SancScreen: Towards a Real-world Dataset for Evaluating Explainability Methods

Matthias Jakobs[1], Helena Kotthaus[1], Ines Röder[2] and Maximilian Baritz[2]

[1]Artificial Intelligence Group, TU Dortmund University, Dortmund, Germany
[2]targens GmbH, Stuttgart, Germany

Abstract
Quantitatively evaluating explainability methods is a notoriously hard endeavor. One reason for this is the lack of real-world benchmark datasets that contain local feature importance annotations done by domain experts. We present *SancScreen*, a dataset from the domain of financial sanction screening. It allows for both evaluating explainability methods and uncovering errors made during model training. We showcase two possible ways to use the dataset for evaluating and debugging a Random Forest and a Neural Network model. For evaluation, we compare a total of 8 configurations of state-of-the-art explainability methods to the expert annotations. The dataset and code is available under https://github.com/MatthiasJakobs/sancscreen.

Keywords
explainability, quantitative evaluation, domain knowledge, financial application

1. Motivation

Explainability methods have become a common tool in recent years to gain insight into the decision process of complex machine learning models like Neural Networks and Random Forests. These models can be so complex that they are often treated as black-box models. Especially for heavily regulated sectors like finance, using black-box models is not possible without methods that explain the model behavior. While various methods have been proposed to extract explanations in the form of local feature importances [1, 2, 3], ways to quantitatively assess which method leads to better explanations of the model behavior have been hard to develop. We argue that one of the reasons is that there are no real-world datasets to benchmark explainability methods against. To the best of our knowledge, there is also no publicly available real-world dataset providing local feature importance annotations by domain experts. While there exist synthetic datasets [4, 5], their applicability for real-world use cases is naturally limited.

We present *SancScreen*, a dataset from the safety-critical real-world domain of sanction screening. It can be used to compare explainability methods against expert annotations of local feature importance. In sanction screening, financial transactions are checked to make sure that international sanctions are upheld. Part of the dataset is annotated by experts in the field of sanction screening, who rated how important each feature is depending on the ground truth

LWDA'22: Lernen, Wissen, Daten, Analysen. October 05–07, 2022, Hildesheim, Germany
matthias.jakobs@tu-dortmund.de (M. Jakobs); helena.kotthaus@tu-dortmund.de (H. Kotthaus)
0000-0003-4607-8957 (M. Jakobs); 0000-0002-7146-5695 (H. Kotthaus)

class label. In our evaluation, we compare different state-of-the-art explainability methods on two popular model families (Neural Networks and Random Forests) by measuring the distance between the generated annotations and the expert annotations. Also, we outline a potential use case for debugging. We measure the influence of a feature that is deemed to be irrelevant by the experts to the explainability methods evaluated in the previous experiment.

Our contributions can be summarized in the following points:

- We provide a novel real-world dataset, which contains local feature importance annotations by experts in the field.
- We show the usefulness of our dataset by conducting two experiments. In the first experiment, we evaluate state-of-the-art explainability methods by comparing them to the expert annotations. The second experiment outlines a possible use case for model debugging.

The rest of the paper is structured as follows. Section 2 discusses synthetic datasets for evaluating explainability methods, as well as state-of-the-art explainability methods. Section 3 presents the dataset and annotation process. In Section 4 we show how the dataset could be used to evaluate state-of-the-art explainability methods for Random Forests and Neural Networks. Lastly, Section 5 summarizes the findings and presents possible future work directions.

2. Related Work

Local explainability methods With the rise of complex machine learning models like Random Forests and Deep Neural Networks, concerns about their black-box nature rose as well. This is especially true for safety-critical application domains. To make models more transparent after training, multiple so called post-hoc explainability methods have been devised over the years. LIME [3] and SHAP [1] are among the most popular. For our experiments, we will use both SHAP and LIME, in addition to Integrated Gradients and Expected Gradients.

LIME generates samples by perturbing around the data point to explain. Then, it trains a linear classifier on the samples and their predictions of the original model. The proximity to the original data point is taken into account during training of the linear classifier, where closer samples get a higher weight. After training, the coefficients of the linear model can be used as local feature importance measures in the neighborhood around the data point.

