university of groningen / faculty of science and engineering

Responsible AI: Behind the Rationale of Neural Networks

Bachelor's Project Thesis

Bram Rijsbosch, s2894645, b.u.rijsbosch@student.rug.nl,
Supervisor: C.C. Steging (PHD-candidate, Responsible AI)
Second assessor: Prof. Dr. B. Verheij

Abstract: Problems with complex machine learning models have led to growing concerns and a spiking interest in responsible artificial intelligence. An important subfield of responsible AI, explainable AI (XAI), has already led to the development of techniques capable of explaining the decision-making of these black-box systems, yet this is not enough; after all, as demonstrated in previous research, machine learning techniques may appear to perform well, obtaining high accuracy levels scores with test data, while actually reasoning with an unsound rationale. Using complex self-learning systems that unknowingly reason with an unsound rationale can have devastating real-world effects. This study therefore further explores the issues concerning the rationale of these complex machine learning systems. Using a new artificial domain, based on real-world conditions, this study confirms the result that neural networks can achieve high levels of performance in terms of classification accuracies, while not learning the conditions that define the data sets. It is demonstrated that the standard techniques, such as using more data, deeper networks or less noise, do not aid in solving this problem. Additional experiments, focused on finding more responsible practices, do reveal that using synthetic training data built upon domain knowledge can help to improve the rationale while maintaining high levels of accuracy.

1 Introduction

Despite their widespread use, many advanced AI models still function as black boxes (Ribeiro et al., 2016). These models can obtain great results in, for example, classification tasks but offer no clear explanations. However, understanding and interpreting the reasoning behind the predictions made by such systems is crucial for establishing trust for users, combating problems such as biased systems, and having the ability to use these techniques in decision-making (Akata et al., 2020). In addition, it can help designers to transform untrustworthy models into trustworthy ones (Ribeiro et al., 2016).

Considerable work has already been conducted on interpreting and explaining the predictions of complex black-box models. This has led to the development of explainable AI (XAI) techniques. The goal of these techniques is to add explanations of the reasoning, while maintaining high levels of performance (Gunning, 2017). Recently, the focus of the XAI techniques has shifted toward explaining (1) the training process, 2) the relationship of the models to the training material, and 3) the reasoning behind the underlying algorithms (Akata et al., 2020). Considering the last point, a popular XAI technique that is used to interpret the rationale of complex machine learning models is SHAP. SHAP (SHapley Additive exPlanations) is argued to be the most accurate and consistent method for explaining the output of machine learning models (Lundberg and Lee, 2017). SHAP computes shapley values (originating from classic cooperative game-theory) for each feature in a data set; these values then indicate the influence that each feature has on the predictions of a model.

Our ability to explain the predictions of complex machine learning models is an important step in developing trust and making AI more responsible, yet this is not enough. An important problem remains, namely that these complex models can appear to make correct decisions, obtaining high accuracies,

without actually using a correct rationale. This is clearly demonstrated in the study of Bench-Capon (1993), in which an artificial problem in the law domain was used to discover the rationale of neural networks. The problem considered was a classification task for the eligibility of welfare benefits. By using a fictional data set, containing 12 relevant features and 52 additional noise attributes, it was revealed that excellent results could be obtained (approximately 99% classification accuracies) without the networks using a fully correct rationale. Investigations into the rationale of the networks revealed that, in fact, only four of the six conditions required for making correct classifications were considered by the neural networks. This suggests that strong performance does not necessarily correspond to a sound rationale. Furthermore, other research using the same artificial domain has demonstrated that other algorithms can achieve similarly high accuracies without using a fully correct rationale (Johnston and Governatori, 2003; Možina et al., 2005). In addition, a recent study has revealed the same effect for neural networks and decisions tree algorithms by using different artificial data sets (Steging et al., 2019). Thus, even when using XAI techniques to explain the predictions of a model, the explanations given can still be irrational if the model reasons with an incorrect rationale.

Working with machine learning algorithms that use such an incomplete or incorrect rationale can have dramatic effects. A recent real-world example of the devastating impact of these complex self-learning AI techniques can be seen in the Netherlands: In the so-called "benefits scandal" (in Dutch: toeslagenaffaire), the Dutch tax authorities had incorrectly and harshly prosecuted thousands of families as fraudulent applicants for childcare benefits. As stated in the investigative reports of this scandal, the tax authorities used a complex self-learning risk-classification model to determine potential fraudulent applicants (Tweede Kamer, 2020; Autoriteit Persoonsgegevens, 2020). Additionally, the reports state that this risk-classification model had an in-proper and discriminatory working and that the employees handling the risk classifications could not see on the basis of which indicators an application had been given a certain risk score. These incorrect classifications, combined with tough prosecution and many mistakes in addressing the problem, eventually led to the resignation of the entire Dutch cabinet.

The Dutch benefits scandal demonstrates the need for responsible AI that is both explainable and reasons in a sound fashion. With the rapid development of AI and the increasing usage of AI models by both governments and companies, this need is extremely high. This research therefore further explores the issues concerning the rationale of these complex machine learning models. As it is impossible to study the wide range of machine learning techniques in only one paper, this study focuses on neural networks, as they are considered to be among the least explainable machine learning techniques, while achieving the highest accuracies (Gunning, 2017).

The goal of this study is to further explore the issues concerning the rationale of neural networks. As Bench-Capon revealed in his 1993 paper, these neural networks can achieve high classification accuracies without actually reasoning with a correct rationale. The first step is to replicate the study of Bench-Capon (1993) to determine how these results compare with the current neural network techniques. As already demonstrated by Steging (2018), replicating the study should lead to similar results. Following this, several additional experiments are performed on the original domain to assess how the findings hold when using different neural networks, more data, less noise, or state-of-the-art XAI techniques.

The study of Bench-Capon uses a fictional "welfare benefit" domain with highly specific conditions and data sets; it is therefore interesting to discover how the results generalize to a domain with different conditions, based on a real-world benefits example. The subsequent step in this study thus involves using the methods of the Bench-Capon study on a new domain, one that uses real-life conditions that are based on eligibility for a Dutch childcare benefit. This domain is selected because problems with these childcare benefits led to the Dutch benefits scandal. Although it is impossible to replicate the risk-classification algorithm that was used, the investigative reports of this scandal are used to make informed assumptions about data sets, conditions, and preprocessing techniques to study the rationale of neural networks in a more real-life setting and as-

sess how the results compare with the findings of the Bench-Capon study.

This study hereby aims to contribute to the existing research by creating a better understanding of the rationale of neural networks and possibly discover new methods that can aid in making our use of these complex models more responsible.

2 Neural Networks and Open Texture

This chapter provides an extensive summary of the 1993 paper "Neural Networks and Open Texture" by Bench-Capon, as the methods of the paper function as a starting point for this study.

2.1 Domain

In his 1993 paper, Bench-Capon aimed to test the potential of neural networks for open texture problems in the law domain. In particular, his goal was to investigate three questions: whether these neural networks could obtain a high degree of success, whether, in that case, they also used a correct rationale, and whether he could discover that rationale. Even though his paper was written nearly 30 years ago, these questions are still relevant today, as the same problems concerning the rationale and explainability of these black-box systems remain.

