Interpretability of a hierarchical fuzzy rule-based model for a broad(er) audience

Lynn Pickering[a,c], Victor Ciulei[b], Paul Merkx[b], Jasper van Vliet[b], Bernard De Baets[c], Kelly Cohen[a]

[a]*AI Bio Lab, Digital Futures, University of Cincinnati, Cincinnati, OH, 45221 USA*
[b]*Inspectie Leefomgeving en Transport (ILT), Graadt van Roggeweg 500, Utrecht, Netherlands*
[c]*KERMIT, Department of Data Analysis and Mathematical Modelling, Gent University, Coupure links 653, Gent, Belgium*

Abstract

Machine learning models can provide valuable decision support in many real-world applications. However a model must be interpretable to those using it. This paper explores using post-hoc model interpretability methods in combination with an intrinsically interpretable model design to create a model that is interpretable to both a model designer and a model end user. We train a hierarchical fuzzy system with a genetic algorithm on a real-world ship breaking use case and discuss the performance-interpretability trade-off of the model with respect to a random forest model. Further, we find an interesting pattern using the post-hoc interpretability method SHAP with potential considerations for the future design of hierarchical fuzzy systems.

Keywords: Genetic fuzzy system, fuzzy logic, interpretable ML, artificial intelligence

1. Introduction

1.1. Importance of Interpretability in Machine Learning

Machine learning models can find patterns in large amounts of data that humans would not be capable of finding. These patterns can help us make better, more informed decisions in our world. However, blindly trusting these models can lead to unfairness or wrong decisions. Therefore, the importance of interpretability for machine learning models used as decision support tools is widely agreed upon. Humans who work with decision support tools must be able to understand why a model gives certain predictions, so that they may decide whether to trust the

model's output. Further, legislation in the European Union in the form of the AI Act outlines the interpretability requirements of a model based on the risk level of that model [1].

We define interpretability in a decision support model as a human user of the model being able to understand the model output and how that output was created, by looking at the model itself and the input data as context. There are many ways we can categorize the interpretability of a model. Here we use a framework of *when* in the model the interpretability is incorporated. The two primary *whens* we speak of in this work are understanding the model itself (intrinsic interpretability), and interpreting the model after it has been created (post-hoc interpretability). For a model itself to be interpretable, it must be simple or constrained in such a way that a human may understand it [2]. With a notion such as feature importance we attempt to understand a model after it has been created, because it is too complex for a human to understand as it is. This is often referred to as interpreting a 'black box' model. A further important aspect of interpretability as it is defined here is *who* seeks to benefit from the model interpretabilty. Here we can define several 'audiences': the *model designers*, the *model end users* and *those who the model makes decisions about*. In our application we focus on the *model designers* and the *model end users*, as the model end users will do the explaining to *those who the model makes decisions about*.

1.2. Interpretable Fuzzy Inference Systems

Fuzzy Inference Systems (FIS) lend themselves naturally to being interpretable. Fuzzy sets allow for an encoding of real-world data that better reflects the uncertainty and 'fuzziness' of those data [3]. The linguistic 'if-then' rules are formatted in the same manner that humans reason, and can be written by experts of data sets for smaller input applications. Though these aspects of fuzzy inference systems can lead to interpretability, there are many considerations and constraints that must be placed during the design of such a system to ensure interpretability [4]. Alonso et al. [5] describe the many existing methods for retaining interpretability at each level of fuzzy system design. In Alonso Moral et al. [6], these methods are put into practice for an up to date (at the time which it was published) example of an interpretable fuzzy system. The application is beer classification, and the number of inputs to the problem is three. In many current applications of machine learning, the number of inputs, or the dimensionality of the dataset, is far greater than three, including the application in this paper.

High-dimensional applications cause an explosion of rules for FISes. A Hierarchical Fuzzy System (HFS) was used in as early as 1999 [7] to reduce the curse of

dimensionality for applications with a large number of inputs. More recently, the focus when building an HFS has shifted to retaining the interpretability of a FIS, while also reducing the number of rules and therefore the FIS complexity [8, 9]. Most relevantly, Magdalena states that *"The use of hierarchical fuzzy systems will only produce an effective interpretability improvement when the design of the hierarchical structure was driven by the semantics of the intermediate variables"* [10]. In other words, the design of the hierarchical structure must have relevance to the application, and the blocks that make up the structure must be meaningful to one who understands the data.

In the present work we seek to follow these interpretability constraints so that the decisions of the model are clear to a user. However, these decisions are locally interpretable in that they can be understood one decision at a time. The other dimension of interpretability is global interpretability - understanding the model as a whole. This is not as straightforward for a fuzzy system. Alonso et al. [11] introduce fuzzy inference-grams to graphically visualize the interaction between rules in a fuzzy system to, for example, find the most significant rules in a system. The most important rules can give us a global understanding of the system; however, we may still not gain insight into the most important features in the FIS. An understanding of the most important features is important for multiple reasons. First, the model designer may remove unimportant features from the dataset and redesign a model that is more interpretable because it is simpler with less features. Second, a model designer may work with an expert model end user to check if the features identified as most important make sense in the context of the data. If not, the model may need to be redesigned. In those cases where we see notions from fuzzy set theory used to find feature importances, it is in the context of aggregating the results of several feature importance methods [12]. We have not seen feature importance calculated for the features of a fuzzy model. Therefore, we apply a post-hoc interpretability method to our FIS for the purpose of finding the importance of the features in our dataset to the model for making predictions.

1.3. Contributions and Plan of the Paper

In this research we work towards creating an interpretable model, and then apply post-hoc interpretability methods on that model. We build a hierarchical fuzzy system that is trained by a genetic algorithm (GA) in a real-world engineering setting for the purposes of interpreting and explaining that model, for both a *model designer* and a *model end user*. Previous work on the data has produced results useful to a *model designer*, but too complex to be useful for a *model end user*. In Section 2 we discuss this previous work, introduce the background of the

application and the organization that provides the real-world data. In Section 3 we describe the design of the fuzzy system, and how it is trained. We show how we constrain the FIS at every step of model development and training to build a model that is as interpretable as possible. In Section 4 we analyze the performance of the trained fuzzy system and compare it to the state-of-the-art performance on this dataset, as well as a decision tree model. In Section 5 we analyze and compare the interpretability of the fuzzy model and the other two models by looking at the models themselves. Then we analyze and compare the local and global explainability of the three models using SHAP (SHapley Additive exPlanations), a feature importance method, to interpret the model post-hoc. Finally, in Section 6 we discuss two challenges to the FIS model trained in this work.

We find that applying SHAP gives us further insight into how our trained fuzzy model makes decisions, especially globally where the fuzzy rules cannot help us. Thus, our unique contributions are (a) a local visualization of the hierarchical fuzzy system to support the fuzzy rules and to understand and interpret the model results. This visualization is useful for both the *model designer* and the *model end user*. (b) Applying SHAP to a fuzzy logic system to further support an understanding of the model for the purpose of model design.

2. Background & Previous Work

In this section we take a look at the data used in this work, as well as the organization that provides that data. We briefly discuss previous models built on this data, as well as previous work done in the organization around interpretability, explainability, and working towards creating machine learning tools that work with human users.

2.1. ILT

This study was conducted in collaboration with the Innovation and Data Lab of the Human Environment and Transport Inspectorate in the Netherlands ("Inspectie Leefomgeving en Transport" in Dutch, abbreviated as "ILT"). ILT is part of the Ministry of Infrastructure and Water Management, and works at improving safety, confidence, and sustainability in regard to transport, infrastructure, environment, and housing [13]. ILT is mandated with the task of verifying compliance with Dutch law and prosecuting any violations. Due to the large number of organizations under supervision and limited inspection capacity, not all organizations can be

inspected. This is why ILT has started to use machine learning models to prioritize inspections effectively.

2.1.1. Previous work by ILT on predicting (open-beach) shipbreaking

One particular focus area for ILT's application of machine learning is the problem of open-beach shipbreaking. After 20 to 30 years, most commercial ships reach their end-of-life stage and need to be scrapped. European law mandates that ship owners must recycle their ships at approved recycling facilities authorized by the European Commission (Regulation 1257/2013). Nonetheless, many European end-of-life ships are scrapped under harmful and hazardous conditions on open beaches in Bangladesh, India, and Pakistan [14].

In the Netherlands, the ILT is the responsible authority for enforcing the regulations for shipbreaking. To assist inspectors in this task, the Innovation and Data Lab of the ILT developed two random forest models to (1) predict whether a ship will soon be scrapped, and (2) whether this will be done on a South-Asian beach. The work here focuses only on the first goal, predicting whether a ship will soon be scrapped.

By utilizing the predictions of a machine learning model, ILT inspectors can contact shipping companies before ships are actually scrapped. Shipping companies will be informed about the ship recycling regulations, and will learn that their actions are being monitored. In this way, ILT aims to prevent open-beach shipbreaking for Dutch ships. When doing these preventive inspections, inspectors need to be able to explain - to the shipping company - the reasons why this specific shipping company is being targeted. This is why not only good model performance, but also good interpretability of the model is a requirement for using the model in practice.

2.1.2. Previous work by ILT on interpretability/explainability

The Dutch Government strives towards achieving a high transparency level about its decisions towards the Dutch society [15], and as a part of the Public administration, the ILT does so, too. These efforts are further justified by public debates about the deployment of low-quality algorithms, which lead to biased outcomes affecting large numbers of individuals (e.g., Hadwick [16]). Therefore, it is of utmost importance to ensure both the transparency of the design process of an AI system [17] and of the outcomes of said AI system.

In an effort to integrate the AI model in practice and enhance the effectiveness of the current inspection processes, the IDlab performed several studies, identifying an inclination towards simple text explanation, which were complemented by

simplified visualisations of original SHAP plots. Taken to the extreme, if we only stick to the simple representations that the *model end user* prefers, we do not fully utilize the potential that an ML algorithm brings as a decision support tool.

As we outlined in the introduction, the audiences we focus on are the *model designers, i.e. ILT ML developers,* and the *model end users, i.e. ILT inspectors*. The skills and interests of these two differ, whereby the *model designers* are technically skilled, and interested in details at the global and local level, while the *model end users* have field experience and aim to interpret individual model predictions, on which which they base their trust in the model.

Our current work aims to bridge this gap and provide non-technical users a simple, easy-to-follow interpretation of model outputs comprised of textual and visual representations, with the potential to streamline the adoption of the shipbreaking model in the ILT daily practices.

2.2. Post-Hoc Interpretable Machine Learning Methods

Post-hoc interpretability methods aim at offering insights in the outcomes of a model prediction by various statistical analyses of the conditions through which the data instance has passed. There are a large number of types of interpretability methods available [18], each with their own advantages and disadvantages. For each method, multiple implementation algorithms have been developed [19], each with unique characteristics. One way to classify interpretability methods is by dividing them into local and global techniques. Global feature importance methods highlight the contribution of a feature over the entire dataset, and can account for interactions with other features [20]. Global importance scores are obtained by averaging the local scores.

In contrast, local methods only analyze one data instance, focusing on the feature importances for one prediction. While the global view traditionally gives model developers more insight overall and can serve as a 'sanity check' for how the model considers the data, local interpretability has shown potential for the end-users, unraveling model outputs for decision makers that intend to use machine learning models as decision support tools, as is the case for the ILT.

In choosing the post-hoc interpretability method for this study, we have assessed several methods, such as the Mean Decrease in Impurity (MDI) [21], using the Gini index [22]. However, this method is prone to biasing high cardinality categorical features, and is able to give insights only on the training data, and not on the model's ability to generalize to unseen data. The Mean Decrease in Accuracy (MDA) [23], also known as permutation feature importance, is a good alternative, which does not suffer from the drawbacks of MDI. However, it is skewing feature

importance on highly correlated variables, and has a high computational cost. Finally, SHAP [20] is based on the Shapley values [24] developed in 1963. SHAP, an additive feature importance method that unifies six other interpretability methods, shows better consistency with human intuition and has a few mathematical properties (completeness, symmetry) that make it a robust model-agnostic interpretability method, hence is is also our chosen method of post-hoc interpretability. We use two implementations of SHAP: for the DT and RF models we use Tree SHAP [25] and for the fuzzy model, we use Permutation SHAP. The latter is a model agnostic implementation and which needs a representation of the structure of the data to produce rules about logical feature coalitions, a representation which we build ourselves.

2.3. Dataset

The dataset is provided by the ILT. The following three data sources have been used to create the data:

1. Open data from the NGO Shipbreaking Platform [26]
2. Data from the Global Integrated Shipping Information System (GISIS) [27], only accessible with an official account
3. Data of port calls from the information system of THETIS-EU of the European Maritime Safety Agency (EMSA) [28], only accessible with an official account

The dataset has 15 inputs, as described in Table 1, and a binary output indicating whether or not a ship is at the end of its life. There are far fewer ships that are dismantled each year than continue working, so the dataset is unbalanced in that for every ship that is at the end of its life, there are 9 ships that are not at the end of their life. A level of unbalance this high is often leads to deteriorated performance for a machine learning model [29]. There are solutions to dealing with unbalanced data sets, such as under sampling and over sampling [30]. However we do not make use of these methods, instead choosing to use an appropriate training function for the model that captures the model performance sufficiently on both output classes, and while training on the data as-is.