SHAP is a framework based on Shapley values [6], which were originally developed in Game Theory to calculate the contributions of players to coalitions of other players. Shapley values, in the context of explainability methods, refer to the average influence of one feature $i \in M$ when it is added to each possible subset $S \subseteq M$ of features. Many Shapley-based explainability methods require a baseline or background distribution to sample missing feature values from, since marginalization over $\bar{S} = M \setminus S$ is required. [1] showed that multiple explainability methods can be reformulated into the SHAP framework. For example, by reformulating LIMEs proximity measure, loss function and regularization term LIME is able to produce Shapley values. The authors refer to this reformulation as KernelSHAP. DeepSHAP, a variant using the fact that Neural Networks provide easy access to gradient information, was proposed to generate Shapley Values in a faster way, if one can assume the model to be a Neural Network.

It is itself a reformulation of DeepLIFT by [7]. Additionally, [8] devise a method to compute Shapley values for Trees and Tree Ensembles called TreeSHAP. The method is able to generate Shapley values in polynomial time as opposed to the theoretically exponential runtime for measuring the influence of one feature to all other sets of features.

In computer vision, one popular approach for generating explanations is Integrated Gradients [2]. While this approach does not compute regular Shapley values like SHAP, it is equivalent to Aumann-Shapley values from Game Theory [9]. They extend Shapley values to infinitesimal changes in players to the output. It computes the integral over gradients of the models output with regard to the input. However, the input is a linear interpolation between a baseline and the data point to explain. Expected Gradient by [10] builds upon Integrated Gradients by taking the expectation of Integrated Gradients over a distribution of baselines, so the user does not have to pick one manually. The authors argue that choosing a distribution of baselines (such as the training distribution) is often easier in practice than choosing one single baseline.

Synthetic datasets for evaluation Recent work experimented with generating *synthetic datasets with known local explanations* [4, 5] to benchmark explainability methods. The authors used normally distributed features with varying amounts of correlation between the features in a regression setting. To generate the target values, [4] used two additive models of one-dimensional functions over each feature, resulting in $y = \sum_i f_i(x_i)$. First, a linear approach is used, where the target label y is a linear combination of each feature x_i with varying coefficients β_i, i.e., $y = \sum_i \beta_i x_i$. Second, the target label is generated by choosing f_i to be piecewise-constant. [5] extent the approach by [4] and include a third way to generate target values, where each function is not piecewise-constant but nonlinear instead. Choices for nonlinear functions include $f_i(x_i) = sin(x_i)$ and $f_i(x_i) = |x_i|$. Since all these approaches are additive and the functions are one-dimensional, this means that feature interactions are not taken into account. While the individual features might be highly correlated, the functions f_i only take one feature into account at the same time. This is unrealistic for many real-world datasets. Consider the trivial example $y = x_1 \cdot x_2$, where the features are interacting and influencing the output together, regardless of their correlation. We argue that these datasets might not be sufficient to evaluate explainability methods in a way that reflects their applicability on complex, real-world datasets, where we cannot assume the absence of feature interactions in general.

3. Description of Dataset

The basis of our dataset is a real-world dataset, consisting of international transactions that entered a sanction screening pipeline. All data points contain 19 features. From these 19 features, 14 are binary indicators. For example, one feature called *Name hit* encodes whether the name of the person involved in the transaction can be found in internal reference data. The remaining features are numeric. For example, *Number Countries* aggregates the number of different countries involved in the transaction. We want to emphasize that, due to the features being indicators and aggregators of internal reference data, no personal information about the people and companies involved in the transactions are exposed in or can be inferred from our dataset. For an overview of all features see Tab. 1. Notice that the meaning of three features

Feature name	Feature domain	Additional Remarks
Indicator_value	Numeric	Aggregation of multiple similarity values
Number_Abnormalities	Numeric	Number of rules violated
hidden_feature_3	Binary	Undisclosed for internal reasons
City_hit	Binary	Match against reference data
Account_hit	Binary	Match against reference data
Name_hit	Binary	Match against reference data
Other_hit	Binary	Match against reference data
hidden_feature_1	Binary	Undisclosed for internal reasons
Country_hit	Binary	Match against reference data
Address_hit	Binary	Match against reference data
Sanction_hit	Binary	Match against reference data
hidden_feature_2	Binary	Undisclosed for internal reasons
Bank_hits	Binary	Match against reference data
Account_Owner_hit	Binary	Match against reference data
Text_hits	Binary	Match against reference data
Specific_Country_hit	Binary	Match against reference data
Unique_hits	Numeric	Number of unique matches
Different_hits	Numeric	Number of matches
Number_Countries	Numeric	Number of countries involved in transaction

Table 1
Description of all 19 features of the *SancScreen* dataset.