To be able to answer these three questions, Bench-Capon created an artificial domain. A real domain would be impractical, because when a real domain is not understood perfectly, it is impossible to effectively evaluate the rationale of the networks. The problem he created was an eligibility test for a fictional welfare benefit, which is supposedly paid to pensioners to visit their spouse in the hospital. To be eligible for this benefit, a person must satisfy each of the following six conditions, as described by Bench-Capon (1993):

C1. The person should be of pensionable age (60 for a woman, 65 for a man);

C2. The person should have have paid contributions in four of the last five relevant contribution years;

C3. The person should be a spouse of the patient;

C4. The person should not be absent from the UK;

C5. The person should have capital resources not amounting to more than 3,000 pound;

C6. If the spouse is an in-patient (living in the hospital), the hospital should be within 50 miles. If the spouse is an out-patient, the hospital should lie beyond this distance.

$(C1 \wedge C2 \wedge C3 \wedge C4 \wedge C5 \wedge C6) \rightarrow \text{Eligible}$

These conditions were selected because they consist of a combination of typical conditions used in data sets. The conditions vary over Boolean conditions (C3, C4), continuous variables (C2, C5), and conditions in which the satisfiability depends on multiple variables (C1, C6), which is supposed to be slightly more difficult to discover for a neural network.

To evaluate a person's eligibility using these six conditions, 12 different features are needed, as the first and sixth conditions depend on two features, while the second condition depends on five separate features. In addition, 52 noise features were added to each instance in the data sets. This was done to test whether the inclusion of irrelevant factors would influence the results, as noise also occurs in real data sets.

2.2 Methods

Bench-Capon used these features to create several different data sets to evaluate both the performance and the rationale of the networks. He assumed that a real-life data set would most likely be one in which cases can fail on multiple conditions simultaneously. Thus, he began by creating a data set in which each ineligible case would definitely fail on one condition while randomly generating the values for the other features, thus leading to cases failing on multiple conditions. Another data set was then used to study the rationale of the networks: In this data set, each ineligible case would fail on only one specific condition. This allows for the rationale to be studied, as each failing case then tests for a single specific condition. These two data sets are referred to as the multiple fail set and single fail set, respectively. Moreover, two additional data sets were used by Bench-Capon to graphically depict the performance on the two most difficult conditions: C1 and C6, whose satisfiability depends on two different features interacting. The multiple and single fail sets are split into both train and test sets. Train sets

were comprised of 2,400 different cases, while test sets were comprised of 2000 cases. To summmarize, this would lead to the following data sets:

- Multiple fail set: In the multiple fail set, each ineligible instance can fail on multiple conditions. Half of the instances satisfy all six conditions and are thus considered to be eligible cases. These eligible cases are generated by varying the values of each feature values randomly over their eligible range. The ineligible cases specifically failed on at least one of the six conditions (divided equally), and the other values were generated randomly over their full range.
- Single fail set: This set is likewise split equally with eligible and ineligible cases. The eligible cases are generated in the same way as in the multiple fail set. The ineligible cases, however, now fail on exactly one of the six conditions. The number of failing cases for each condition is divided equally.
- Age set: A set in which all conditions except the age condition (C1) are satisfied. This way, a network that has not learned the age condition correctly will not perform well on this Age set. The age set is created by varying the age feature from 0 to 100 in steps of five, for both men and women. This yields both eligible and ineligible cases and allows for the age effect to be plotted graphically.
- Distance set: Similar to the age set, but now the ineligible cases fail only on the distance condition (C6). The distance set is created by varying the distance feature in steps of five from 0 to 100, for both in- and out-patients.

Finally, Bench-Capon employed three different neural networks in his study. All three networks use the conventional triangular shape, in which the number of nodes are decreases over the different layers. The networks have an input layer with 64 nodes, corresponding to the 64 features used, and a single output node representing the eligibility label. They networks vary with respect to the number of nodes and hidden layers:

- One hidden layer: 12 nodes in the hidden layer.
- Two hidden layers: 24 nodes in the first hidden layer, and 6 nodes in the second hidden layer.
- Three hidden layers: 24 nodes in the first, 10 in the second and 3 in the third hidden layer.

2.3 Results and Discussion

Bench-Capon begins by evaluating whether the networks can obtain a high degree of success on his artificial problem. This is done by training and testing the networks with the multiple fail set. The following classification accuracies are obtained:

- One hidden layer: 99.25%
- Two hidden layers: 98.90%
- Three hidden layers: 98.75%

As stated by Bench-Capon (1993): "this was a very encouraging level of performance and might be considered acceptable, even in a legal application" (p. 294).

Next, he investigates the rationale of the neural networks by testing the trained networks with the single fail data set, yielding the following results:

- One hidden layer: 72.25%
- Two hidden layers: 76.67%
- Three hidden layers: 74.33%

These are surprising results, as the low accuracies reveal that although the performance of the networks appears to be good, the rationale used by the networks is actually incomplete. If the networks would have learned all the six conditions, they would be able to correctly classify the cases in the single fail set. This incomplete rationale is clearly expressed when Bench-Capon uses the the special age- and distance sets to plot the effect on these conditions. The distance graph is a straight line at an output of one for all three networks, indicating that the networks do not consider this conditions at all when determining the eligibility of an instance. Meanwhile, the age graph reveals that the one- and two-layered networks consider ages starting at 20–30 to be satisfiable, which is far lower than the actual ages of 60 for women and 65 for men. Moreover, the three-layered network is a straight line at an output of one again.

Hereby, Bench-Capon has demonstrated that neural networks with apparently good levels of performance can actually rely on an inadequate rationale. This hypothesis, however, depends on strong

assumptions regarding the nature of the training data used. Thus, to further analyze these results, he uses the single fail set to train the networks. This yields the following accuracies when tested on the multiple fail set again (the three-layered neural network now not longer converges):

- One hidden layer: 99.25%
- Two hidden layers: 99.0%

Again, he investigates the rationale by testing with the single fail test set:

- One hidden layer: 97.91%
- Two hidden layers: 98.08%

Both networks achieve high accuracies on the multiple fail set. This time, however, high accuracies are likewise obtained on the single fail test set, indicating an improved rationale. This is confirmed when the age effect is plotted again, which indicates that ages of around 45–50 are now considered to be eligible, which suggests an improved, though still imperfect, rationale.

The results of Bench-Capon indicate that the performance of neural networks on test data is not a clear guide for the correctness of the rationale, especially if one's domain knowledge is limited. The results achieved with the single fail set, however, reveal that with the right training data, the rationale can be improved, while maintaining high performance on the original test data. Yet, preparing such training data critically depends on one's domain knowledge.

3 Replication Study: Welfare Benefits Domain

Prior to implementing the new child benefits domain, a replication of the study of Bench-Capon is performed. This is done to get a further grasp of the problem, to determine how the results compare with our current technologies, and to implement several additional experiments on the original domain. As already demonstrated by Steging (2018), a replication should lead to the same general findings, but small differences in the accuracies can occur, as several assumptions must be made regarding the original domain. These assumptions are discussed in the methods section, after which the results are discussed, and additional experiments are performed.

