see that all the methods identified those features, though in LRP they do not have the highest ranks and confidence is lower.

5) Website Phishing: All algorithms have high predictive accuracy and consistent feature rankings over this dataset. We can see in Fig. 10c that all methods recognise either features 1 or 3 as the most important.

This is the only experiment in which the results of LRP are strongly correlated with RFs, and where our adapted IPM method (ACIPM) can significantly contribute to finding useful features. These display a very strong positive correlation in Fig. 10a and 10b. LRP agrees on feature ranks with most other algorithms, and this is very much against the trend of other results presented in this paper. One intuitive way to explain the sudden rise in confidence is due to the structure of the dataset. The original feature values are trinary (1, 0, -1) and ordered. This means that there is a smaller possible input space ($3^{10} = 59049$ combinations) and algorithms now have much tighter solution bounds. The training data covers approximately $\frac{1}{59}$ of all possible results, so perhaps a single-layer NN can develop a better fit; thus improving the explanations of LRP. The prominence of the NN and LRP explanation measure means that it would be worth exploring the properties of these data in future research.

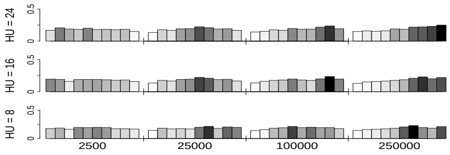


Fig. 11: Experiments focusing on LRP.

C. Further Simulation Over LRP

This research has explored various datasets, and the explanation results produced by LRP are surprising. We show in Fig. 1 that our implementation of LRP and NN setup produces visually accurate explanations over pixel inputs, whereas SA is incredibly scattered. LRP is unsure of feature importance on our simulated data, but other research [14] shows that LRP can provide very accurate explanations. We therefore explore other hyperparameter settings and sizes of data to generate more confident results.

Figure 11 shows further experiments focused on LRP. We again explore the problem in Sec. III-D:Dataset 4 (each feature has relative importance depending on feature index), and the setup of the experiments is similar to those in Tab. II (10 features, 1250 epochs, 32 BS). The control variables in each experiment are the number of simulated instances (x-axis), and the number of hidden units (external y-axis).

This figure is zoomed into the 0-0.5 range as explanations are not confident, but there is a pattern. As the number of instances and hidden units increase, the network becomes more certain of feature importances. This may indicate that NNs with more hidden layers could be better for LRP. The sub-graph in the top right (250,000 instances: 24HUs) shows a clear "triangle" of importance scores. While SA performs well on smaller datasets, LRP requires more data and greater hyperparameter values. We were unable to evaluate extremely large datasets in our original scenarios due to limitations in the R implementation of CFs.

V. Conclusion

To the best of our knowledge, this is the first systematic comparison of the explanations extracted from random forests and neural networks over data of different size and structure. Our results show that these methods can be consistent, but this is largely dataset dependent. We have seen

(Website Phising and Abalone) that explanations are the most correlated over data with few features and instances, and our results show that high predictive accuracy does not necessarily guarantee similar explanations between algorithms (Abalone and Diabetic Retinopathy). We thus conclude that small traditional datasets using informative features may produce the most similar explanations between sensitivity analysis using shallow neural networks and random forests.

Experiments in this paper extend the work of [15] by testing additional data types and sizes, and as recommended by [27], the IPM method has experienced additional deployment and increased dimensionality. It was observed that the methods which explain decisions of NNs are the most consistent with IPM over unbiased random forests. This is shown through the correlation metrics in Fig. 8a & 8b and the average ranks in Fig. 8c, 9c & 10c of Sec. IV. This confirms that removing bias from random forests [21], [27] is a sound objective for IPM and related methods. We have also identified that small values of the "mtry" parameter may bias feature importance, and this should be investigated further.