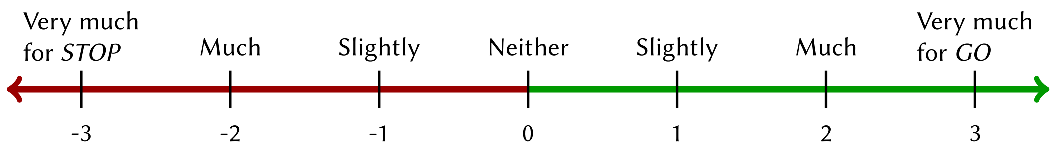

Figure 1: The rating scale given to the domain experts. Higher values indicated *GO* to the expertd. The opposite is true for the prediction of *STOP*.

(*hidden_feature_1*,*hidden_feature_2* and *hidden_feature_3*) cannot be disclosed in respect to trade secrets. The task is to classify whether a transaction needs to be halted. We will further refer to the possible outcomes of the sanction screening process as *STOP* and *GO*, respectively.

We sampled a representative set of transactions to provide local feature importance annotations for each data point and devised an annotation guide to rate each features relevance. Our guide is based on a numeric rating scale, ranging from -3 to $+3$ (see Fig. 1). We chose the range to be large enough to allow for some nuance in the annotation process but not be too overwhelming at the same time. The lower the value, the more the feature indicates the label *STOP*, while positive values indicate *GO*. It is also possible to rate the features influence as *neither*, meaning that the value of the feature is uninformative to the decision.

This results in a quantification of local feature importance, encoding expert knowledge in a way that allows us to compare it with attribution-based explanations. See Tab. 2 for the amount of data labeled with class labels and expert annotations.

In total, our dataset contains $1,002,860$ data points for which only the class labels are available.

Annotation type	n_{GO}	n_{STOP}	n_{total}
Class labels	$1,001,427$	$1,433$	$1,002,860$
Class labels + expert annotations	519	234	753

Table 2
Breakdown of the dataset by amount of annotations and class distribution. Due to the application setting, data points that need to be halted (i.e., *STOP*) are rare, which is reflected in this dataset.

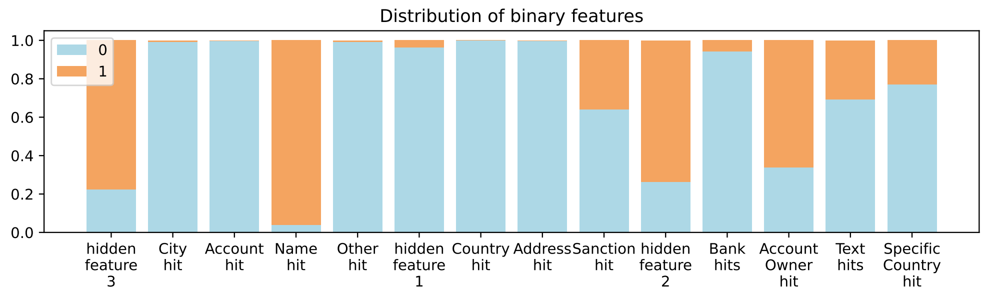

Figure 2: Distribution of feature values for all binary features in the dataset. Notice that some features, such as *Account hit* consists almost entirely of one value (in this case 0).

We split the dataset into training, validation and test parts to train the models. $2,288$ data points with equal amount of class labels were used for the training split. Additionally, 572 data points with equal amount of class labels were used for the validation split. We made sure to use equal amounts of both classes for training and validation in order to not prefer one class in production. This is due to the fact that unsuspicious transactions vastly outnumber those that are. We randomly picked a day in the logs of the real-world sanction screening pipeline and noticed that only 3 transactions were suspicious. This is reflected in the test split, where 3 transactions were suspicious and the remaining $999,997$ were not.

To gain more insight into the dataset, we calculated different statistics. First, we plotted the distribution of the binary features, shown in Fig. 2. The distributions show that many feature values are highly imbalanced. As an example, consider the features *Other hit*, *Address hit* and *Account hit*, which are zero for most data points, meaning that these indicators are probably highly relevant if they are one. The opposite case as with *Name hit* indicates that the presence of a name being recognized in the reference data is not unusual and is thus likely less informative. We conducted a similar analysis of the numerical features, as shown in Fig. 3. As can be seen, *Unique hits*, *Different hits* as well as *Number of countries* feature a small amount of variation, while the *Indicator value* is most spread out since it is a percentage value.