3.1 Methods

In the original study, six conditions determine a person's eligibility for the fictional welfare benefit, and 12 features are needed to evaluate these conditions. In addition, each instance in the data sets includes 52 noise features. A list of all features and their ranges of values are provided in Table 3.1. The range of values for each feature was not specified precisely by Bench-Capon, therefore some small differences might exist here. Appendix A.1 includes a logical representation of the six conditions for the welfare benefit problem using the features of Table 3.1.

Table 3.1: The features used in the data sets of the welfare benefit domain

Feature	Range
Age	0-100
Gender	Male/Female
Paid contributions: five separate features	True/False
Spouse	True/False
Residence	True/False
Capital resources	0-10,000
Patient type	in/out
Distance from hospital	0-100
Noise variables: 52 separate features	0-100

Next, alongside the data sets presented in Section 2.2, this replication study uses four additional data sets to evaluate the exact performance of the networks on all six conditions, instead of evaluating the performance on only the age and distance conditions. Table A.1 in Appendix A.2 precisely outlines the structure and cases in each data set.

Furthermore, Bench-Capon mentions that he implements the three neural networks with the–now outdated–Aspirin software (Leighton and Wieland, 1991), therefore several assumptions and alterations are made here: First, the networks are implemented with the Scikit-learn Python library, in which the standard MLP classifiers are used. Sec-

ond, as explained in the replication study of Steging (2018), the sigmoid activation function was the most common activation function at that time, so this is used here, as well. Furthermore, as Steging also demonstrated, the type of gradient descent had little influence on the results, thus the common mini-batch approach is used. Other parameters such as the learning rate and batch size are found through a hyperparameter optimization process. A summary of the exact settings of the networks is presented in Appendix A.3.

3.2 Results and Discussion

The networks are trained with the multiple- and single fail sets again. As done in the original study, the trained networks are then tested on all the test sets. Tables 3.3 and 3.4 present the resulting accuracies when averaged over 100 runs (as is the case for all results in this study). Table 3.2 presents the accuracies as reported by Bench-Capon again, for comparison.

The accuracies of Tables 3.3 and 3.4 confirm the results of the original study. When trained with the multiple fail set, the networks can obtain a high level of performance (98+% average accuracies on the multiple fail test set), while they reason with an incomplete rationale (74+% accuracies on the single fail test set). Moreover, when trained with the single fail set, the performance remains high, although slightly lower than in the original study, while the rationale has improved considerably to accuracy scores of around 90% on the single fail set.

Further evaluating the rationale through the use of the special data sets for each condition expresses these results more clearly. The accuracies on the special condition test sets in Table 3.3 reveal that the easier Boolean conditions are learned relatively well, though these still have lower accuracies than expected (around 86%). Meanwhile, the other conditions all have accuracy scores below 80%, with the distance condition accuracy of 50% being the lowest. In his paper, Bench-Capon concludes that theoretically, one could obtain accuracy levels of around 99% on the multiple fail set, while learning only four of the six conditions. However, the results of the replication demonstrate that the networks do not even learn any of the conditions perfectly when trained with the multiple fail set, while still obtaining high accuracy levels of 98+%. This suggests that the networks might have learned a different–possibly more difficult–pattern from the data.

As seen in Table 3.4, when the networks are trained with the single fail set, the results on the condition test sets have improved considerably to accuracies of 85-96%, indicating that the networks have learned the underlying structure of the data much better.

The same trends can be presented graphically: Figure 3.1a illustrates that when the networks are trained with the multiple fail set, the effect for the distance condition (C6) is a straight line at 1, which likewise occurred in the original study. Figure 3.1b then displays the improved rationale after training with the single fail set, which more closely resembles the plot of a perfect rationale in Figure 3.1c.

3.3 Additional Experiments

Before moving on to the new domain, several additional experiments are implemented on the original domain of Bench-Capon in order to further analyze the problem. As our knowledge of neural networks and our technology has improved considerably over the past decades, it is interesting to assess how the results compare with those achieved using deeper neural networks and more training data. In addition, the models are analyzed using the state-of-the-art XAI technique SHAP (Lundberg and Lee, 2017), and the effect of noise is studied.

- **Neural networks:** Bench-Capon does not discuss how the neural networks were selected and optimized. Thus, a question that could be raised is whether the problem of an incomplete rationale still exists when using a state-of-the-art neural network that is deeper (in terms of both layers and neurons) and optimized for all parameters. This, however, does not turn out to be the solution. When using deeper networks with the same number of hidden layers and training with the multiple fail set, the accuracies on the single fail test set increase slightly to a maximum of around 80%, compared to the 74+% accuracies of the original

Table 3.2: Accuracy scores on the test sets in the original study, when trained with the multiple- and single fail sets respectively (Bench-Capon, 1993)

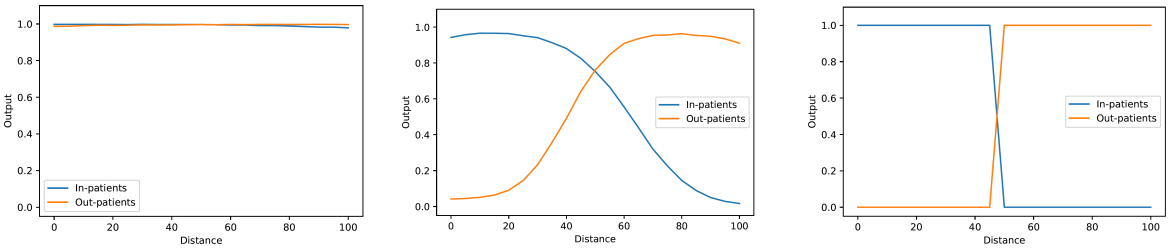
Neural network	Multiple fail	Single fail	Multiple fail	Single fail
1 hidden layer	99.25	72.25	99.25	97.91
2 hidden layers	98.90	76.67	99.0	98.08
3 hidden layers	98.75	74.33	X	X

Table 3.3: Accuracy scores on all test sets, when trained with the multiple fail set

Neural network	Multiple fail	Single fail	Age (C1)	Contributions (C2)	Spouse (C3)	Residence (C4)	Capital (C5)	Distance (C6)
1 hidden layer	98.64	74.21	59.15	76.87	84.48	85.18	78.77	50.53
2 hidden layers	98.27	74.75	63.18	76.58	88.10	88.68	80.74	50.25
3 hidden layers	98.66	74.44	59.22	78.35	86.35	87.023	79.99	50.27

Table 3.4: The accuracy scores on all test sets, when trained with the single fail set

Neural network	Multiple fail	Single fail	Age (C1)	Contributions (C2)	Spouse (C3)	Residence (C4)	Capital (C5)	Distance (C6)
1 hidden layer	96.13	90.24	85.73	92.80	96.05	96.06	88.30	86.14
2 hidden layers	96.16	90.37	85.93	93.01	96.10	96.11	88.40	85.07
3 hidden layers	96.31	89.83	84.86	92.93	95.87	95.90	88.13	81.05



(a) Trained with multiple fail set **(b) Trained with single fail set** **(c) Perfect rationale**

Figure 3.1: Plots of the output on the distance test set (C6) using the best-performing, 3-layered, neural network

networks, but the incomplete rationale problem still clearly exists. When using deeper networks in terms of layers, the networks do not even converge when run on the same data sets.