Table 1: The 15 input features

Input Name	**Description**
GSS Type numeric	type of ship
Age in months	age of ship
GSS Propulsion numeric	type of ship propulsion
GSS Main engines Number of main engines	number of ship main engines
GSS Main engines Max power	maximum power of main engines
GSS Service speed	the service speed of the ship in calm waters
GSS Main engines Model numeric	maximum age in months of this engine model in the training dataset
GSS Main engines Designer numeric	maximum age in months of this engine designer in the training dataset
GSS Main engines Builder code numeric	maximum age in months of this engine builder code in the training dataset
GSS Gross tonnage	volume of the ship
GSS Deadweight	carrying capacity of the ship
GSS TEU	carying capacity relevant to container ships
GSS Insulated capacity	carrying capacity relevant to refrigerated ships
GSS Length between perpendiculars	length of ship between the perpendiculars
GSS Length overall	overall length of ship

3. Part I: Design of a Fuzzy System

A hierarchical fuzzy system is designed to predict if a ship has reached the end of its life. Fuzzy logic is often heralded as a method for creating highly interpretable models. However, for a system trained on a high-dimensional dataset and without taking careful consideration, this interpretability is easily lost. At every step of creating the fuzzy system, interpretability is our primary design goal. To retain the interpretable properties of a fuzzy system, the hierarchy must be designed so that the intermediate features are interpretable within the problem structure [10]. With careful consideration by those familiar with the dataset, the hierarchical fuzzy system is built for the ship breaking application. The system is trained using a genetic algorithm on data provided by the ILT. Other design choices that maintain interpretability are constraining the training of the membership functions.

3.1. Methodology

An overview of the methodology used in this work is given in Figure 1. A genetic algorithm is used to train the rules and membership functions in a hierarchical fuzzy system. The GA uses the real-world dataset introduced in Section 2.3 to train the fuzzy logic model in 3.1.1 and is described further in Section 3.1.2. The dataset is split into five folds in a stratified manner so that the output classes are reflected equally in each fold. The model is trained five times, each time holding a different fold back as the test data set.

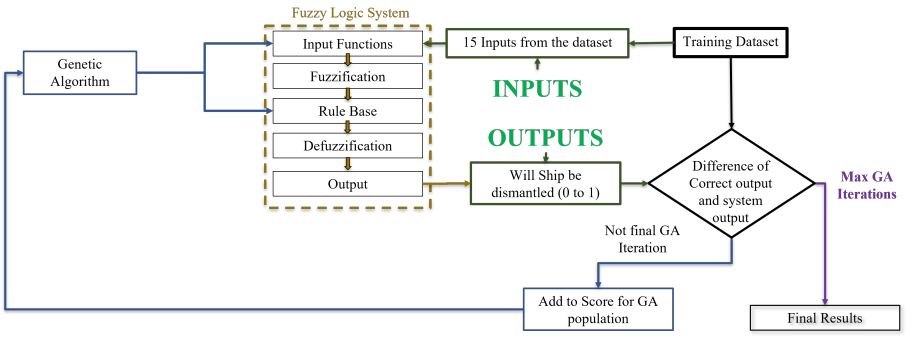


Figure 1: Overview of our approach

3.1.1. Fuzzy Logic System

A fuzzy logic system takes input values, fuzzifies them by assigning membership values in their respective input spaces, and uses a set of if-then rules to

determine the output membership function that is then used to defuzzify the result, leading to a final, crisp output. This is illustrated in Figure 1. The type of fuzzy systems used in this research are Mamdani–Assilian inference models [31, 32]. The method of defuzzification to compute the final output of the system is the center of gravity method [33]. Trapezoidal membership functions are trained by the GA. Only six values, and therefore six genes in the GA, are needed to describe a fuzzy partition of a domain as seen in Figure 2 for an input in this dataset. To simplify things further, the GA trains these six values as floating values between 0 and 100, and before these values create the membership functions they are sorted from smallest to largest, and interpolated to fit the input range. This constraint ensures that the membership functions remain a strong fuzzy partition, and fit the entire input space of an input feature.

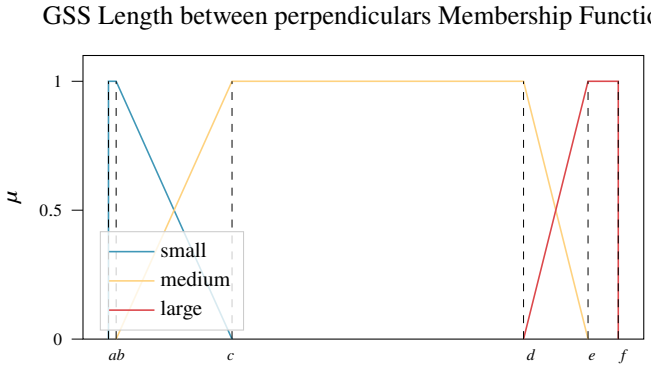


Figure 2: Six values describe a fuzzy partition of a domain. The example is taken from the trained input membership functions for *GSS Length between perpendiculars*

Rule explosion is a big challenge for high input fuzzy systems. The number of rules grows exponentially with the number of inputs. The number of rules to describe a fuzzy system is given as :

$$\text{number of rules} = N_O * (N_{mf})^{N_I}, \tag{1}$$

where N_O is the number of outputs, N_{mf} is the number of membership functions for each input, and N_I is the number of inputs. For the shipbreaking application, with 15 inputs, assuming 3 membership functions for each input, this would give us 14,348,907 rules. Such a large number of rules creates a massive space for the GA to train in, making it difficult to find the optimal solution. Further, and more important to us, it reduces the interpretability of the system. A system with this many rules cannot be considered interpretable, as no human comprehend this many

rules at once. Building a hierarchical structure, breaking the large FIS into smaller input FISs that are connected in layers, allows for a reduction of the number of rules. The highest reduction would result from creating only 2 input intermediate FISs, however, the number of layers in the hierarchy would be at a maximum with this method. In other works, fuzzy hierarchical systems have been created by allowing the GA, or other optimization method, to train the hierarchy itself [34]. While this may allow for a higher accuracy to be achieved, the result is a loss in interpretability, which is the primary motivation in this work for building a fuzzy system. For a hierarchical fuzzy system to retain the interpretable properties of a fuzzy system, the hierarchy must be designed so that the intermediate features are interpretable within the problem structure [10]. The structure was therefore built to reduce the number of rules, while also building intermediate FISs that have a meaning to the dataset and application.

With these guiding principles, the structure of the hierarchical FIS was designed with the expertise of those who had spent time in the field and had knowledge of the data, as well as with some influence from the Random Forest model trained on the data previously. SHAP analysis of this earlier model, see Section 2.1.1, showed that age had double, if not more, importance than other input features. Therefore it remains as an input directly to the final FIS. Ship type is a distinct feature as humans understand it, so this remains ungrouped into intermediate FISs as well. Due to the interpretable design of the system, it is clear in Figure 3 to see why the other input features and intermediate FISs are grouped as they are.

Figure 3 shows the hierarchy designed. The outputs from each FIS in the first layer are the inputs to the next layer of FISs, continuing until the final FIS. The system has 6 intermediate FISes, and one final FIS. The number of rules needed to describe the system is 507. Almost half of those rules are needed for the final FIS, because 4 is large number of inputs, and there are 8 types of ship (GSS Type numeric), requiring a larger number of rules as well. The membership functions that cover the input space, as well as the input and output space of the intermediate FISs require 154 values total to describe them. The input categorical features (GSS Type numeric and GSS Propulsion numeric) are covered with fixed membership functions. The remaining input feature spaces are covered with three trained membership functions, and an additional membership function if a no data category is required, such as for Service Speed. The intermediate input and output membership functions are covered with three membership functions as well, but the output of the final FIS has just two membership functions, because the classification problem is binary. With these aspects in place acting as the constraints on the system that allow for interpretability, we encode the system as

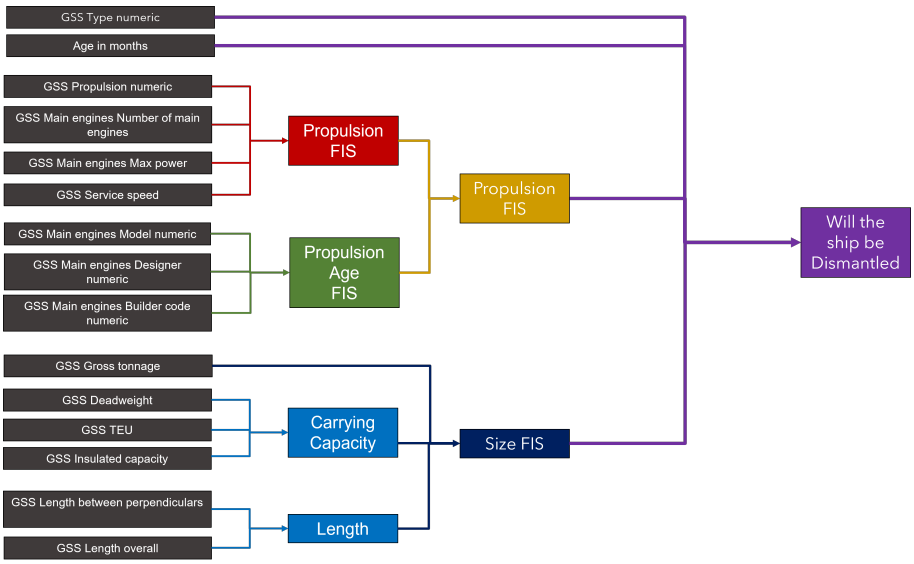


Figure 3: Hierarchical Fuzzy System for the Ship Breaking application

the fitness function for a GA optimization problem.

3.1.2. Training the System

A genetic algorithm [35, 36] is used to optimize the rules and input membership functions that describe the fuzzy logic system (FIS) and is implemented with the guidance of [37]. Optimization methods requiring the derivative of what is being optimized are not feasible, because we cannot calculate the derivative of the fuzzy system. A GA is a simple optimization method that can optimize any fitness function, which in our case, is the FIS itself. For a more detailed explanation of GAs and other optimization algorithms, we refer the reader to [38].

The the number of genes needed to define the fuzzy system, or the length of a chromosome, is 661. The number of chromosomes in the GA population is 40. The percentage of crossover, percentage of mutation, and percentage of elitism are 90, 40, and 10 percent respectively. A maximum number of generations to run the GA for was set at 200, and across the majority of the runs, the GA performance did not increase after 150 generations. One such training curve is given in Figure 4 In large part due to the defuzzification time of the fuzzy system in the fitness evaluation for each chromosome, in each generation, the run time of the GA approached 70 hours.

The unbalanced nature of the dataset makes optimizing the system challenging. A number of methods were used. Initially, the fitness was calculated as the inverse

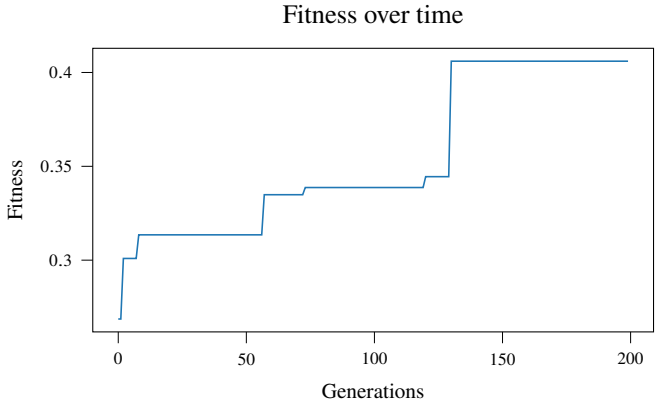
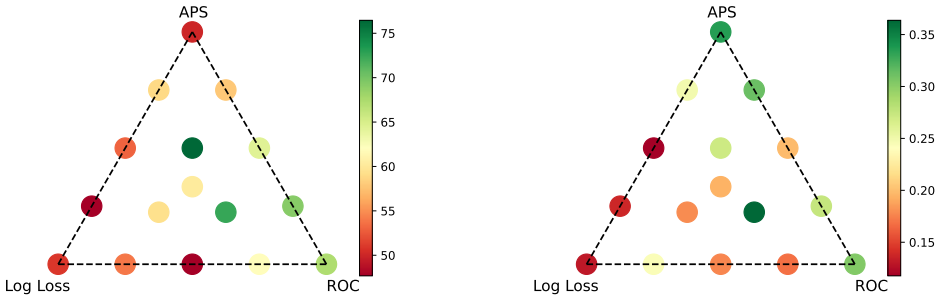


Figure 4: Convergence of the Genetic Algorithm

of the total error of the chromosome, which achieved a high overall accuracy for the model, but failed to classify any points from the smaller class. To perform a systematic test on training performance measures the GA was trained using three measures: *log loss*, *area under the receiver operator curve(AUC)*, and *average precision score(APS)*. Each measure was used as the singular fitness function for the GA, and then in combination with the other measures at various levels. The results are shown in Figure 5. Due to the societal cost of missing a ship that has the potential to be beached, recall is a measure just as important to accuracy to us, if not more, as discussed in more detail in Section 4.3. Therefore, Figure 5a shows the performance achieved by the systems trained on the various measures for an average of accuracy and recall. Figure 5b shows the performance of these systems given by the average precision score as this measure is more robust to imbalanced data compared to the standard AUC score [39]. Further, because our system outputs a prediction between 1 and 0, but we need class labels, and class 1 is more important, the area under the average precision score is the best measure for the success of our system.