Lastly, we investigated the distribution of the expert annotations per feature. The distributions are shown in Fig. 4. As can be seen, many features contain a median value of zero, meaning that they can either be salient for a *GO* or a *STOP*, depending on the context of the other features. Notice that the median value of *Name hit* is positive, with most of distribution of annotations being positive as well. Additionally, *Account hit* is mostly negative or zero.

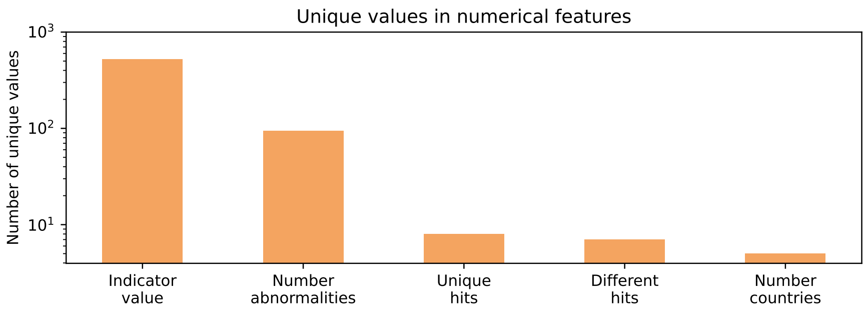

Figure 3: Number of unique values for the five numerical features in the dataset. Notice that the y axis is logarithmic for better readabililty. While *Number countries* has a small amount of variation, the *Indicator value* is spread more widely since it is a percentage.

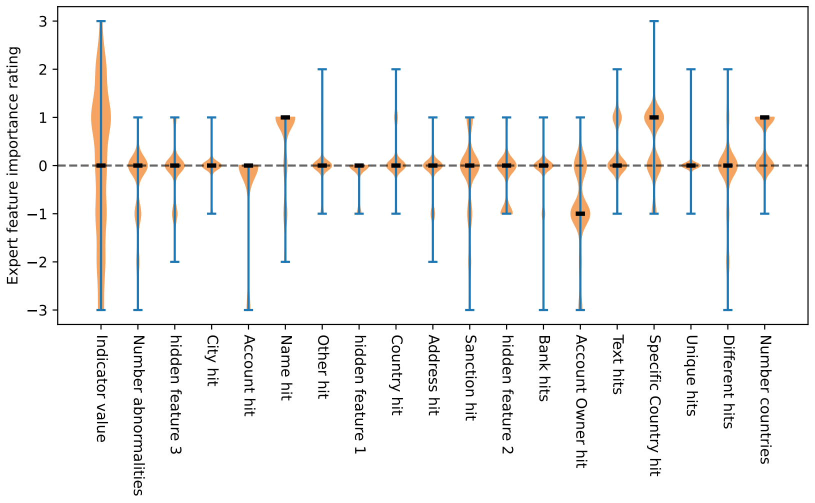

Figure 4: Distribution of the expert annotations over all annotated data points. The black horizontal bar for each feature indicates the median value of the annotation. Notice that some features, such as *Account hit* never indicate a positive feature importance, while *Indicator value* is fairly uniformly distributed.

4. Evaluation

To show the potential of the dataset, we conducted two experiments. First, we compare explanations generated by multiple state-of-the-art explainability methods on Random Forests and Neural Networks to the expert annotations. Second, we show a potential use case for debugging the models by focusing on which method better identifies unimportant features.

Metrics Since to our knowledge there appears to be no consensus in the literature on one specific metric, we propose using the following three metrics, since they are scale independent and vary in what aspect they measure. We will define $e^{(i)}$ to be an explanation or annotation of data point i and $r^{(i)}$ being its corresponding ranking.