- **More data:** A well-known solution in the world of machine learning is to use more data, which, with the current technologies, is much easier to implement than at the time of Bench-Capons's original study. Using more data influences the results considerably. As seen in Figure 3.2a, when the amount of data in the multiple fail train set is increased, the performance on the single fail set improves, as accuracies reach about 90%. This indicates an improved, although still incomplete, rationale.

When training with the single fail set, a similar increase in performance on both test sets can be observed. As illustrated in Figure 3.2b, when using 100,000 datapoints in the single fail set, the underlying conditions are learned al-

most perfectly by the networks. However, the same effect, that of the networks obtaining higher accuracies on the multiple fail test set than on the single fail test set, is still visible.

- **XAI techniques:** At the time of the original study, no sophisticated and easily implementable XAI techniques were available. Bench-Capon attempts to explain the predictions by inverting the networks so that the input factors become the output, and he can use the output numbers to study their significance. Inverting the networks revealed that the Boolean features and the five contribution features all have a positive influence, while the capital feature has a strong negative influence, thus confirming some of his findings. The two features necessary for evaluating the distance condition do not appear among the most significant features, but this could occur even when the networks used a perfect rationale, as the distance feature can both negatively and positively impact the decision. Bench-Capon then concludes that using this explainable AI technique to study the rationale of a network would be useful only if one has complete domain knowledge and would know which values are missing or should not be appearing among the most significant features.

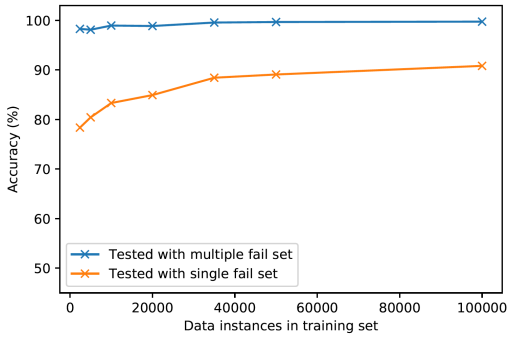
  It is interesting to see how his approach compares with a state-of-the-art XAI technique such as SHAP (Lundberg and Lee, 2017), in which the more sophisticated shapley values somewhat resemble the idea of Bench-Capon's significance features. Figure 3.3a presents a SHAP summary of the three-layered neural network when trained and tested with the multiple fail sets. The SHAP summary reveals that the easier Boolean features and the capital feature have a strong impact on the output of the model. Moreover, the incomplete rationale problem can be clearly observed, as the shapley values indicate that the distance and patient-type features needed for evaluating the distance condition (C6), are not used at all. Even a random noise feature has a larger impact on the decisions than these two features. Figure 3.3b presents the improved rationale when SHAP is used one the three-layered network trained with the single fail set. Now, the distance and patient-type features are used by the network, and the impact of other features more closely corresponds to a correct rationale.

  The SHAP figures thus confirm the results of Bench-Capon, with one addition: By using SHAP, it is confirmed that the distance condition is not considered by the networks. However, the same conclusion remains; namely that these explainable AI techniques are useful only if one has complete domain knowledge and can know whether something is missing.

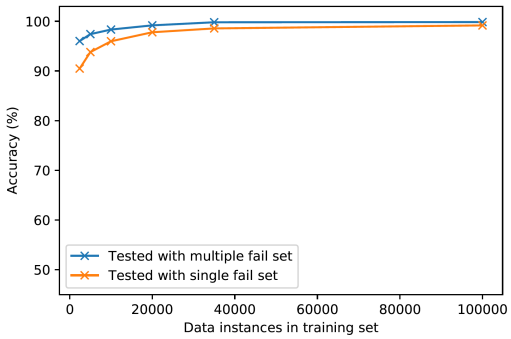
- **Noise:** Bench-Capon includes 52 noise features for each instance in the data sets to assess "whether performance degrades if irrelevant noise factors are included" (Bench-Capon, 1993, p. 292). By inverting the networks, he demonstrates that two noise variables cause "some degree of spurious correlation" (Bench-Capon, 1993, p. 296), but he does not present the exact impact of noise levels on the accuracies. As real data sets almost always include noise, it is interesting to see the exact impact that noise can have on the rationale problem and accuracy scores. As already demonstrated in the replication study of Steging (2018), noise influences the overall performance of the networks in this artificial problem. Both plots in Figure 3.4 confirm this result; the accuracy scores on both test sets degrade slightly when the noise level is increased. However, even when all noise features are eliminated from the data sets, the rationale problem still exists, as the accuracy scores on the single fail test set when trained with the multiple fail set remain around 85% for all three networks. Additionally, when the networks are trained with the single fail set, the results on the single fail test set are still clearly lower than on the multiple fail test set.

4 Child Benefits Domain

The study of Bench-Capon uses a fictional welfare benefit problem, which has since been used in other studies as well (Johnston and Governatori, 2003; Možina et al., 2005; Steging, 2018). As this domain uses specific conditions and data sets,

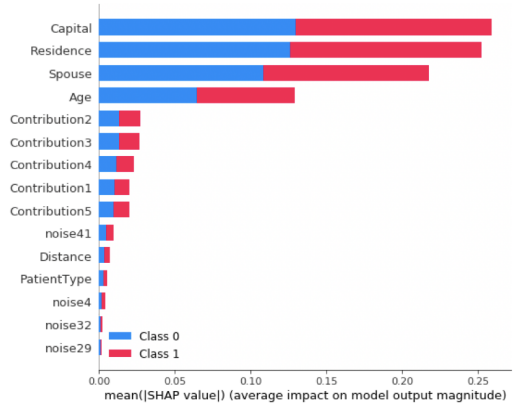


(a) Trained with multiple fail set

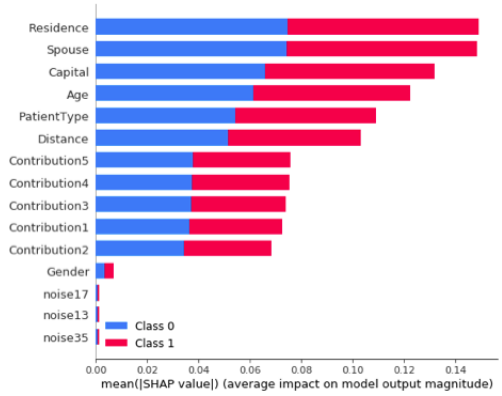


(b) Trained with single fail set

Figure 3.2: The effect of varying the amount of data instances in the training sets on the performance of the networks (averaged for the three neural networks)