Figure 5a gives us secondary measure to how the systems trained by the various fitness functions perform, however we use the average precision score as shown in Figure 5b to chose the best five fitness functions from our test. These five fitness functions are used to train the system on all five folds of the data, rather than just one fold. The precision recall curve of the average results over the folds is given in Figure 6, where the legend denotes [weight of ROC, weight of APS, weight of Log Loss]. The 'best' of these fitness functions is subjective and depends on the performance measure(s) you hope to optimize for. In our case we chose the fitness



(a) Average of Accuracy and Recall

(b) Average Precision Score

Figure 5: Performance evaluation of the GA fitness functions tested

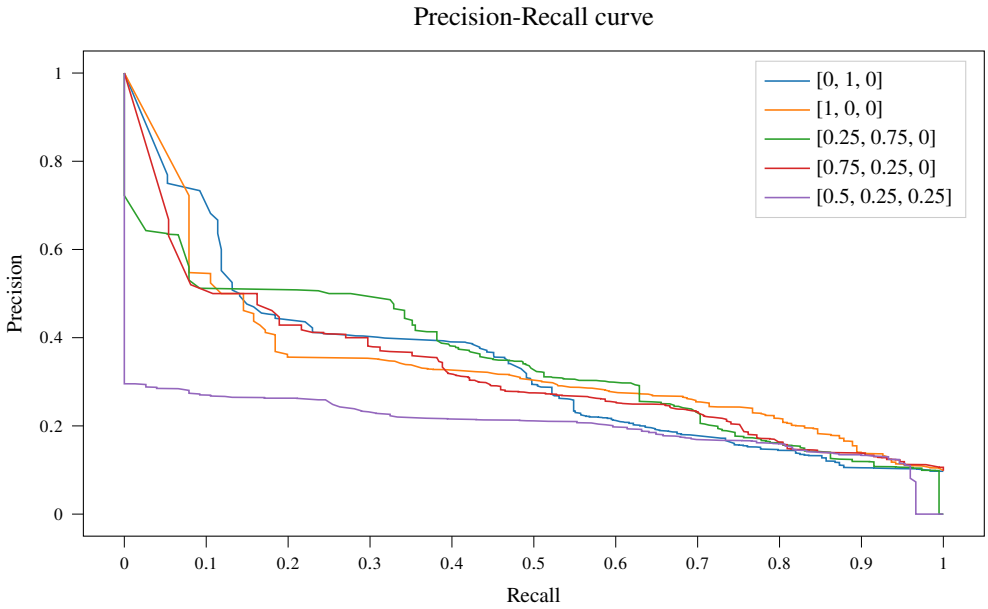


Figure 6: Average over 5 folds: score comparison across best GA fitness functions

function [0, 1, 0] that optimizes only for the average precision score.

4. Part II: Performance of Fuzzy System

It is a popular belief among machine learning researchers that interpretability comes at a cost to performance and accuracy of a system. Though some researchers, such as Rudin, disagree [2]. The performance of the fuzzy logic system is compared to two other models trained on the same data, and discussed. The two models for comparison are a decision tree(DT), and a random forest(RF) model.

4.1. Comparison Models

This section describes the parameters of the models compared to the fuzzy logic system. The RF model is the best performing model trained on this data by the ILT in past work, and is an obvious baseline model for comparison. The DT is an interesting comparison because it is commonly categorized as an intrinsically interpretable model [40]. Therefore, we compare our FIS to a model that has a high accuracy performance, and a model that is categorized as an intrinsically interpretable.

The decision tree is constrained to a depth of three, as this is the depth of the FIS structure. To more accurately represent the performance of a model, we perform nested cross validation, where every time, a different a subset of the data is used to train, and the remaining subset is used as a held-out test set. This is repeated 5 times (folds), with different subsets every time. The performance of the folds are then averaged to avoid any outlier performance due to the choice of the subset. For every subset, a grid-search is performed to find the best hyperparameters. The decision tree classifier has a fixed depth of 3 and uses the gini criterion to measure the quality of a split. The random forest has 500 trees in the forest, a minimum of one sample required to be a leaf node, and considers ten features when looking for the best split. The score used to train these other two models was the same used to train the fuzzy system - the average precision score.

4.2. Results

The average result of each model across the folds is given in Table 2. Accuracy is the number of points overall correctly classified. Recall is the proportion of actual positives correctly identified, or the measure of our model correctly identifying True Positives. For all the ships that are actually dismantled, recall tells us how many we correctly identified as being dismantled. Precision is the proportion of positive identifications actually correct, or the ratio between the True Positives and

Table 2: Average over 5 folds: score comparison across models

Model	**Random Predictor**	**Decision Tree**	**Random Forest**	**FIS**
Accuracy	51.23	90.49	**94.46**	48.68
Recall	47.87	34.42	57.98	**79.23**
Precision	9.69	52.04	**80.20**	14.16
Avg Precision Score	0.39	0.40	**0.77**	0.33
Area under ROC	0.51	0.85	**0.96**	0.73
Area under precision recall curve	0.11	0.44	**0.77**	0.33

all the Positives. For our purposes, that is the number of broken ships that we correctly identify as being broken out of all the ships we identify as being broken.

The average precision recall curves over the five folds are plotted for all models in Figure 7. A random predictor that randomly chooses a floating value between 0 and 1 is included as a baseline, which also illuminates the effect of the imbalanced dataset.

4.3. Results Discussion

For this application, correctly identifying the highest number of dismantled ships is a very important measure. It is better for our model to flag these ships for further review, than to miss them. However, we must balance this goal with the goal of reducing the number of ship inspections due to the reality of having a limited number of ship inspectors. To meet both goals we must balance recall vs precision, and therefore this is the metric we highlight in Figure 7, rather than the AUC-ROC curve which is more commonly used.

We give Table 2 and Figure 7 for completeness, but this is a real application, so let us discuss what the results mean in terms of ships inspected. Our dataset contains about 2000 ships, collected over 4 years, which means about 500 ships up for potential inspection in a year. (This data has been reduced, there are more ships than that up for inspection every year). Out of those 500 ships, 50 of them are dismantled in that year. The fuzzy model would flag, and therefore recommend us to inspect 280/500 of those ships, and it would catch 40/50 ships that were

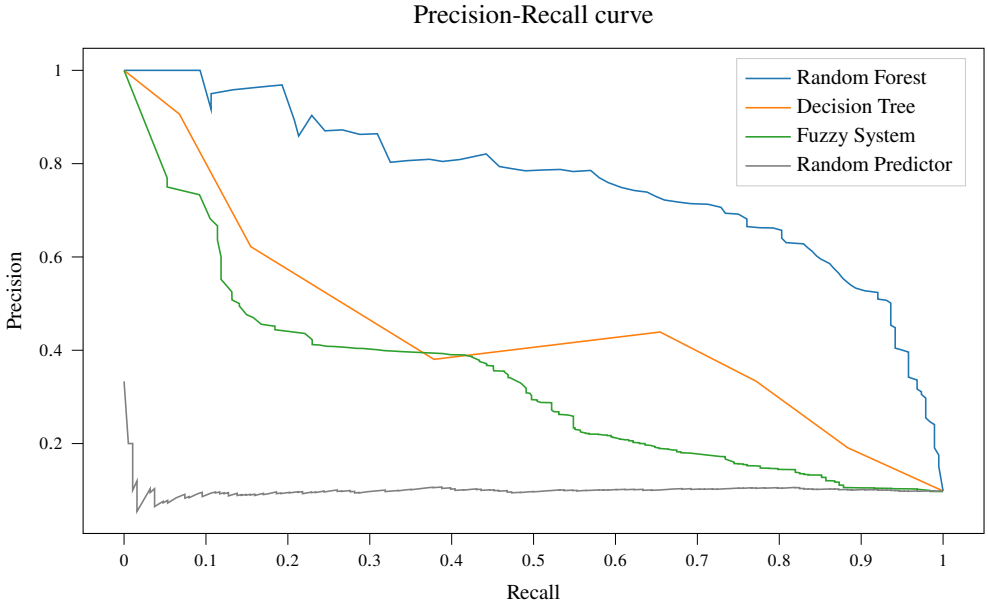


Figure 7: Average precision recall curves for all models

dismantled. The RF would flag 36/500 ships for inspection and would catch 29/50 of the ships that were dismantled. Finally, the DT would flag 33/500 ships for inspection and would catch 17/50 of the ships that were dismantled.

This is the explicit performance trade-off that we ask when we look to use any of these models. If there is the reasonable capacity to inspect 280/500 ships, as the FIS recommends, it is likely worth it to catch those extra dismantled ships. However, if we do not have enough capacity, and want to catch the most amount of ships given out time, then the RF model would be the best model to choose. Performance is not the only important aspect of our models. In Section 5 we analyze the explainability and interpretability of these three models.

5. Part III: Interpretability/Explainability

The primary goal of building a fuzzy model on this dataset was to build a model whose output can be understood by a *non-technical end-user*. While difficult to do without a user study, we attempt in this section to analyze the extent to which that goal was achieved. In addition, we also assess how the interpretability of a FIS compares to the other models in this study: what information can *model designers* and *model end users* extract from these interpretability methods, related

to the 3 chosen algorithms. We examine the RF model because it achieves the best performance when our capacity to perform inspections is low, and the DT model because, it may better meet our primary interpretability goal (with a constrained depth).

In building the FIS, interpretability is taken as the main design goal of each decision. Interpretability can be seen as a set of constraints that allow the final model or model output to be understood by the intended human audience. The constraints taken into account in building the FIS are discussed in the design of the FIS in Section 3.1.1. We also develop a visualization of the fuzzy model to complement the raw rule outputs of the fuzzy system, therefore attaining a superior level of insight compared to any other model in terms of interpretability.

With the purpose of comparing the FIS and the RF, and the constraint imposed by the high amount of features, we turn to SHAP, the state-of-the-art post-hoc interpretability method, as outlined in 2.2. Using SHAP, we can identify which features are most important and in what way. This can support *model end users* in validating/identifying locally, unexpected or potentially false behaviour, or to complement the information they already have. For *model designers* it can provide global feedback about a model after a prediction has been validated, and a coherence check with data being used.

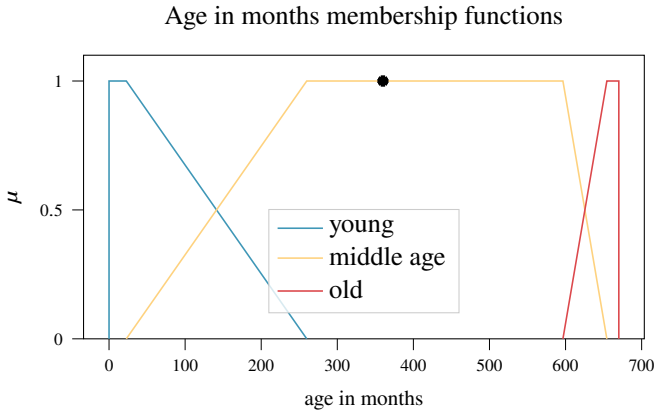
In the rest of the section we first visualize the models themselves, and analyze the level of interpretability and readability without further intervention. We then turn to SHAP to gain first a local and then a global understanding of the models: we use the SHAP waterfall and the summary plots to compare and discuss local and global interpretability aspects of all three models.

5.1. Visualizing the Models

To visualize the models themselves, we chose a data point that is correctly classified as a dismantled ship by all three models. Data instance 44.

5.1.1. The Fuzzy Model

In Figure 9, we display a decision made by the trained fuzzy system for data instance 44. Figure 10 visualizes the same decision, but starts deeper in the tree. Figure 8a gives the trained input membership functions for the input *age in months* as an example. The rules that describe this same visualization with words are given in Table 3. They are broken down by which layer they are in the hierarchical FIS. A total of 7 rules are needed to describe any one decision.