First, we compute a ranking of both generated explanations and expert annotations to see the relative importance of each feature. Since during the rating process multiple features can be equally (un)informative, we compute fractional rankings, meaning that features get assigned the mean of all possible, equally valid rankings. As an example, consider an expert annotation $e^{(i)} = (+3, 0, 0, -1)$. Clearly, both rankings $r^{(i)} = (4, 3, 2, 1)$ and $r^{(i)} = (4, 2, 3, 1)$ are equally valid. Using fractional ranking, the computed rank of $e^{(i)}$ is equal to $\bar{r}^{(i)} = (4, 2.5, 2.5, 1)$, where the ranking of each value is given by the mean of all possible equal rankings. This procedure is also necessary if an explanation generated by an explainability method contains multiple identical values. From now on, we will refer to fractional rankings $\bar{r}^{(i)}$ simply as $r^{(i)}$ for ease of notation. Then, we compute the distance between both rankings using Spearman's Footrule:

$$dist_{sf}(e^{(a)}, e^{(b)}) = \left(\lfloor \frac{1}{2} n^2 \rfloor\right)^{-1} \sum_{i}^{n} | r_i^{(a)} - r_i^{(b)} |$$

As shown by [11], Spearman's Footrule is upper-bounded by $\lfloor \frac{1}{2} n^2 \rfloor$, where n is the number of values in the ranking. We thus normalize the distance accordingly, meaning that the worst possible distance between two rankings is 1.

Second, we use cosine distance between two explanations to get a sense of alignment between them without taking scale into account. We argue for the use of cosine distance since different explanation methods provide varying scales, which also vary from the rating scale.

$$dist_{cos}(e^{(a)}, e^{(b)}) = \frac{1}{2}\left(1 - \frac{e^{(a)} \cdot e^{(b)}}{||e^{(a)}||_2 \; ||e^{(b)}||_2}\right)$$

Third, we propose a distance measure based on Hamming distance. We average three Hamming distances h between indicators of positive values ${}^{+}e$, negative values ${}^{-}e$, and those values whose absolute value is smaller than or equal to a fixed value $\epsilon = 10^{-8}$ indicated by ${}^{\epsilon}e$.

$$dist_{hamming}(e^{(a)}, e^{(b)}) = \frac{1}{3} \sum_{s \in \{+,-,\epsilon\}} h({}^{s}e^{(a)}, \; {}^{s}e^{(b)})$$

Model training We train both a Random Forest and a fully-connected Neural Network on a split of the dataset for which only class labels are present. See Tab. 3 for a breakdown of the model performance on the training and test split used. We used the implementation of *scikit-learn* for the Random Forest and *PyTorch* for the implementation of our Neural Network.

For our Neural Network model architecture, we chose a simple fully-connected Neural Networks with two hidden layers, of which each layer contains 50 neurons. Since we achieved good results using this model architecture, we did not investigate the choice of these hyperparameters further. We trained the model with a learning rate of 0.0003 using the Adam optimizer and a binary cross-entropy loss. To avoid overfitting, we monitored the loss on the validation split and utilized early stopping if the loss did not reach a new minimum for 50 consecutive epochs. We saved a model checkpoint on every new minimum validation loss value and reset the model to the latest checkpoint if early stopping came into effect.

	AUROC (train)	AUROC (test)
Random Forest	0.997 ± 10^{-5}	0.993 ± 10^{-4}
Neural Network	0.990 ± 10^{-3}	0.998 ± 10^{-3}

Table 3
Mean Area under ROC curve on train and test split for both trained classifiers after 10 repetitions. We also provide the standard deviation.

Name	Method	Models	Configuration
KS-Zero	KernelSHAP	Both	Baseline of all-zeros
KS-Data	KernelSHAP	Both	Background distribution from training data
LIME	LIME	Both	Background distribution from training data
TreeSHAP-Int	TreeSHAP	RF	Background distribution from training data
TreeSHAP-Cond	TreeSHAP	RF	Utilizes information from RF training
IG	Integrated Gradients	NN	Baseline of all-zeros
EG	Expected Gradients	NN	Background distribution from training data
DeepSHAP	DeepSHAP	NN	Background distribution from training data

Table 4
Overview of the explainability methods and their configurations. The first column shows the name we use to refer to these configurations. The second and third column show the name of the explainability method and which models they were applied to (Random Forest, Neural Networks or both). The last column summarizes the configuration parameters in terms of background distributions or used baselines.

After training the network, we tested it on the test split and found that no mistakes in predicting the suspicious transactions were made, and 22, 083 transactions were falsely identified to be suspicious. While the error rate seems high, we stress that it is of utmost importance that no truly suspicious transactions gets misclassified as unsuspicious. A comparatively small error rate on unsuspicious transactions of 22% is considered a good result in our application setting.