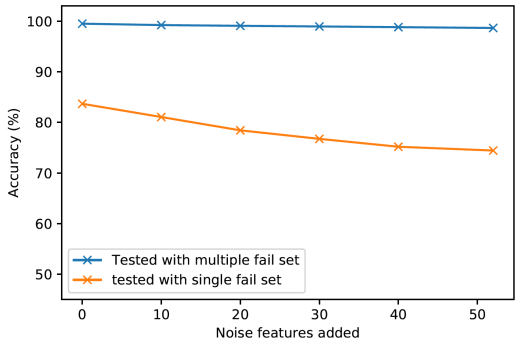


(a) Trained with multiple fail set

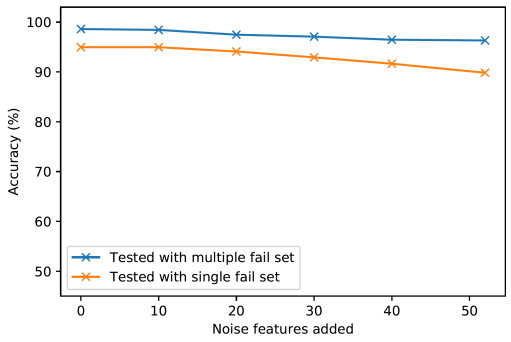


(b) Trained with single fail set

Figure 3.3: SHAP summaries displaying the most important shapley values when testing on the multiple fail set using the best-performing, three-layered, neural network



(a) Trained with multiple fail set



(b) Trained with single fail set

Figure 3.4: The effect of varying the number of noise features in the data sets on the performance of the networks (averaged for the three neural networks)

it is interesting to observe how the results generalize to a domain with different conditions, based on a real-life benefits example. This chapter therefore applies similar methods as those used in the Bench-Capon study to a benefits domain that is based on eligibility for Dutch childcare benefits. This domain has been selected because problems with these childcare benefits resulted in the Dutch "benefits scandal." As a complex self-learning risk classification algorithm used by the Dutch tax authorities was at the root of the entire scandal, the official investigation reports of this scandal can be used to make informed assumptions about preprocessing techniques, data sets, and specific features. Combining these assumptions with real conditions allows us to study the rationale of neural networks in a more real-life setting.

This chapter begins by discussing the new childcare benefits conditions, after which the methods are discussed, followed by the results and a discussion.

4.1 Domain

Childcare benefits are a financial contribution from the Dutch government that aids parents in covering the high costs of childcare. This allowance has existed since 2005. The most important conditions for receiving childcare benefits are outlined on the site of the Dutch Ministry of Finance (Ministerie van Financiën, nd). Using several simplifications (discussed below) and disregarding highly specific situations such as co-parenting or adoption, a person's eligibility for childcare benefits is determined by the following six conditions, all of which need to be satisfied for the person to be eligible for the benefits:

C1. The person should have Dutch nationality or a valid residence permit;

C2. The person should receive (general) child benefits;

C3. The person's child should be registered at a registered childcare center;

C4. The person should have a written agreement with the childcare center, specifying the number of daycare hours (which must be higher than or equal to 12 hours per week) and the hourly rate of the center (must be higher than 10 euros per hour);

C5. The person should work, study, follow a trajectory to find work, or follow an integration course. The same also applies to the person's partner (if applicable);

C6. The person's (or collective income) should be below 100,000 with one child and below 200,000 with multiple children.

$$(C1 \wedge C2 \wedge C3 \wedge C4 \wedge C5 \wedge C6) \rightarrow \text{Eligible}$$

Two conditions have been simplified here: When applying for the actual benefit, the number of daycare hours and hourly rate (C3) do not use a threshold for eligibility but instead influence the amount of benefits received. This is because parents must pay part of the childcare costs themselves. Similarly, the collective income (C6) does not use a threshold; rather, the specific income and the number of children influence the amount of benefits that a person receives.

4.2 Methods

The six eligibility conditions represent a range of typical conditions, including Booleans, strings, continuous variables, and more difficult conditions whose satisfiability depends on the interaction of multiple variables. Evaluating the six eligibility conditions for childcare benefits requires a total of 15 features. In addition to these 15 features, 85 noise features are added to each instance in the data sets. The amount of noise features is based on the number of features used in the risk-classification model. A letter from the Dutch State Secretary of Finance about the results of an investigation by the Dutch Data Protection Authority states that the model used approximately 100 indicators, of which around 20 were significant enough to be used in the risk assessment (van Huffelen, 2019). The report of the Data Protection Authority also states that indicators such as a person's nationality and the number of children were among the indicators used by the model, and these two features are also used in this domain (Autoriteit Persoonsgegevens, 2020, p. 34). The exact features and their range of values are presented in Table 4.1. Appendix B.1 provides a logical representation of the six eligibility conditions, using the features from Table 4.1.

Multiple and single fail sets are once again used to train and test the networks. These sets are gener-

Table 4.1: The features used in the data sets of the childcare benefits domain

Feature	Range
Nationality	194 strings
Residence permit	True/False
Child benefits	True/False
Registered centre	True/False
Daycare hours	0-30
Hourly rate	0-10
Work	True/False
Work hours	0-40
Study	True/False
Partner	True/False
Work partner	True/False
Work hours partner	True/False
Study partner	True/False
Income	0-500,000
Kids	1-8
Noise features: 85	0-100

ated in the same way as specified in Section 2.2, so that no specific conditions are over represented in the data sets. What differs from the replication study is that the training data sets are now comprised of 28,000 cases and the test sets of 2,000 cases (compared with 2,400 and 2,000, respectively, in the original study). These values are selected because the risk-classification algorithm was also trained and tested with a total of 30,000 cases (van Huffelen, 2019). Special test sets are again used for testing whether each of the six conditions has been learned. These test sets use 10,000 cases to create smoother graphs for the results section. When generating the instances for the data sets, the feature "Work hours" is set to false if the feature "Work" is False. Similarly the features "Study partner" and "Work hours partner" are set to false when the "Partner" feature is False and the feature "Residence permit," is set to false when the "Nationality" feature is set to Dutch. Table B.1 in Appendix B2 precisely outlines the distributions and number of cases in the data sets.

The same neural networks are used as discussed in Section 3.1. This is because the investigation reports reveal no details about the learning algorithms that were used in the benefits scandal, and this makes it easier to compare the results to those of the welfare benefit domain. Furthermore, tests with deeper networks, in terms of both layers and neurons, again revealed no real improvements in the accuracy scores of this new domain. Appendix B3 outlines the exact settings of the neural networks.

4.3 Results and Discussion

Table 4.2 presents the results on all the test sets when the three networks are trained with the multiple fail set, which is assumed to more likely resemble a real data set. The accuracy of around 98.5% with the one-layered neural network indicates that an acceptable performance can be achieved on this domain. The other columns, however, suggest the same problem as in the Bench-Capon study: Although the performance is good, the rationale used by the networks is inadequate. The accuracies on the single fail set have improved slightly, compared with the study of Bench-Capon (+-82% versus +-74.5%) but are around the same level as in the replication study when the amount of training data is increased (Figure 3.2). The Boolean conditions are learned quite well, with accuracies approaching 100%, but when conditions get more difficult, the performance on the specific test sets decreases. In the two conditions where the eligibility depends on multiple features (C3 and C4), the accuracies for the one-layered are around only 65%, indicating that these conditions are far from correctly learned. The same trend be presented graphically when the condition test sets are used to plot these effects. When comparing the resulting Figures 4.1a and 4.2a to the to the situation of a perfect rationale, as illustrated in 4.1c and 4.2c, the inadequate rationale is clearly visible.