(a) Trained input membership functions for the input *age in months*. The star represents the value for ship age for data instance 44

(b) Standard color scale relating block color to membership function

Figure 8: The supplementary information for the FIS visualization

Table 3: The rules that describe the FIS decision for data point 44

Rules
Input Layer
If GSS Propulsion numeric is type 0 and GSS Main engines Number of main engines is medium and GSS Main engines Max power is medium and GSS Service speed is medium then propulsion power is high
If GSS Main engines Model is middle age parts and GSS Main engines Designer is middle age parts and GSS Main engines Builder code is middle age parts then propulsion age is high
If GSS Deadweight is large and GSS TEU is no data and GSS Insulated capacity is no data then carrying capacity is low
If GSS Length overall is large and GSS Length between perpendiculars is medium then length is low
Layer 1
If GSS Gross tonnage is large and carrying capacity is medium and length is low then size is low
If propulsion power is medium and propulsion age is high then propulsion is low
Final Layer
If GSS Type numeric is type 0 and age in months is middle age and size is low and propulsion is medium then Shipbreaking is high

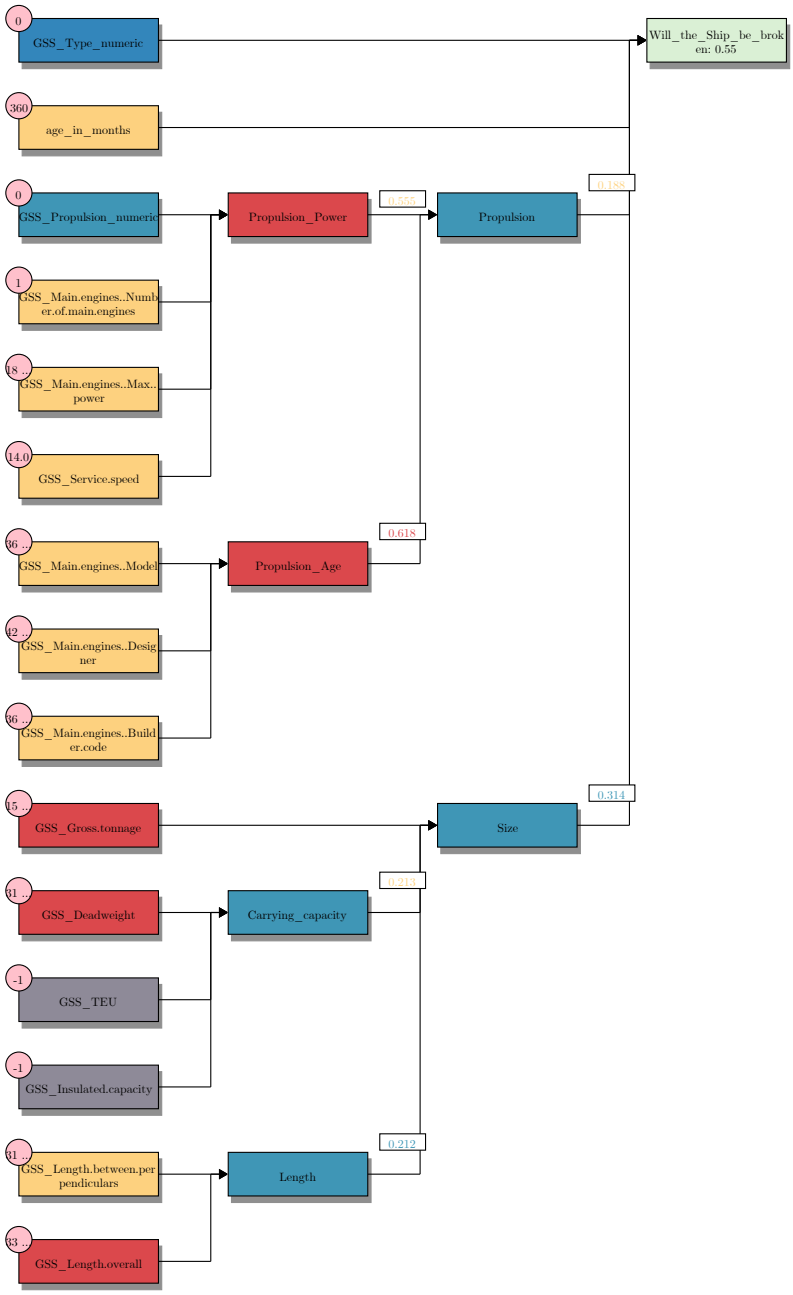


Figure 9: FIS decision visualization for data instance 44

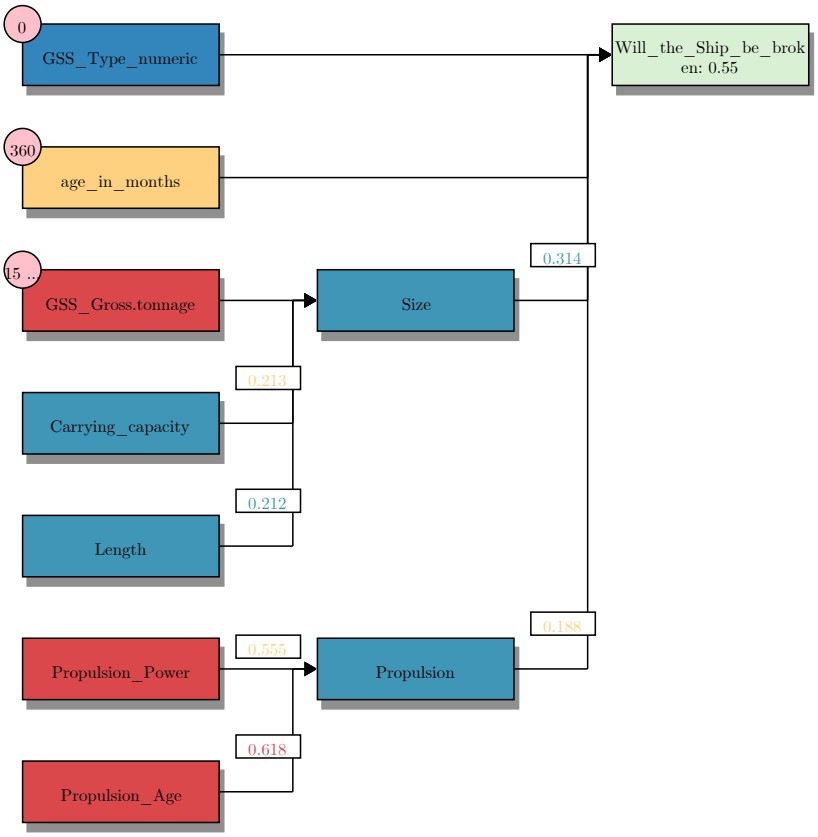


Figure 10: FIS decision visualization for data instance 44 without input layer. The input blocks are colored based on which input membership function is activated the most (low, medium or high), and the pink bubbles on the input blocks contain the actual feature value. The numbers exiting those input blocks represent the output value of the FIS, and are colored based on which input membership function is activated in the next FIS layer. These colors correspond to the color scale as seen in Figure 8b. The final output block is colored green if the prediction is correct, and red if the FIS prediction is incorrect. The darker either color, the more extreme the score is towards one of the two classes.

5.1.2. The Decision Tree Model

The decision tree is shown in Figure 11.

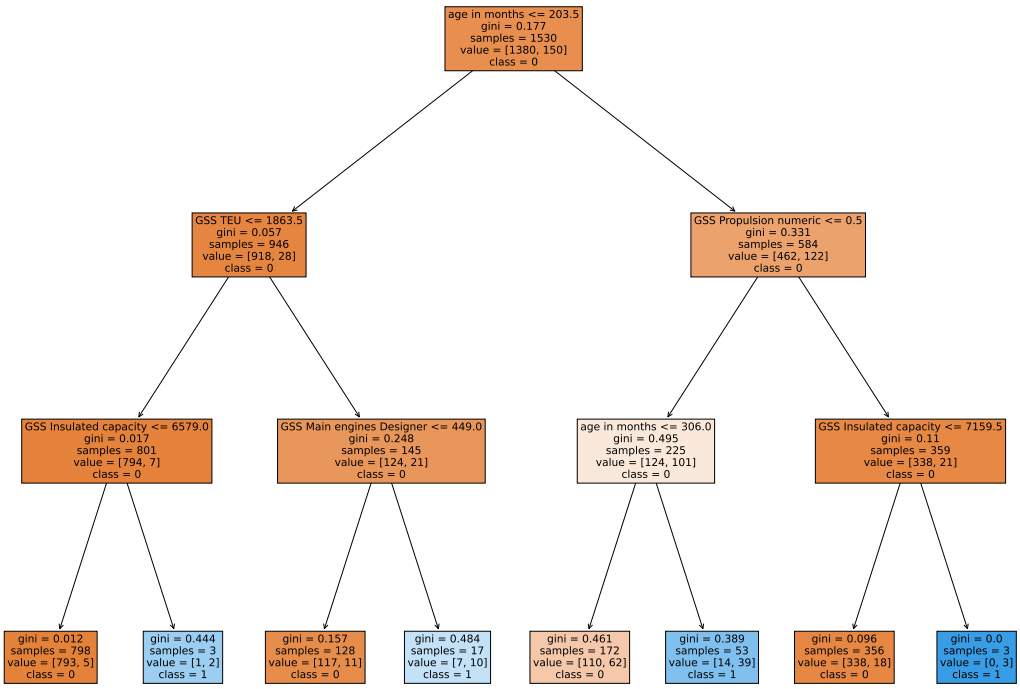


Figure 11: Trained decision tree model graph on one of the folds

The rules that describe the decision tree are created with the help of [41], and shown in Table 4. The rules that activate for the decision made on data instance 44 are highlighted in bold. As to be expected from a decision tree that is limited to a depth of 3, the rule is very simple and does not take many of the input features into account.

5.1.3. The Random Forest Model

A visualization of the random forest model directly would not fit into this work. To interpret the model we will turn to a post-hoc interpretability method: SHAP.

Table 4: The rules that describe the trained decision tree

Rules
if (age in months <= 203.5) and (GSS TEU <= 1863.5) and (GSS Insulated capacity <= 6579.0) then class: 0 (proba: 99.37%) — based on 798 samples
if (age in months > 203.5) and (GSS Propulsion numeric > 0.5) and (GSS Insulated capacity <= 7159.5) then class: 0 (proba: 94.94%) — based on 356 samples
if (age in months > 203.5) and (GSS Propulsion numeric <= 0.5) and (age in months <= 306.0) then class: 0 (proba: 63.95%) — based on 172 samples
if (age in months <= 203.5) and (GSS TEU > 1863.5) and (GSS Main engines Designer <= 449.0) then class: 0 (proba: 91.41%) — based on 128 samples
if (age in months > 203.5) and (GSS Propulsion numeric <= 0.5) and (age in months > 306.0) then class: 1 (proba: 73.58%) — based on 53 samples
if (age in months <= 203.5) and (GSS TEU > 1863.5) and (GSS Main engines Designer > 449.0) then class: 1 (proba: 58.82%) — based on 17 samples
if (age in months > 203.5) and (GSS Propulsion numeric > 0.5) and (GSS Insulated capacity > 7159.5) then class: 1 (proba: 100.0%) — based on 3 samples
if (age in months <= 203.5) and (GSS TEU <= 1863.5) and (GSS Insulated capacity > 6579.0) then class: 1 (proba: 66.67%) — based on 3 samples

5.2. *Local Interpretability*

In Machine Learning, local interpretability represents the possibility of a human audience to understand one single model output. In this application, **is a *model designer* and/or *model end user* able to understand why a certain ship was predicted to be dismantled?**. The SHAP waterfall plot displays a single decision taken by each of the three models of data instance 44 in Figures 12, 13, 14. $E[f(x)]$ is the baseline value, i.e the average output for a model across all its training (background) points. A waterfall plot attempts to explain the difference between this base prediction, $E[f(x)]$, and the output of the model while decomposing the average marginal contribution of each feature on this difference. In comparing this base value across the models, the RF and DT model have similar low values($\approx$ 0.1), while the FIS has a higher value ($\approx$ 0.5).

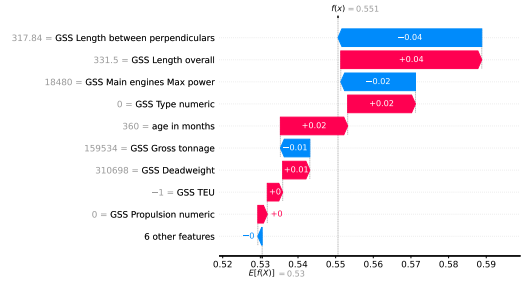


Figure 12: FIS waterfall plot for data instance 44

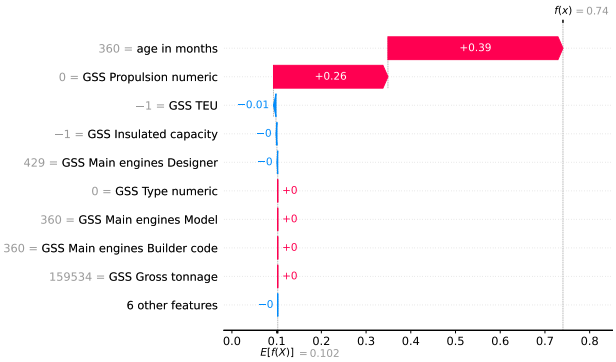


Figure 13: DT waterfall plot for data instance 44

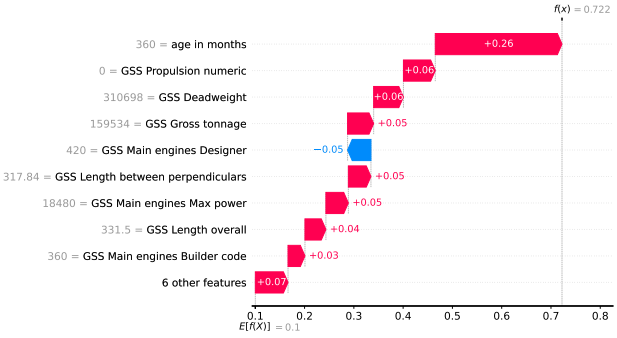


Figure 14: RF waterfall plot for data instance 44

The waterfall plot for the decision tree in Figure 13 is as expected in that only three features contribute significantly to the final prediction, while in the RF model (Figure 14) all of the features have an impact on the final output. The waterfall plot for the FIS, Figure 12 shows 8/15 features as having a significantly impact on this particular decision.