To train the Random Forest, we used the implementation provided by *scikit-learn*. We combined the training and validation splits again and provided both to the Random Forest for training. After some experimentation, we found good results on the dataset using 300 estimators. All other hyperparameters are default to the provided implementation, meaning that we did not use a maximum depth of the trees, we used Gini Impurity and we used Bootstrapping. As discussed in the training of the Neural Network, we did not feel the need to further optimize the Random Forest's hyperparameters due to a satisfying fit on the training and test data.

After fitting the Random Forest model, we observed zero transactions that were wrongly classified to be unsuspicious. In comparison, the error rate on unsuspicious transactions was 15, 657. The Random Forest is thus able to achieve a lower error rate on the test set compared to the Neural Network, while also not wrongly classifying any suspicious transactions.

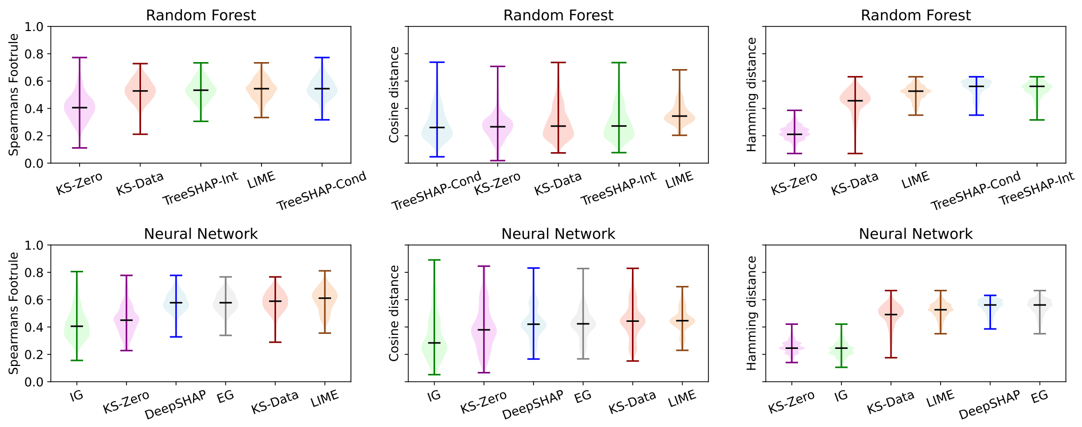

Figure 5: Distances between the explanations generated by the explainability methods and the expert annotations, measured using Spearman's Footrule, Cosine distance and Hamming distance. The methods in each plot are sorted according to their median distance.

4.1. Measuring explanation alignment to the expert annotations

Using the previously trained models, we extract explanations using state-of-the-art explainability approaches. For the Random Forest, we generate explanations using TreeSHAP [8], both in its interventional and conditional variant, KernelSHAP [1] and LIME [3]. For KernelSHAP, to model a feature being absent, its value can either be sampled from a reference dataset or set as a fixed baseline value. We evaluate both the use of the original training distribution for sampling and using a fixed baseline of all zeros to model feature absence. We will further refer to these variants as *KS-Data* and *KS-Zero*, respectively. Additionally, we configure TreeSHAP in two different ways. First, the so-called interventional approach (*TreeSHAP-Int*) requires a background dataset, similar to KernelSHAP, because missing features are sampled independently from the background dataset. On the other hand, the conditional approach (*TreeSHAP-Cond*) utilizes information gained during the Random Forest training process to sample missing values conditioned on the present features. For the evaluation of the Neural Network model, we use the model-agnostic explainability methods (KernelSHAP and LIME) and add *DeepSHAP*, *Integrated Gradients (IG)* and *Expected Gradients (EG)*. These methods are designed to be used for gradient-based models and can thus be easily used with Neural Networks. *IG* also needs a specific baseline used in its interpolation process, and we also choose a baseline of all zeros for better comparison with KernelSHAP. See Tab. 4 for an overview of all configurations used. We used the SHAP library [1] for KernelSHAP, DeepSHAP and TreeSHAP, the official LIME library [3] for LIME and Captum [1] for Integrated Gradients. We implemented Expected Gradients ourselves using the information provided in [10].