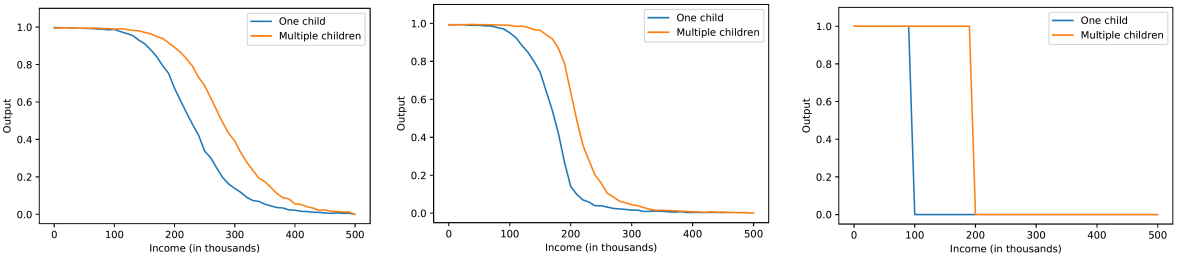
Next, Table 4.3 presents the results when the networks are trained with the single fail set. It is seen that the performance on the multiple fail set remains good, with the one-layered network achieving accuracies of 98%, but that the rationale has improved considerably: with accuracy levels well above 90% for the one- and two-layered networks when tested on the single fail set. The three-layered network does not converge here, as was also the case in the original study of Bench-Capon. The improved rationale can be seen in Figures 4.1b and 4.2b, which more closely resemble the perfect rationale Figures 4.1c and 4.2c.

Table 4.2: Accuracy scores on all test sets, when trained with the multiple fail set

Neural network	Multiple fail	Single fail	Child benefits (C1)	Reg. center (C2)	Nationality (C3)	Hours & rate (C4)	Work/ study (C5)	Income (C6)
1 hidden layer	98.57	82.92	98.12	98.11	62.85	69.21	85.15	82.83
2 hidden layers	97.69	81.58	96.19	96.14	62.64	69.33	83.02	81.17
3 hidden layers	93.29	77.15	91.98	91.71	57.52	65.12	77.40	77.83

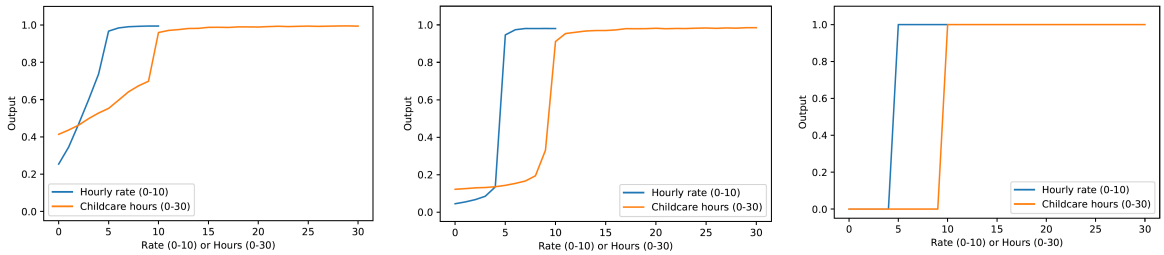
Table 4.3: Accuracy scores on all test sets, when trained with the single fail set

Neural network	Multiple fail	Single fail	Child benefits (C1)	Reg. Center (C2)	Nationality (C3)	Hours & rate (C4)	Work/ study (C5)	Income (C6)
1 hidden layer	98.06	94.87	98.73	98.76	90.93	94.84	95.38	92.18
2 hidden layers	96.53	93.81	97.54	97.57	92.77	92.70	93.79	90.66
3 hidden layers	63.50	61.74	64.42	63.23	60.56	60.72	61.71	60.27



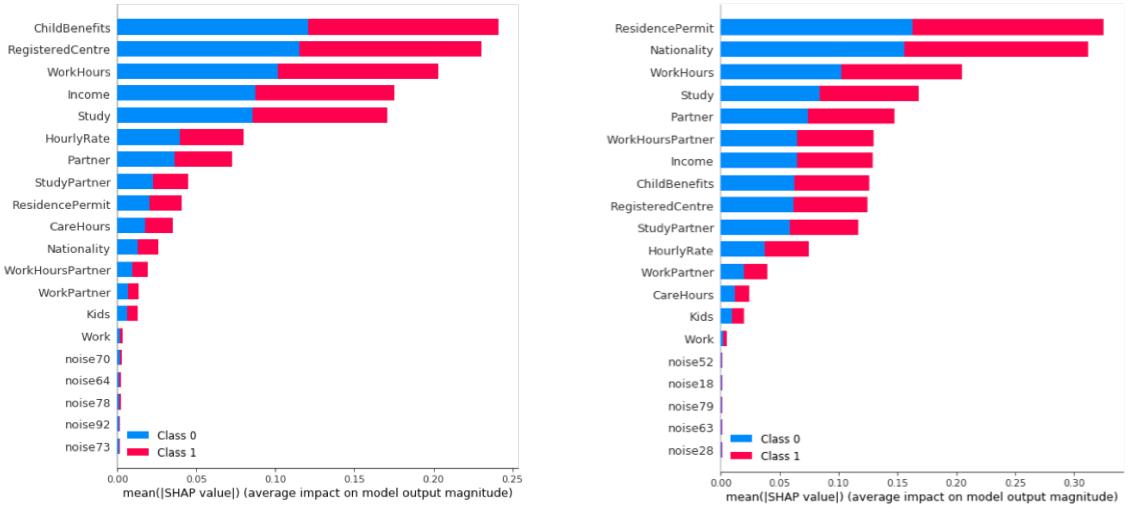
(a) Trained with multiple fail set **(b) Trained with single fail set** **(c) Perfect rationale**

Figure 4.1: Plots of the output on the income test set (C6) using the best-performing, 1-layered, neural network



(a) Trained with multiple fail set **(b) Trained with single fail set** **(c) Perfect rationale**

Figure 4.2: Plots of the output on the hours test set (C4) using the best-performing, one-layered, neural network. (When a case fails on only one of the two features, the output is only averaged into the plot of the feature on which the case fails.)



(a) Trained with multiple fail set

(b) Trained with single fail set

Figure 4.3: SHAP summaries displaying the most important shapley values when testing on the multiple fail set using the best-performing, 1-layered, neural network

4.3.1 SHAP explanations

To further investigate the rationale, SHAP is run on the best-performing, one-hidden-layer, neural network. Figure 4.3 presents the resulting SHAP summaries when training with both data sets and testing on the multiple fail set. It can be seen that when the network is trained with the multiple fail set, the simpler Boolean and threshold features have the largest impact on the output decisions and the features evaluating the more difficult, interacting, conditions have a much smaller impact; a result similar to that of the welfare benefits domain. Moreover, the network considers each of the non-noise features in its decisions, which is a result that one would wish to see. This differs from the SHAP result of the welfare benefits domain, where the features evaluating the distance condition were not considered by the network when trained with the multiple fail set (Figure 3.3a). However, if one has insufficient domain knowledge, these explanations might cause one to more easily accept an unsound network, as the problem of the inadequate rationale is not clearly observable in the explanations here.