5.2.1. Local Interpretability Discussion

How can this post-hoc local interpretability analysis help *model designers* and/or *model end users* to understand and compare the models? Are we able to achieve an understanding of how the RF model makes its decision, especially with no understanding from the model itself?

In Figure 13 the waterfall plot of the DT indicates exactly what the rules are. It gives more information about the impact of *Age in months* compared to the *GSS Propulsion numeric* and *GSS TEU*. Reading Figure 13 we see that *Age in months* has almost twice the impact on the output compared to *GSS Propulsion numeric*, and that *Age in months* has an almost 40 times greater impact on the output compared to *GSS TEU*. We could have partially inferred some of this information from the rules themselves, yet this visualisation paints a clearer and more informative picture. The waterfall plot of the FIS does give added information about the most important features to the model. One way that a *model designer* may be able to make use of this added model interpretability is to reduce the rules given in Tab. 3 for presentation to the *model end user*. The waterfall plot shows us which input features have little impact and can be removed from the rules, resulting in simple, yet informative rules, shown in in Tab. 5. For this example, according to SHAP, the rules shown account for 97% of the decision made. We can further reduce the rules to any given percentage, depending on the *model end user* preference, such as 84% in Figure 6. In this way, the number of rules is reduced, without simplifying the model itself, and the end-user may be helped, as well.

As the most complex model, a post-hoc interpretation method is offering relatively the most insight for the RF. Looking at the RF waterfall plot in Figure 14, we can see that *age in months* is the most important input for this decision. However, the rest of the features have similar contributions, and thus, a *model designer* has minimal added value for using SHAP waterfall plot.As mentioned above, a *model end user* would likely not look at waterfall plots at all.

Table 5: The rules that describe the FIS decision for data point 44, reduced by SHAP importance in waterfall plot, accounting for 97% of the decision made

Rules
Input Layer
If GSS Deadweight is large and GSS TEU is no data then carrying capacity is low
If GSS Length overall is large and GSS Length between perpendiculars is medium then length is low
Layer 1
If GSS Gross tonnage is large and carrying capacity is medium and length is low then size is low
Final Layer
If GSS Type numeric is type 0 and age in months is middle age and size is low and GSS Main engines Max power is medium then Shipbreaking is high

Table 6: The rules that describe the FIS decision for data point 44, reduced by SHAP importance in waterfall plot, accounting for 85% of the decision made

Rules
Input Layer
If GSS Length overall is large and GSS Length between perpendiculars is medium then length is low
Final Layer
If GSS Type numeric is type 0 and age in months is middle age and length is low and GSS Main engines Max power is medium then Shipbreaking is high

5.3. Global Interpretability

Global interpretability can be viewed as the ability of a target human audience to understand the model as a whole. In this application, **is a target audience able to understand in general why ships are dismantled or not, according to a certain model?** The target audience for global interpretability for this application is primarily the *model designer*, however, the *end user* may be given global interpretability tools when first introduced to a new model as context for how best to use the model. We use SHAP summary plots as our initial global interpretability tool. The Figures 15, 16, 17 give the violin plots for the FIS, DT and RF models respectively. Due to the difference of distribution in the test sets considered for each of the folds during the nested cross-validation, the violin plots for each model vary slightly across the folds described in Section 3.1, so the violin plot is given for the same fold for each model. The remaining folds are given in Appendix A.

5.3.1. Violin Plot Analysis

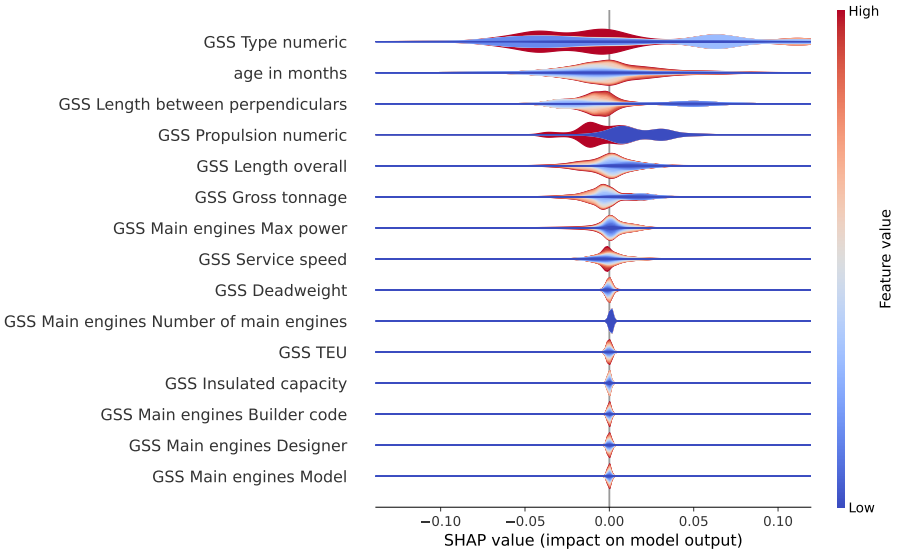


Figure 15: FIS violin plot for fold 0. For each test point assessed, we have the impact of the individual features from SHAP, and the value of the feature from the data set. The violin plot stacks these points, grouping those points together that have the same SHAP value, and coloring the grouped points by the feature value. In this way we have an impression of how each feature affects the model in a global manner.

The plot in Figure 16 shows what we may glean from the rules in Section 5.1.2: only five of the input features impact the output of the model. We can easily see

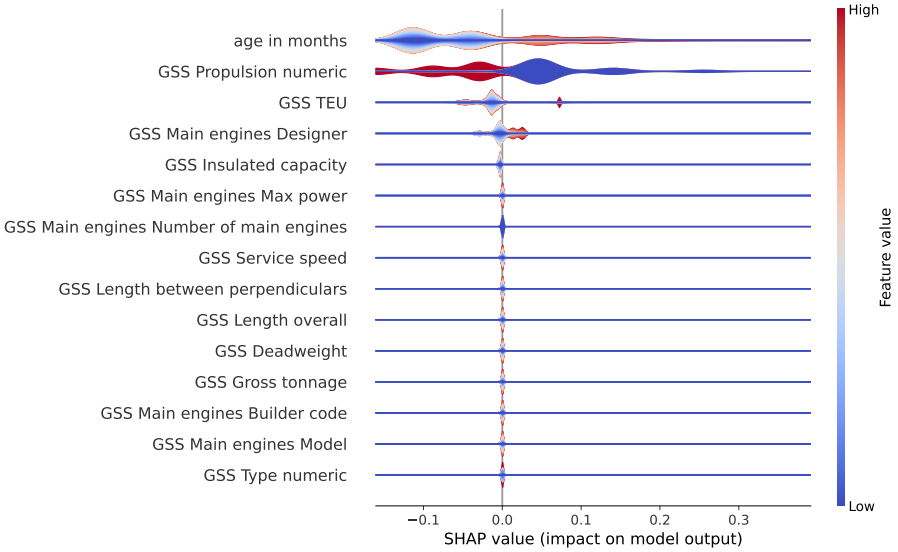


Figure 16: Decision Tree violin plot for fold 0

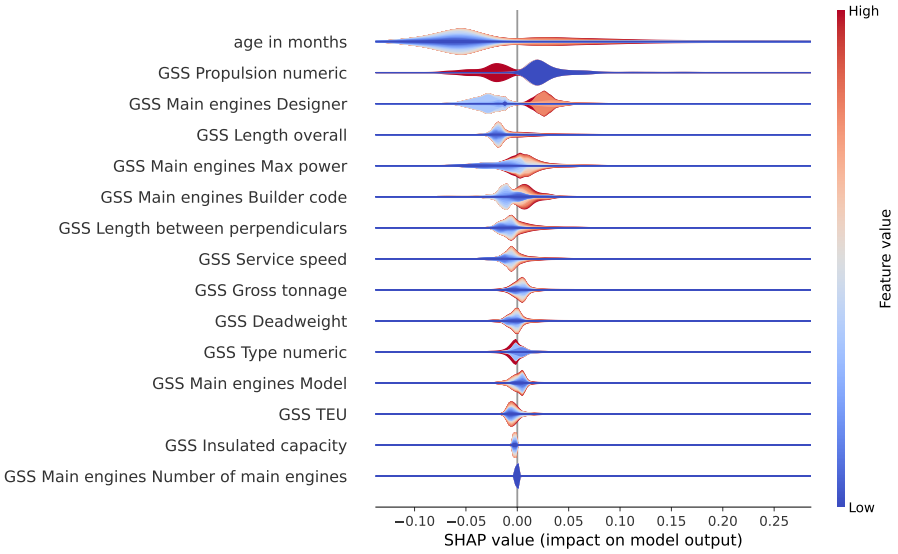


Figure 17: Random Forest violin plot for fold 0

that a higher *age in months* and a lower *GSS propulsion numeric* have an impact in pushing the output to the positive class. The impact of the other three features are not quite as clear. Higher *GSS TEU* and *GSS main engines designer* have an

impact in pushing the output to the positive class for a smaller number of data points, but the full effect cannot be distilled by this visualisation or method alone.

The RF violin plot, in Figure 17 shows that all inputs other than *GSS main engines number of main engines* have a global impact on the output of the model. As shown in Section 5.2, we do not find interpretability in the model itself or the at the local level of the RF model for the *model designer* or the *model end user*. This violin plot gives the *model designer* a first insight into how the model makes decisions on a global level, and what features are the most important.

We turn our analysis to the FIS summary plot. In the FIS violin plot, Figure 15, *GSS propulsion numeric* impacts the model in a similar manner as it does in the DT and RF models, which is expected because this feature is a categorical variable. However, most of the other features do not impact the model in a very clear, distinguishable manner. This shows us that the model does not use those features in a clear manner to discriminate between the two classes. The *model designer* can still glean information about the most important features, and some basic patterns that show how those features impact the model output. We see, for example, a confirmation that a higher age corresponds to a positive SHAP value, but that some lower ages also correspond to a positive SHAP value (the dark blue line through the center of the *age in months* violin bar). Further, *GSS Type numeric* is a categorical feature, so it is logical that there are clear groupings (strong colors together) and that the assigned numeric category does not lend to a clear division between the high and low feature values. However, for the further analysis that the *model designer* will conduct, we turn to a different view of this global interpretability data.

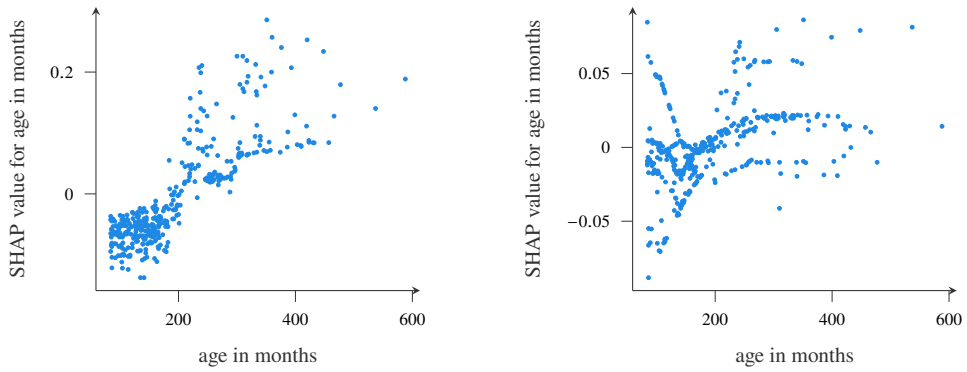
5.3.2. Dependence Plot Analysis

It is interesting to compare the SHAP violin plots of the FIS with those of the RF as *model designers* in order to gain insights into what accounts for the differences in what the models have learned. Especially if we would aim to build a model that identifies very well a few ships for inspection (current RF model performance strength), while still identifying a high number of those dismantled (current FIS model performance strength). To do this we look at the violin plots in Figures 15 and 17, as well as the SHAP dependence plots. The SHAP dependence plot visualizes the value of an input compared to the SHAP value of that input feature, per data point. If there is a trend to be found in the dependence plot, this tells us in more detail the effect of an input on the output of the model.

Both models agree that *age in months* is very important, though the RF model has it as the most important, and the FIS has it as the second most important. A

comparison of the dependence plot of this input feature across the two models, given in Figure 18, illustrates some similarities and differences. Just as for the violin plots, we plot the dependence plots for a single fold, and give the remaining fold dependence plots in Appendix B. For both models, an age of 200 months shows to be an inflection point. For the RF model, ages above 200 months correspond to positive SHAP values, and those below correspond to negative SHAP values. For the FIS, there are various linear-like patterns, a small one of which even decreases in SHAP value as the age approaches 200 months. However, the largest number of data instances that have an age lower than 200 months correspond also to negative SHAP values. Many of the ages above 200 months correspond to slightly positive SHAP values, and small amounts of ages correspond to either higher positive SHAP values or slightly negative SHAP values. We note further that the highest FIS SHAP values do not reach even $1/3$ of the highest SHAP values reached in the RF. We learn that 200 months is a discriminatory age, and that the age-model output relationship is far more complex for the FIS, likely depending on other variables, than the age-model output relationship for the RF. To look at what these other variables might be, we plot the same figures, but color them with the value of a third variable. In this case we use *GSS Propulsion numeric*, because it is a very important feature in each model. In both Figures 19b for the FIS and 19a for the RF we see clear patterns emerging from the interaction of *GSS Propulsion numeric*.