Our experiments show that LRP results were not always consistent with other methods, and we believe that this is because it is designed for deep neural networks and low-level (un-informative) features. We have seen that neural networks can identify feature importance, as sensitivity analysis performs well, but LRP struggles to find this. Finding real data that is equally applicable to both algorithms would be interesting to explore in future research. We may have unintentionally disadvantaged LRP through use of informative and high-level features, and discovering optimal scenarios for both learners would be interesting. We partially explore alternative neural network parameters in Sec. IV-C, but the objective of this paper was to explore explanation methods over RFs, CFs and NNs, and this proved difficult over larger datasets. We have begun to identify that correctness of LRP explanations increases with more hidden units and training data.

VI. Acknowledgements

We would like to thank Colin Johnson for suggesting corrections and proof reading this work.

References

[1] Alexandre Pupo. Cognitivity everywhere: the omnipresence of intelligent machines and the possible social impacts. *WFR*, 6(2):114–119, 2014.

[2] Benjamin Wilson, Judy Hoffman, and Jamie Morgenstern. Predictive inequity in object detection. *arXiv preprint arXiv:1902.11097*, 2019.

[3] W Whitt and A Maria. Introduction to modeling and simulation. In *WSC*, pages 7–13. IEEE, 1997.

[4] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[5] Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[6] Florian Tramer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, Jean-Pierre Hubaux, Mathias Humbert, Ari Juels, and Huang Lin. Fairtest: Discovering unwarranted associations in data-driven applications. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 401–416. IEEE, 2017.

[7] Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a right to explanation. *AI Magazine*, 38(3):50–57, 2017.

[8] Joshua A Kroll, Solon Barocas, Edward W Felten, Joel R Reidenberg, David G Robinson, and Harlan Yu. Accountable algorithms. *U. Pa. L. Rev.*, 165:633, 2016.

[9] Benjamin Letham, Cynthia Rudin, Tyler H McCormick, David Madigan, et al. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015.

[10] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc of CVPR*, pages 1–9, 2015.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Jozef Zurada. Could decision trees improve the classification accuracy and interpretability of loan granting decisions? In *2010 Hawaii International Conference on System Sciences*, pages 1–9. IEEE, 2010.

[13] Herbert L Jensen. Using neural networks for credit scoring. *Managerial Finance*, 18(6):15–26, 1992.

[14] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 2018.

[15] Irene Epifanio. Intervention in prediction measure: a new approach to assessing variable importance for random forests. *BMC Bioinformatics*, 18(1):230, 2017.

[16] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1096, 2019.

[17] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7), 2015.

[18] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[19] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[20] Torsten Hothorn, Kurt Hornik, and Achim Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical statistics*, 15(3):651–674, 2006.

[21] Carolin Strobl, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, 8(1):25, 2007.

[22] Jindong Gu, Yinchong Yang, and Volker Tresp. Understanding individual decisions of cnns via contrastive backpropagation. *arXiv preprint arXiv:1812.02100*, 2018.

[23] Andrea Peters, Torsten Hothorn, and Berthold Lausen. ipred: Improved predictors. *R news*, 2(2):33–36, 2002.

[24] Huiyu Zhou, Yuan Yuan, and Chunmei Shi. Object tracking using SIFT features and mean shift. *Computer Vision and Image Understanding*, 113(3):345–352, 2009.

[25] Shachar Kaufman, Saharon Rosset, Claudia Perlich, and Ori Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data*, 6(4):15, 2012.

[26] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *DSS*, 47(4):547–553, 2009.

[27] Stefano Nembrini. Bias in the intervention in prediction measure in random forests: illustrations and recommendations. *BMC Bioinformatics*, 2018.

[28] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[29] David Clark, Zoltan Schreter, and Anthony Adams. A quantitative comparison of dystal and backpropagation. In *Australian Conference on Neural Networks*, 1996.

[30] Bálint Antal and András Hajdu. An ensemble-based system for automatic screening of diabetic retinopathy. *Knowledge-based systems*, 60:20–27, 2014.

[31] Neda Abdelhamid, Aladdin Ayesh, and Fadi Thabtah. Phishing detection based associative classification data mining. *Expert Systems with Applications*, 41(13):5948–5959, 2014.