When comparing this SHAP summary to that of a network with an improved rationale (trained with the single fail set), it is seen that the same features are considered but that the impact of each feature on the decisions has changed. The lower impact of the noise features now displays signs of an improved rationale, but the rationale is still not perfect, as, for example, the feature of the number of daycare hours and the feature for work are not really used, yet they are needed to evaluate the satisfiability of conditions C3 and C5 respectively. Moreover, the features nationality and residence permit, required for evaluating condition C3, have a considerably larger impact on the output than the other, evenly important, features. This result again differs from the situation in the welfare benefits domain, where after training with the single fail set, most features had an evenly large impact on the output. However, the welfare benefits domain did not include a categorical string feature such as the nationality feature, which could explain this result.

5 Conclusion

In this study, two artificial domains were used to investigate the rationale of neural networks. The results in both domains demonstrate that in using complex AI techniques, such as neural networks, strong performance in terms of high accuracy scores on test data does not necessarily correspond to the networks using a correct rationale. In the replica-

tion of the study of Bench-Capon (1993), this is confirmed by using an artificial domain with six interacting conditions. By using test data sets for each condition, it is shown that the problem is even greater than initially believed, as accuracy scores of 99% are achieved on the test data, but the networks do not even learn any of the six conditions correctly and almost completely disregard two of the six conditions. Additional experiments on the same domain reveal that commonly used solutions, such as deeper networks, more training data, or decreasing noise levels can slightly improve the rationale, but do not fix the problem. With a completely new domain based on real-life conditions, in addition to features and methods based on real-life settings, we see exactly the same effects occurring.

The implications of these results can be severe: If such complex networks are used in a real-life setting, high accuracies on test data might lead to an acceptance of the network, but if the underlying rationale of the networks is actually incomplete or incorrect, new and possibly different data can result in completely wrong classifications, with all its consequences. Plus, both of the domains used in this study are relatively small, consisting of only six interacting conditions and 12-15 relevant features. The problem of an incomplete or incorrect rationale could be amplified when using larger or more difficult domains. Moreover, the artificial data sets used in both domains are carefully crafted so that no conditions are missing or over-represented in the data sets. In real-life settings, where data sets can have different distributions, the rationale problem could again be amplified. An example of this is the use of adversarial images in image classification tasks, where even slight alterations to pictures, not visible to the human eye, can result in completely different classifications (Yuan et al., 2019).

The methods employed in this study do identify several possibilities for fixing this problem. The most important result is that domain knowledge can help in creating synthetic training data that can be used to improve the rationale of neural networks while maintaining high accuracy scores. In the two domains of this study, this meant using so-called single fail data sets, where each ineligible case would fail on only a single specific condition, compared with the standard data sets consisting of cases in which conditions failed on multiple variables simultaneously. In addition, explainable XAI techniques, such as SHAP, can assist in identifying an incomplete rationale, but again only if one has sufficient domain knowledge to know what is missing or incorrect in the explanations. Otherwise, as demonstrated with the second domain, SHAP explanations can seem to indicate a complete rationale, as all of the required and expected features are used by a network, while the rationale is still inadequate.

In our quest to achieve responsible AI, these results are a step forward in increasing our knowledge of the rationale of complex AI techniques. If we wish to use AI responsibly, it is greatly important that next to our ability to explain the predictions of black-box algorithms, we use techniques that reason with a correct rationale. The results demonstrate that this is not as easy as it appears and that high accuracies are no clear guide in verifying the rationale of neural networks. Also, possible solutions, as identified in this study, require strong, nearly perfect, domain knowledge, yet, complex black-box AI algorithms are not normally used in situations where one has such domain knowledge. This suggests that we must be careful in applying techniques such as neural networks before we can fully understand how and why these effects occur.

The results and possible solutions found in this study are limited to the specific domains and learning algorithms used here. It would therefore be interesting to use new, larger, domains in combination with different learning algorithms to study whether similar effects occur and whether one can then use domain knowledge to create synthetic data, which, combined with the original data, can help to improve the rationale of the algorithms. In addition, it would be interesting to study these effects with data sets that include forms of implicit biases in the data, as the data sets in this study were carefully crafted as not to include such biases, which is not necessarily standard in machine learning problems.

References

Akata, Z., Balliet, D., De Rijke, M., Dignum, F., Dignum, V., Eiben, G., Fokkens, A., Grossi, D., Hindriks, K., Hoos, H., et al. (2020). A research

agenda for hybrid intelligence: Augmenting human intellect with collaborative, adaptive, responsible, and explainable artificial intelligence. *Computer*, 53(8):18–28.

Autoriteit Persoonsgegevens (2020). De verwerking van de nationaliteit van aanvragers van kinderopvangtoeslag [processing the nationality of applicants for childcare benefits]. `https://autoriteitpersoonsgegevens.nl/sites/default/files/atoms/files/onderzoek_belastingdienst_kinderopvangtoeslag.pdf`.

Bench-Capon, T. (1993). Neural networks and open texture. In *Proceedings of the 4th international conference on Artificial intelligence and law*, pages 292–297.

Gunning, D. (2017). Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), nd Web*, 2(2).

Johnston, B. and Governatori, G. (2003). Induction of defeasible logic theories in the legal domain. In *Proceedings of the 9th international conference on Artificial intelligence and law*, pages 204–213.

Leighton, R. R. and Wieland, A. (1991). The aspirin/migraines software tools, user's manual. *Technical Report MP-91W00050*.

Lundberg, S. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*.

Ministerie van Financiën (n.d.). Kan ik kinderopvangtoeslag krijgen? [can i receive childcare benefits?]. `https://www.belastingdienst.nl/wps/wcm/connect/nl/kinderopvangtoeslag/content/kan-ik-kinderopvangtoeslag-krijgen/`.

Možina, M., Žabkar, J., Bench-Capon, T., and Bratko, I. (2005). Argument based machine learning applied to law. *Artificial Intelligence and Law*, 13(1):53–73.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "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.

Steging, C. (2018). *Explainable AI: On the Reasoning of Symbolic and Connectionist Machine Learning Techniques*. Master's thesis, University Groningen.

Steging, C., Schomaker, L., and Verheij, B. (2019). The xai paradox: Systems that perform well for the wrong reasons. In *BNAIC/BENELEARN*.

Tweede Kamer (2020). Verslag parlementaire ondervragingscommissie kinderopvangtoeslag - ongekend onrecht [report parliamentary questioning childcare benefits]. `https://www.tweedekamer.nl/sites/default/files/atoms/files/20201217_eindverslag_parlementaire_ondervragingscommissie_kinderopvangtoeslag.pdf`.

van Huffelen, A. (2019). Aanbiedingsbrief bij het schriftelijk overleg over de reactie op het rapport van de autoriteit persoonsgegevens. [letter of government]. `https://www.rijksoverheid.nl/documenten/kamerstukken/2020/11/17/bijlage-2-antwoorde-aut-20190804`.