For each explainability method, we compute explanations on the fully-annotated part of the dataset and plot their distances to the respective expert annotations. As a distance measure, we use Spearman's Footrule, Cosine distance and Hamming distance, as presented earlier. As can be seen in Fig. 5, *KS-Zero* performs very well for the Random Forest model in terms of median

[1]https://captum.ai/

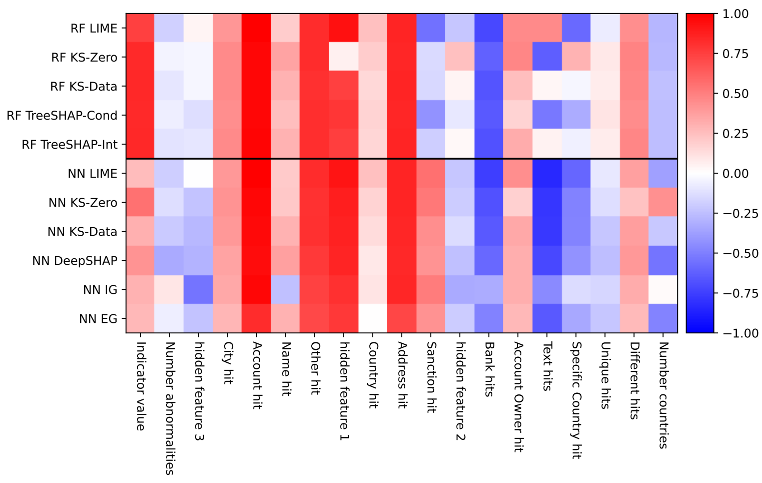

Figure 6: Pearson correlation coefficients between the expert annotations and the explanations generated by several methods (left). Red values indicate positive correlation between expert annotations and explanation method, while blue values indicate negative correlation.

distance according to all metrics. The same can be said for *KS-Zero* for the Neural Network model, where *IG* also performs well. Since both *IG* and *KS-Zero* feature the same baseline of an all-zero vector, we suspect that this baseline is the main cause of their performance since most features are binary. We noticed that Hamming distance gave the clearest indication of the best performing methods. For example, consider that the most distant explanation in *KS-Zero* for the Random Forest is still closer than the median distance for *KS-Data* (see top right plot in Fig. 5).

We additionally conducted a Wilcoxon signed-rank test to test whether the differences in computed distances between each pair of explainability methods are significant. With a significance level of $p = 0.01$, all differences are statistically significant except for the difference between *LIME* and *KS-Data* for the Neural Network using Cosine distance.

Finally, we investigated how well the explanations between the annotations and each of the previous explainability methods align, and if there are patterns transcending explainability methods and model classes. The results are shown in Fig. 6. We noticed that *Account hit*, *Other hit* as well as *hidden_feature_1* strongly correlate with the annotations for nearly all explainabilitity method, regardless of whether the Random Forest or Neural Network was used. This is consistent with the finding in Sec. 3 that these features are fairly imbalanced in terms of their values, meaning that unexpected values are probably highly salient. Interestingly, *Bank hits* shows the opposite phenomenon. In the annotations, *Bank hits* tended to be either uninformative or in favor of *STOP* (see Fig. 4), suggesting that the Neural Network and Random Forest where able to determine an effect towards *GO* instead.

4.2. Debugging use case: Identifying irrelevant features

Next, we present a simple use case for the debugging the trained models using expert annotations. While investigating the expert annotations, we found that some features were almost always

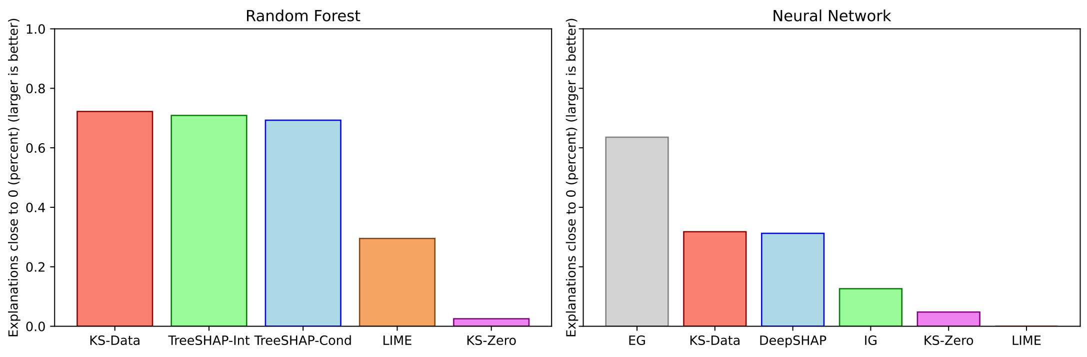

Figure 7: Comparison of how often the feature *Unique Lists* was attributed zero relevance. The experts rated this feature to be irrelevant for 96% of data points. We defined a tolerance of 0.01, meaning that we set the explanations to zero if the absolute value was less than 0.01.