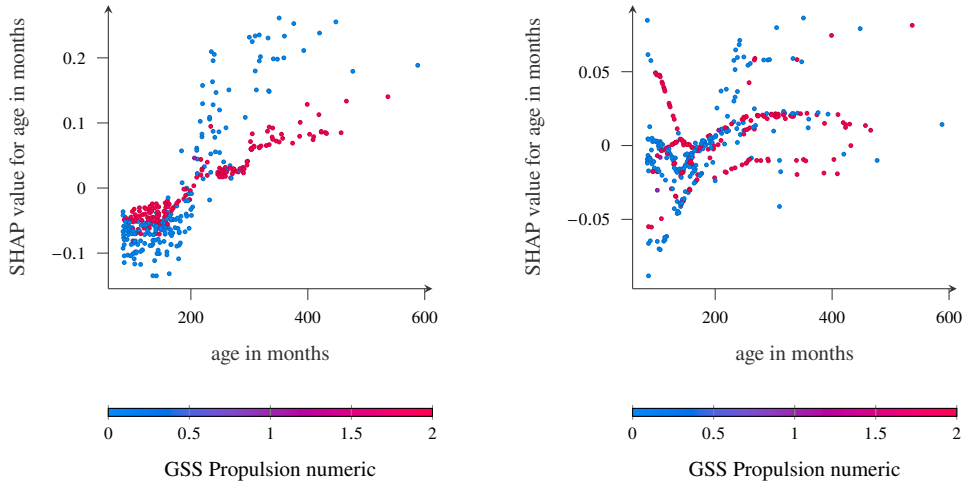
This is an especially useful insight for the *model designers* of the fuzzy system. It appears that when *GSS Propulsion numeric* is category 2, and *age in months* is low, the feature *age in months* has a positive SHAP value. This sounds like a rule, because this is part of rules in the model. If the *model designers* believe this is a pattern that the FIS has falsely learned, we can find this rule in the FIS, and edit the rule to be more in line with how we would expect this model to work. This once again shows the strength of the fuzzy system, and gives an interesting example of how a post-hoc interpetability method such as SHAP can be useful.



(a) RF *age in months* SHAP dependence plot

(b) FIS *age in months* SHAP dependence plot

Figure 18: *Age in months* dependence plot RF - FIS comparison for fold 0

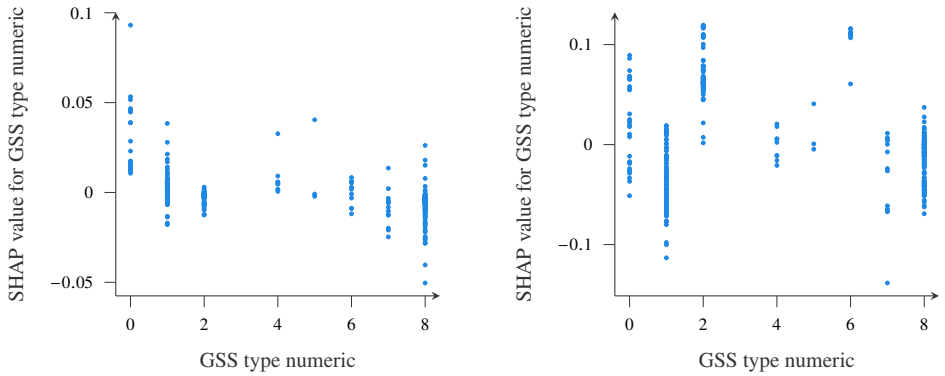


(a) RF *age in months* SHAP dependence plot

(b) FIS *age in months* SHAP dependence plot

Figure 19: *Age in months* dependence plot RF - FIS comparison with *GSS Propulsion numeric* interaction coloring for fold 0

A dependence plot comparison is given in Figure 20, for the input feature *GSS type numeric*, or ship type. We compare this feature because the *GSS type numeric* rank is highest for the FIS, and fifth least important for the RF model. We see in Figure 20 that the scale of SHAP impact for the RF is on average less than half that of the FIS. For the RF model in Figure 20a, most ship types do not push much towards either class, other than type 0 (bulk carrier), which pushes towards class 1. However, for the FIS, $4/8$ ship types push distinctly a class.



(a) RF *GSS type numeric* SHAP dependence plot

(b) FIS *GSS type numeric* SHAP dependence plot

Figure 20: *GSS type numeric* dependence plot RF - FIS comparison for fold 0

GSS Main engines Designer is an input feature that stands out for being of much higher importance for the RF than the FIS, while *GSS Length between perpendiculars* and *GSS Gross tonnage* stand out for being of higher importance for the FIS than the RF. The remaining features not mentioned are more or less in agreement in terms of ordered importance rank to each model.

5.3.3. Global Interpretability Discussion

Overall it appears that the hierarchy chosen for the FIS has some priority over marginal contributions as measured by SHAP. Table 7 gives the most important features as identified by SHAP values, averaged over all the folds. The features are colored to match the FIS that they first enter, matching the FIS structure given in Figure 3. We see that the two inputs that directly enter the final FIS, *GSS type numeric* and *age in months* are the two most important features. Especially noticeable, is that *GSS type numeric* is a very unimportant feature for the RF model. The other feature, *GSS gross tonnage* that passes through only one FIS before entering the final FIS, is the fourth most important feature. We hypothesize that when the features that have a direct connection with the final FIS are removed, an immediate effect on the output is had. However, other features that go through two intermediate FISes first likely have a more diluted effect when 'removed'. (Important, of course, that in this application these other features still do have an effect on the output, and that features grouped in intermediate FISes together are not necessarily grouped together here). The nature of the design is such that these input features are inputs to intermediate FISes, the outputs of which continue as inputs to the next layer, so the impact of the feature may not be as direct. Though

Table 7: The top features for the FIS across all folds

Color in FIS Structure	Ordered Top features	# FISs between input and final FIS
1	GSS Type numeric	**0**
2	age in months	**0**
3	GSS Propulsion numeric	2
4	GSS Gross tonnage	**1**
5	GSS Main engines Max power	2
6	GSS TEU	2
7	GSS Length between perpendiculars	2
8	GSS Length overall	2
9	GSS Service speed	2
10	GSS Deadweight	2
11	GSS Main engines Designer	2
12	GSS Main engines Number of main engines	2
13	GSS Main engines Builder code	2
14	GSS Insulated capacity	2
15	GSS Main engines Model	2

this is just one dataset, and one method of feature importance calculation, and further tests need to be conducted, this may have an important impact on the design of hierarchical fuzzy systems for interpretability in the future. Any person designing such a system will need to account for intermediate FISes that have meaning in the context of the data [10], as well considering that the distance an input has from the final FIS in terms of 'layers' may impact the effect that that input can have on the output.

Now that we have finished our look at global interpretability, we look back on the value added by our SHAP analysis. We find that visualizing the decision by visualizing the FIS itself is much more informative than the SHAP waterfall plot. However, this waterfall plot can be used to minimize the rules and therefore increase the interpretability of the FIS. From the violin summary plot, we cannot gain much relevant insight because the interactions between most of the features and the output are too multi-dimensional. This plot does provide a good starting point for the dependence plots which capture subgroup hetero/homogeneity, the most useful plot for this study. While the dependence plots are not directly useful for *model end users*, the *model designers* could make use of these plots to write some imprecise global rules about the model, or even to edit rules that the model has learned.

6. Discussion

We discuss two relevant aspects, or challenges to the FIS model trained in this work. The first challenge is created by the simplification of the real world data so that a model can be trained on it. The second challenge briefly discussed is the absence of monotonicity with respect to the age of a ship.

6.1. Time Challenge

The obvious challenge to applying the trained fuzzy model created here is the low accuracy, even if that is not our most important performance measure. This is in big part due to the challenges of real world, unbalanced data sets, such as the one we work with in this paper. A random forest model, which is often trained on 'toy' data sets with accuracy's over 99% achieves 94.4% accuracy on this data set, although the more telling value for such an unbalanced data set is the 0.77 average precision score it achieves. To apply machine learning models to real world applications, we must simplify the data. One such simplification made for this data set is to remove the time aspect of the data set and turn the problem into a classification problem. The question of dismantlement of a ship is a value that changes over time. In the current classification form of the problem, we are asking the models to classify at an exact month in time, if a ship is dismantled or not. The month before a ship is dismantled, it is not dismantled. This is clearly a demanding task for these models, on top of the unbalanced nature of the data set.

We perform a test to, in a sense, add the time back into the problem. We copy the ships in the test data sets and change only those inputs that change with time (by the interval months) to a certain interval. The model predicts across this time interval, and the predictions are weighted before being combined into a new final prediction. The weights that gave the best results were a simple average and maximum, though we tested other methods of weighing. Table 8 gives the new results. The maximum time interval tested was 10 years, or +/- 5 years on either side of the ship age. For both models, the best results are shown per weighting method as compared to the original model results. The FIS is able to achieve a higher accuracy and a higher recall by taking the average prediction over a period of 10 (+/- 5) years. The model achieves an even higher recall taking the max over 4 (+/- 2) years, however the accuracy decreases. The RF model also achieves a higher recall when taking the maximum over 10 (+/- 5) years with a hit to accuracy, though it does not achieve the recall of the FIS.

We found that by returning some time aspect into the data set itself, we can achieve a higher performance with both models. This test shows us the importance

Table 8: The time interval test for the FIS and RF

weights	time (+/-) year	accuracy	recall	precision
Fuzzy Inference System				
none	0	49.18	75.99	15.13
average	5	51.17	84.21	15.02
max	2	37.34	94.74	13.14
Random Forest Model				
none	0	93.19	51.35	70.37
average	2	93.46	48.65	75
max	5	87.43	67.57	40.98

of being aware of the limitations we put on our models when we simplify real world data so that we may build models on it. An important factor in building interpretable models as a *model designer* analyzing these limitations and making note of them.

6.2. Monotonicity Challenge

Monotonicity is an interpretability constraint that ensures, for example, as a feature increases(/decreases) the corresponding output is monotone increasing or decreasing. A monotone relationship is not necessarily present. However in the design of interpretable models, monotonicity that obeys structural knowledge of a problem domain is a useful constraint *model designers* must make use of when possible.

A ship inspector would tell us (as the global SHAP analysis did in Section 5.3), that age is a very important factor in whether or not a ship is dismantled. In fact, the relationship between age and dismantlement could be a prime example for the constraint of monotonicity. If all the features of a ship remained the same, but the age of the ship increased, we can expect that the prediction of dismantlement remain the same or increase, but certainly not decrease. But can we expect this? Experts working with the data have seen that ships that reach a certain age are likely to remain in service for a long time. But, if *model end users* would assume a monotone relationship that does not exist in the model, this can create confusion and distrust of the model. Therefore, *model designers* must take care in their introduction and presentation of such models to *model end users*.

7. Contributions and Conclusions

In the Netherlands there are only 20 of ship inspectors and 10s of thousands of ship entering each year. Given that an **interpretable** machine learning model can be a decision tool for ship inspectors to more efficiently chose ships for inspection, we present a hierarchical fuzzy system trained by a GA as a better interpretable model than a Random Forest model. We emphasize the importance of the audience when working towards interpretability, and we focus on two audiences in our work. The ship inspectors, or the *model end users*, are the ultimate target for our interpretable model. However, interpretability aimed at the *model designers* is important in our work as well, as it helps us create that interpretability for the *model end users*. We thoroughly discuss the various performance metrics and trade-offs as they relate to the ship-dismantling application, and find that the FIS flags more ships for inspection, but also finds more of the dismantled ships. We show the interpretability of the HFS using tools built from the model itself, and then turn to a feature importance method SHAP to gain additional interpretability insights and compare these to the RF. We propose a method to use local SHAP to reduce the rules shown to a *model end user* while maintaining full coverage of the input space and without reducing the rules in the model itself. The global SHAP explanations give us model interpretability for the *model designer*. From the global SHAP explanations we see that those features with the most effect on the fuzzy model predictions are the same as the features that directly contribute to the final FIS. In other words, the structure of the chosen hierarchy appears to have a great impact on the impact of the features, according to SHAP.

We can examine and present the trade off between interpretability and performance, however, the final choice of 'best' model application for this real world problem relies on managing decisions within the model interpretability, constraints of population size, available capacity and model performance.

7.1. Future Work

There are many avenues of future work that the authors would like to explore. The first is to expand the FIS to apply it to the ship beaching problem as well, and to achieve a better performance in training the FIS on both the ship breaking problem described in this work, and the new work. This could involve investigations into multi objective fitness functions for the GA in training the fuzzy system. To further the interpretability of the FIS, we could build on this work by eliminating the least important features according to the global SHAP analysis, and iteratively train new models. Fewer inputs make it easier for a human to interpret the output of

the model. We could also investigate rule reduction methods to remove potential redundant, erroneous or conflicting rules [42]. On this subject we could consider using SHAP as a method for rule reduction, in addition to previously used methods. Further, it would be interesting to look into the effect of the architecture of the FIS on the output, and importance of the features. We would apply the same theory to a different dataset, with a similar number inputs, but with a more balanced output division to test this theory.