Yuan, X., He, P., Zhu, Q., and Li, X. (2019). Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824.

A Additional information for the welfare benefits domain

Appendix A includes detailed information about the features, data sets and neural networks settings that are used for the replication study of the 1993 Bench-Capon paper (Chapter 3).

A.1 Logical representation of the six conditions

Using the features from Table 3.1, the eligibility for the six conditions of the welfare benefits can be described logically:

- C1: (Gender(x) = Male $\wedge$ Age(x) $\geq$ 65) $\vee$ (Gender(x) = Female $\wedge$ Age(x) $\geq$ 60)
- C2: Contribution1(x) - Contribution5(x): at least 4 out of 5 have to be True
- C3: Spouse(x) = True
- C4: Residence(x) = True
- C5: Capital(x) $\geq$ 3000
- C6: (patientType(x) = in $\wedge$ Distance(x) $\leq$ 50) $\vee$ (patientType(x) = out $\wedge$ Distance(x) $>$ 50)

(C1 $\wedge$ C2 $\wedge$ C3 $\wedge$ C4 $\wedge$ C5 $\wedge$ C6) $\rightarrow$ Eligible

A.2 Data set distributions

Table A.1: Outline of the specific data sets that are used, and a description of the conditions on which the ineligible cases fail

Data set	Cases	% Ineligible	% Fail on C1	% Fail on C2	% Fail on C3	% Fail on C4	% Fail on C5	% Fail on C6
Multiple fail train	2400	50	34.92	31.71	28.75	28.71	36.83	29.71
Multiple fail test	2000	50	33.65	33.90	29.10	28.70	36.80	29.95
Single fail train	2400	50	8.33	8.33	8.33	8.33	8.33	8.33
Single fail test	2000	50	8.30	8.30	8.40	8.35	8.35	8.3
Age (C1)	10,000	59.65	59.65	0	0	0	0	0
Contributions (C2)	10,000	50	0	50	0	0	0	0
Spouse (C3)	10,000	50	0	0	50	0	0	0
Residence (C4)	10,000	50	0	0	0	50	0	0
Capital (C5)	10,000	50	0	0	0	0	50	0
Distance (C6)	10,000	50	0	0	0	0	0	50

A.3 Neural network settings

Table A.2 and A.3 show the optimal parameters found after an hyperparameter tuning process. As explained in Section 3.1, the number of hidden layers and neurons, the activation function, and mini-batch setting were all fixed during the tuning process. Moreover, the standard solver "adam" is used for all networks.

Table A.2: Parameters used for the neural networks trained with the multiple fail set

Neural network	Neurons	Activation	Batch size	Learning rate	Max. iterations
1 hidden layer	12	logistic	32	0.01	5000
2 hidden layers	24,6	logistic	64	0.005	1000
3 hidden layers	24,12,3	logistic	50	0.01	150

Table A.3: Parameters used for the neural networks trained with the single fail set

Neural network	Neurons	Activation	Batch size	Learning rate	Max. iterations
1 hidden layer	12	logistic	50	0.005	5000
2 hidden layers	24,6	logistic	32	0.01	1000
3 hidden layers	24,12,3	logistic	32	0.01	5000

A.4 GitHub repository

A GitHub repository containing all the code used for the replication study can be found at `https://github.com/BramRijsbosch/Bachelor-Project`.

B Additional information for the childcare benefits domain

Appendix B includes detailed information about the features, data sets and neural networks settings that are used for the experiments with the childcare benefits domain (Chapter 4).

B.1 Logical representation of the six conditions

Using the features from Table 4.1, the six eligibility conditions for the childcare benefits domain can be described logically:

- C1: Nationality(x) = Dutch $\vee$ residencePermit(x) = True
- C2: childcareBenefits(x) = True
- C3: Registered(x) = True
- C4: dayCareHours(x) >= 10 $\wedge$ hourlyRate(x) >= 5
- C5: (Work(x) = True $\wedge$ workHours >= 15) $\vee$ Study(x) = True) $\wedge$ (IF Partner(x) = True $\rightarrow$ ((workPartner(x) = True $\wedge$ workHoursPartner >= 15) $\vee$ studyPartner(x) = True))
- C6: (Children(x) = 1 $\wedge$ Income(x) <= 100,000) $\vee$ (Children(x) $\geq$ 2 $\wedge$ Income(x) < 200,000)

(C1 $\wedge$ C2 $\wedge$ C3 $\wedge$ C4 $\wedge$ C5 $\wedge$ C6) $\rightarrow$ Eligible

B.2 Data set distributions

Table B.1: Outline of the specific data sets that are used, and a description of the conditions on which the ineligible cases fail

Data set	Cases	% Ineligible	% Fail on C1	% Fail on C2	% Fail on C3	% Fail on C4	% Fail on C5	% Fail on C6
Multiple fail train	28,000	50	29.25	29.10	18.43	34.42	26.42	37.55
Multiple fail test	2000	50	29.75	29.50	19.15	34.95	26.35	38.75
Single fail train	28,000	50	8.33	8.33	8.33	8.33	8.33	8.33
Single fail test	2000	50	8.35	8.30	8.35	8.3	8.35	8.35
Child benefits (C1)	10,000	50	50	0	0	0	0	0
Registered center (C2)	10,000	50	0	50	0	0	0	0
Nationality (C3)	10,000	50	0	0	50	0	0	0
Hours & hourly rate (C4)	10,000	50	0	0	0	50	0	0
Work & study (C5)	10,000	50	0	0	0	0	50	0
Income (C6)	10,000	50	0	0	0	0	0	50

B.3 Neural network settings

Table B.2 and B.3 show the optimal parameters found after an hyperparameter tuning process. The number of neurons and hidden layers were set to the same values as in the Bench-Capon Replication Study and thus fixed during the tuning process. Plus, since the activation function and batching settings made little difference on the results, both these parameters were also set to the same settings as used in the replication study. This time, however, the "lbfgs" solver is used for all networks, as the tuning process showed the highest accuracies with this setting.

Table B.2: Parameters used for the neural networks trained with the multiple fail set

Neural network	Neurons	Activation	Solver	Batch size	Learning rate	Max. iterations
1 hidden layer	12	logistic	lbfgs	32	0.01	5000
2 hidden layers	24,6	logistic	lbfgs	50	0.01	1000
3 hidden layers	24,12,3	logistic	lbfgs	50	0.005	1000

Table B.3: Parameters used for the neural networks trained with the single fail set

Neural network	Neurons	Activation	Solver	Batch size	Learning rate	Max. iterations
1 hidden layer	12	logistic	lbfgs	50	0.001	5000
2 hidden layers	24,6	logistic	lbfgs	32	0.01	10,000
3 hidden layers	24,12,3	logistic	lbfgs	100	0.005	5000

B.4 GitHub repository

A GitHub repository containing all the code used for the childcare benefits domain can be found at `https://github.com/BramRijsbosch/Bachelor-Project`.