irrelevant. A particularly striking example was the feature *Unique Lists*, which encodes how many internal datasets contained information from the transaction. In 96% of annotations this feature was said to be irrelevant. We conducted an experiment where we investigate how important this feature was to both the Random Forest and the Neural Network, according to all previously used explainability methods. We round the generated explanations for *Unique Lists* to exactly zero if the absolute value was lower than 0.01 for fairer comparison. The results are shown in Fig. 7. Notice that the best performing explainability methods from the previous experiment, namely *KS-Zero* and *IG*, were among the worst methods for both models in terms of importance of feature *Unique Lists*. We observe that the methods that incorporate a background dataset of training data points (*KS-Data*, *TreeSHAP-Int* and *EG*) tend to perform better than the ones with simple all-zero baselines. However, even these methods still attribute many data points with a non-zero importance for a feature which, according to the annotations, is irrelevant most of the time. This suggests that the Random Forest and Neural Network sometimes utilized this feature in their decision making, even though the feature is highly irrelevant. With this insight gained, one could focus on specifically regularizing mostly irrelevant features such as *Unique Lists* during the training process, which will result in models that are more trustworthy due to them not relying on spurious correlations in irrelevant features.

5. Conclusion

We presented *SancScreen*, a real-world dataset containing local feature importance annotations derived from domain knowledge. By conducting two simple experiments we showed the potential of the dataset for evaluating explainability methods and debugging a trained model. Additionally, using three different metrics of alignment we compared different explainability methods, regardless of the absolute values of their explanation, to the expert knowledge. Finally, we discussed the impact of datasets like *SancScreen* on the future evaluations of explainability methods and outlined a procedure which we hypothesize could be used to gain more insight into up- and downsides of popular explainability methods.

In the future, we plan to incorporate some of the expert annotations during the training process of the models in order to regularize the models to be more in line with the experts. This will result in a more fair comparison, since we can take a part of the fully-annotated annotations for training to prime the model to reason along the same lines as the expert. Fully-annotated test data can then hopefully be used as ground truth annotations. Moreover, we plan to derive ways to incorporate explanations in more machine learning models besides neural networks, for which some work exists, namely by incorporating explanations in tree ensembles like Random Forests. This should lead to a wider adoption of evaluating explainability methods in application fields where neural networks are not commonly used.

Acknowledgments

This research has been funded by the Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the Lamarr-Institute for Machine Learning and Artificial Intelligence, LAMARR22A.

References

[1] S. M. Lundberg, S.-I. Lee, A Unified Approach to Interpreting Model Predictions, in: Advances in Neural Information Processing Systems 30, 2017, pp. 4765–4774.

[2] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: ICML, Sydney, Australia, 2017, pp. 3319–3328.

[3] M. T. Ribeiro, S. Singh, C. Guestrin, ”Why Should I Trust You?”: Explaining the Predictions of Any Classifier, in: SIGKDD, Association for Computing Machinery, San Francisco, USA, 2016, pp. 1135–1144.

[4] K. Aas, M. Jullum, A. Løland, Explaining individual predictions when features are dependent: More accurate approximations to Shapley values, Artificial Intelligence 298 (2021).

[5] Y. Liu, S. Khandagale, C. White, W. Neiswanger, Synthetic Benchmarks for Scientific Research in Explainable Machine Learning, arXiv:2106.12543 [cs, stat] (2021). ArXiv: 2106.12543.

[6] L. S. Shapley, A value for n-person games, Contributions to the Theory of Games 2 (1953) 307–317.

[7] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: ICML, Sydney, Australia, 2017, pp. 3145–3153.

[8] S. M. Lundberg, G. G. Erion, S.-I. Lee, Consistent Individualized Feature Attribution for Tree Ensembles, arXiv:1802.03888 [cs, stat] (2019). ArXiv: 1802.03888.

[9] R. J. Aumann, L. S. Shapley, Values of Non-Atomic Games, 1974.

[10] G. Erion, J. D. Janizek, P. Sturmfels, S. M. Lundberg, S.-I. Lee, Improving performance of deep learning models with axiomatic attribution priors and expected gradients, Nature Machine Intelligence (2021) 1–12.

[11] P. Diaconis, R. L. Graham, Spearman’s footrule as a measure of disarray, Journal of the Royal Statistical Society: Series B (Methodological) 39 (1977) 262–268.