While we have attempted to comment on the interpretability of the models tested, as well as the local and global explainability, to truly test these aspects of a model, a user study should be conducted. The purpose of any model built for this application would be as a decision support tool for ship inspectors, or those authorities deciding which ships should be inspected. Before such a study were conducted, there would be a focus on presenting data in a non misleading and understandable way, such as an interactive dashboard.

8. Acknowledgments

We would like to specifically thank Antonio Pereira Barata from the ILT for his help with the experimental setup suggestions for assessing model performance
.

While performing this work, in two separate time periods, Lynn Pickering was a recipient of a Fellowship of the Belgian American Educational Foundation and a Fulbright Ghent University Award.

References

[1] C. European Commission, Directorate-General for Communications Networks, Technology, Proposal for a regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts, 2021. URL: https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=celex:52021PC0206#document2.

[2] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature machine intelligence 1 (2019) 206–215.

[3] L. A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353. URL: https://www.sciencedirect.com/science/article/pii/S001999586590241X. doi:10.1016/S0019-9958(65)90241-X.

[4] J. M. Alonso, L. Magdalena, Special issue on interpretable fuzzy systems, Information Sciences 181 (2011) 4331–4339. URL: https://www.sciencedirect.com/science/article/pii/S002002551100315X. doi:https://doi.org/10.1016/j.ins.2011.07.001, special Issue on Interpretable Fuzzy Systems.

[5] J. M. Alonso Moral, C. Castiello, L. Magdalena, C. Mencar, Designing Interpretable Fuzzy Systems, in: J. M. Alonso Moral, C. Castiello, L. Magdalena, C. Mencar (Eds.), Explainable Fuzzy Systems: Paving the Way from Interpretable Fuzzy Systems to Explainable AI Systems, Studies in Computational Intelligence, Springer International Publishing, Cham, 2021, pp. 119–168. URL: https://doi.org/10.1007/978-3-030-71098-9_5. doi:10.1007/978-3-030-71098-9_5.

[6] J. M. Alonso Moral, C. Castiello, L. Magdalena, C. Mencar, Design and Validation of an Explainable Fuzzy Beer Style Classifier, in: J. M. Alonso Moral, C. Castiello, L. Magdalena, C. Mencar (Eds.), Explainable Fuzzy Systems: Paving the Way from Interpretable Fuzzy Systems to Explainable AI Systems, Studies in Computational Intelligence, Springer International Publishing, Cham, 2021, pp. 169–217. URL: https://doi.org/10.1007/978-3-030-71098-9_6. doi:10.1007/978-3-030-71098-9_6.

[7] L.-X. Wang, Analysis and design of hierarchical fuzzy systems, IEEE Transactions on Fuzzy Systems 7 (1999) 617–624. doi:10.1109/91.797984, conference Name: IEEE Transactions on Fuzzy Systems.

[8] Y. Zhang, H. Ishibuchi, S. Wang, Deep takagi–sugeno–kang fuzzy classifier with shared linguistic fuzzy rules, IEEE Transactions on Fuzzy Systems 26 (2018) 1535–1549. doi:10.1109/TFUZZ.2017.2729507.

[9] T. R. Razak, J. M. Garibaldi, C. Wagner, A. Pourabdollah, D. Soria, Interpretability indices for hierarchical fuzzy systems, in: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2017, pp. 1–6. doi:10.1109/FUZZ-IEEE.2017.8015616.

[10] L. Magdalena, Designing interpretable Hierarchical Fuzzy Systems, in: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2018, pp. 1–8. doi:10.1109/FUZZ-IEEE.2018.8491452.

[11] J. Alonso, O. Cordon, A. Quirin, L. Magdalena, Analyzing interpretability of fuzzy rule-based systems by means of fuzzy inference-grams, in: World Congress on Soft Computing, 2011, pp. 181–185.

[12] C. Kokkotis, C. Ntakolia, S. Moustakidis, G. Giakas, D. Tsaopoulos, Explainable machine learning for knee osteoarthritis diagnosis based on a novel fuzzy feature selection methodology, Physical and Engineering Sciences in Medicine 45 (2022) 219–229.

[13] M. of Infrastructure, W. Management, About the ilt, 2023. URL: https://english.ilent.nl/about-the-ilt.

[14] S. Barua, I. M. Rahman, M. M. Hossain, Z. A. Begum, I. Alam, H. Sawai, T. Maki, H. Hasegawa, Environmental hazards associated with open-beach breaking of end-of-life ships: a review, Environmental Science and Pollution Research 25 (2018) 30880–30893.

[15] Rijksoverheid, Main aspects of the law 'the open government' (in dutch), 2023. URL: https://www.rijksoverheid.nl/onderwerpen/wet-open-overheid-woo/vraag-en-antwoord/hoofdlijnen-woo.

[16] D. Hadwick, S. Lan, Lessons to be learned from the dutch childcare allowance scandal: a comparative review of algorithmic governance by tax administrations in the netherlands, france and germany, World tax journal Amsterdam 13 (2021) 609–645.

[17] E. Kazim, A. Koshiyama, Explaining decisions made with ai: a review of the co-badged guidance by the ico and the turing institute, Available at SSRN 3656269 (2020).

[18] D. V. Carvalho, E. M. Pereira, J. S. Cardoso, Machine learning interpretability: A survey on methods and metrics, Electronics 8 (2019) 832.

[19] P. Linardatos, V. Papastefanopoulos, S. Kotsiantis, Explainable ai: A review of machine learning interpretability methods, Entropy 23 (2020) 18.

[20] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Advances in neural information processing systems 30 (2017).

[21] Scikit-Learn, Plot permutation importance, 2023. URL: https://scikit-learn.org/stable/auto_examples/inspection/plot_permutation_importance.html.

[22] C. Gini, Variabilità e mutabilità: contributo allo studio delle distribuzioni e delle relazioni statistiche. [Fasc. I.], Studi economico-giuridici pubblicati per cura della facoltà di Giurisprudenza della R. Università di Cagliari, Tipogr. di P. Cuppini, 1912.

[23] L. Breiman, Random forests, Machine learning 45 (2001) 5–32.

[24] L. Shapley, 7. a value for n-person games. contributions to the theory of games ii (1953) 307–317, Classics in Game Theory; Princeton University Press: Princeton, NJ, USA (2020) 69–79.

[25] S. M. Lundberg, G. G. Erion, H. Chen, A. J. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, S. Lee, Explainable AI for trees: From local explanations to global understanding, CoRR abs/1905.04610 (2019). URL: http://arxiv.org/abs/1905.04610. arXiv:1905.04610.

[26] N. S. Platform, 2023. URL: https://shipbreakingplatform.org/annual-lists/.

[27] I. M. Organization, 2023. URL: https://gisis.imo.org/.

[28] E. M. S. Agency, 2023. URL: https://portal.emsa.europa.eu/web/thetis-eu/.

[29] E. R. Q. Fernandes, A. C. P. L. F. de Carvalho, X. Yao, Ensemble of classifiers based on multiobjective genetic sampling for imbalanced data, IEEE Transactions on Knowledge and Data Engineering 32 (2020) 1104–1115. doi:10.1109/TKDE.2019.2898861.

[30] R. Mohammed, J. Rawashdeh, M. Abdullah, Machine learning with oversampling and undersampling techniques: Overview study and experimental results, in: 2020 11th International Conference on Information and Communication Systems (ICICS), 2020, pp. 243–248. doi:10.1109/ICICS49469.2020.239556.

[31] S. Assilian, Artificial intelligence in control of real dynamic systems, Ph.D., Queen Mary, University of London, 1974. URL: http://qmro.qmul.ac.uk/xmlui/handle/123456789/1450, accepted: 1974.

[32] E. H. Mamdani, Application of fuzzy algorithms for control of simple dynamic plant, Proceedings of the Institution of Electrical Engineers 121 (1974)

1585–1588. URL: https://digital-library.theiet.org/content/journals/10.1049/piee.1974.0328. doi:10.1049/piee.1974.0328, publisher: IET Digital Library.

[33] R. Kruse, F. Klawonn, J. Gebhardt, Foundations of fuzzy systems, Wiley & Sons, Chichester, West Sussex, England ; New York, 1994.

[34] L. Pickering, K. Cohen, Toward explainable ai—genetic fuzzy systems—a use case, in: J. Rayz, V. Raskin, S. Dick, V. Kreinovich (Eds.), Explainable AI and Other Applications of Fuzzy Techniques, Springer International Publishing, Cham, 2022, pp. 343–354. doi:10.1007\978-3-030-82099-2_31.

[35] D. Golberg, J. Holland, Guest editorial: genetic algorithms and machine learning, Machine learning 3 (1988) 95–99.

[36] J. H. Holland, Genetic algorithms, Scientific American 267 (1992) 66–73. URL: http://www.jstor.org/stable/24939139.

[37] S. Mirjalili, Evolutionary algorithms and neural networks, in: Studies in computational intelligence, volume 780, Springer, 2019.

[38] M. J. Kochenderfer, T. A. Wheeler, Algorithms for Optimization, MIT Press, 2019. Google-Books-ID: uBSMDwAAQBAJ.

[39] T. Saito, M. Rehmsmeier, The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets, PLOS ONE 10 (2015) e0118432. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0118432. doi:10.1371/journal.pone.0118432, publisher: Public Library of Science.

[40] M. Du, N. Liu, X. Hu, Techniques for interpretable machine learning, Communications of the ACM 63 (2019) 68–77.

[41] P. Plonski, Extract rules from decision tree in 3 ways with scikit-learn and python, 2021. URL: https://mljar.com/blog/extract-rules-decision-tree/.

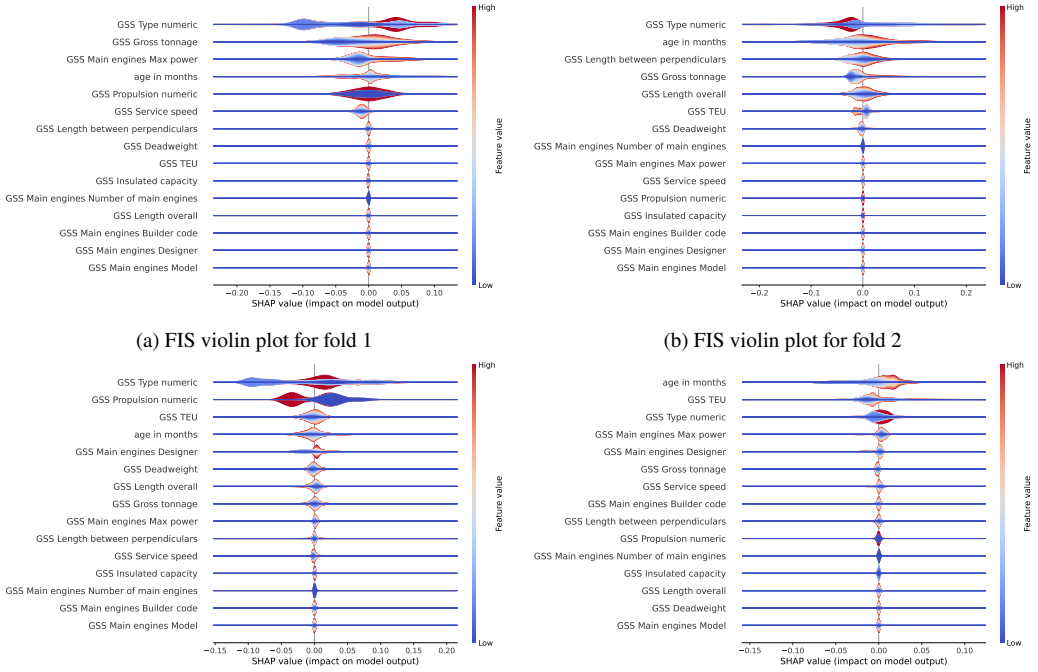
[42] J. Alcala-Fdez, R. Alcala, F. Herrera, A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning, IEEE Transactions on Fuzzy Systems 19 (2011) 857–872. doi:10.1109/TFUZZ.2011.2147794.

Appendices

A. Global SHAP Violin Plots

Due to the difference of distribution in the test sets, the violin plots for each model vary slightly across the folds. The violin plots for fold 0 are given in Section 5.3, and the violin plots are given for the remaining folds here.

The Figures 21, 22, 23 give the violin plots for the FIS, DT and RF models respectively on the remaining folds.



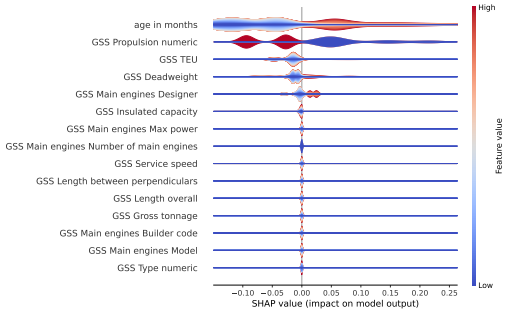
(a) FIS violin plot for fold 1

(b) FIS violin plot for fold 2

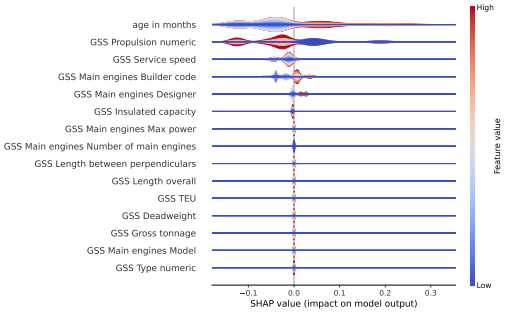
(c) FIS violin plot for fold 3

(d) FIS violin plot for fold 4

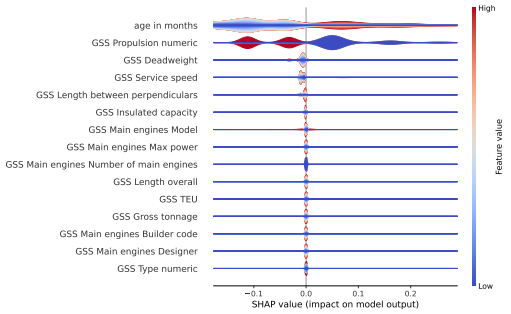
Figure 21: FIS violin plots for remaining folds



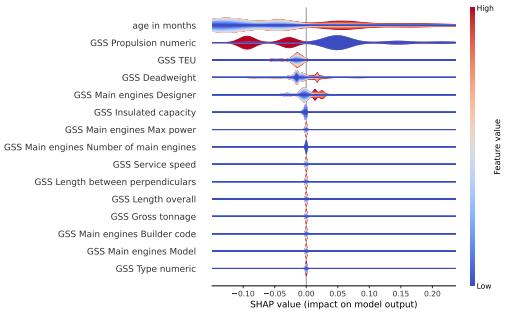
(a) Decision Tree violin plot for fold 1



(b) Decision Tree violin plot for fold 2

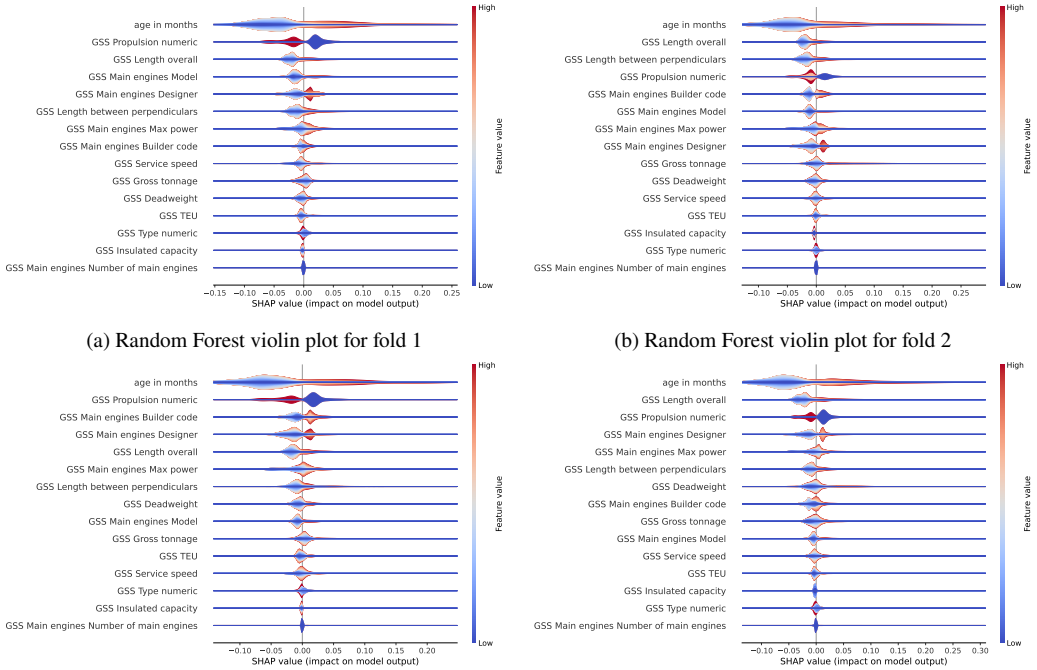


(c) Decision Tree violin plot for fold 3



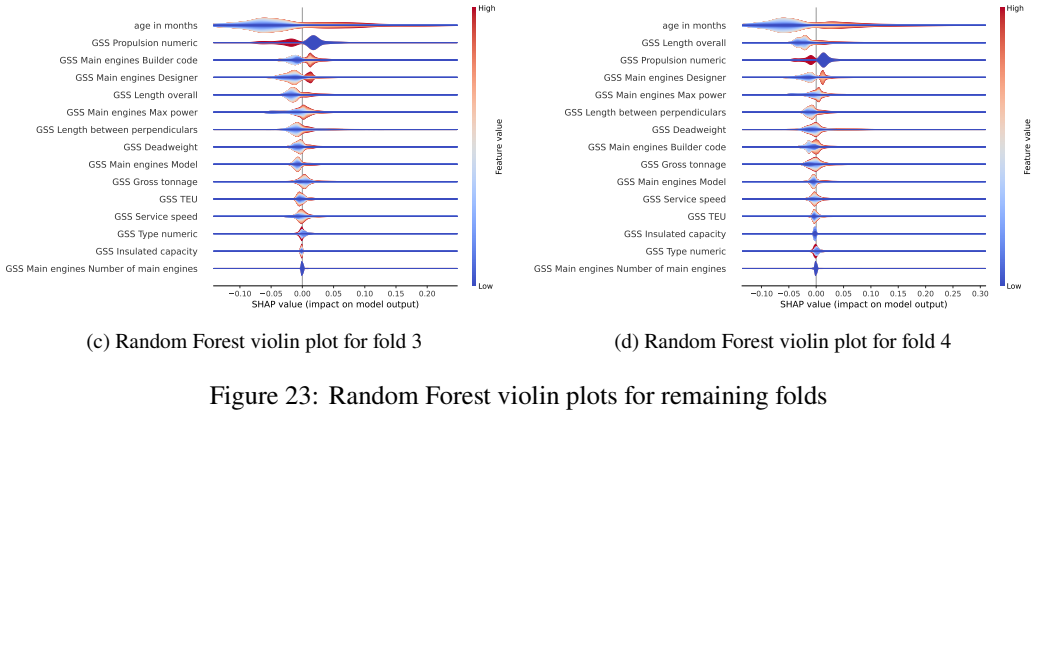
(d) Decision Tree violin plot for fold 4

Figure 22: Decision Tree violin plots for remaining folds



(a) Random Forest violin plot for fold 1

(b) Random Forest violin plot for fold 2



(c) Random Forest violin plot for fold 3

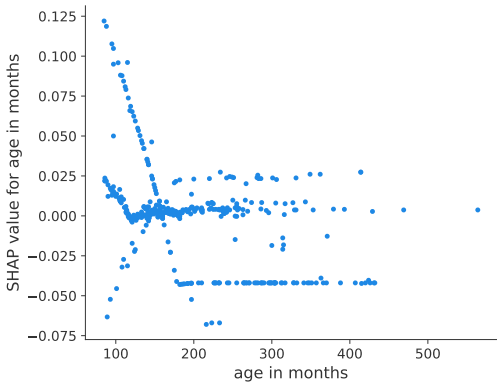
(d) Random Forest violin plot for fold 4

Figure 23: Random Forest violin plots for remaining folds

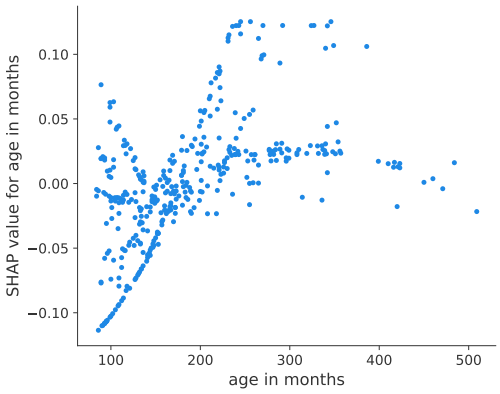
B. Global SHAP Dependence Plots

Due to the difference of distribution in the test sets, the dependence plots for each model and input feature vary slightly across the folds. The dependence plots *age in months* and *GSS Type numeric* for fold 0 are given in Section 5.3.2, and the dependence plots for the remaining folds are given here.

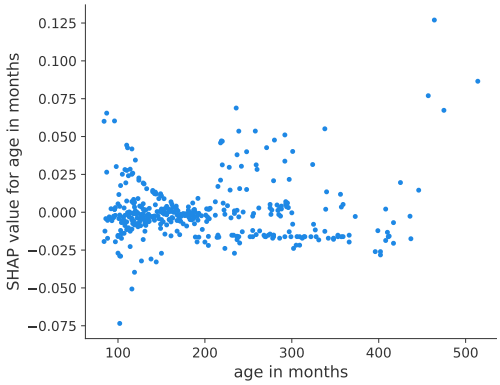
The Figures 24 and 25 give the *age in months* dependence plots for the FIS and RF models respectively on the remaining folds. The Figures 26 and 27 give the *GSS Type numeric* dependence plots for the FIS and RF models respectively on the remaining folds.



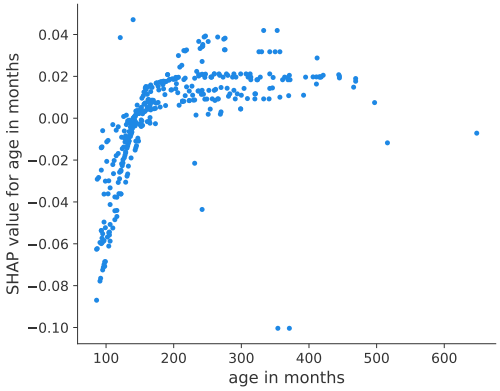
(a) FIS *age in months* SHAP dependence plot for fold 1



(b) FIS *age in months* SHAP dependence plot for fold 2

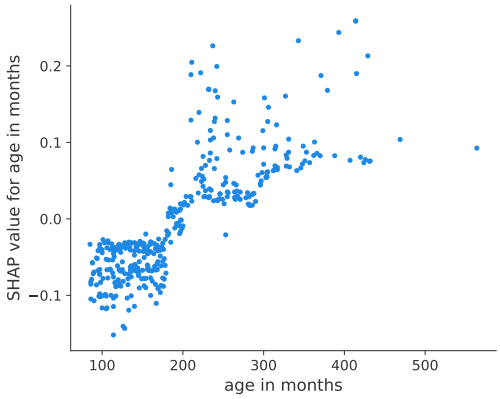


(c) FIS *age in months* SHAP dependence plot for fold 3

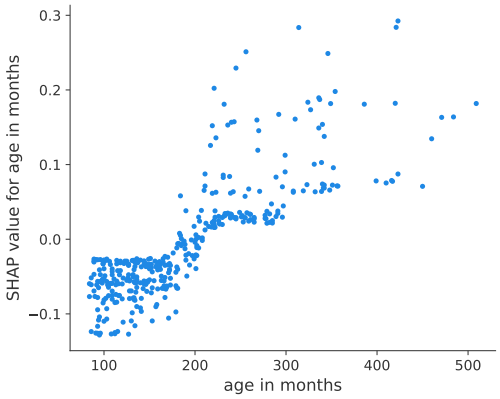


(d) FIS *age in months* SHAP dependence plot for fold 4

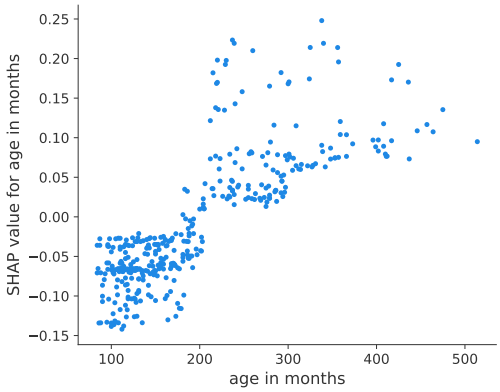
Figure 24: FIS *age in months* SHAP dependence plots for the remaining folds



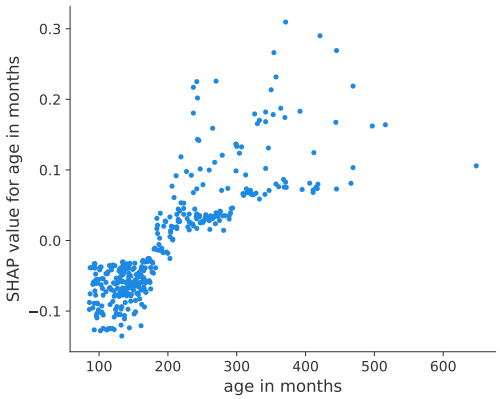
(a) RF *age in months* SHAP dependence plot for fold 1



(b) RF *age in months* SHAP dependence plot for fold 2

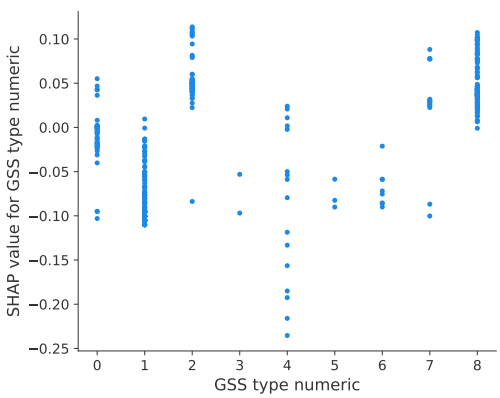


(c) RF *age in months* SHAP dependence plot for fold 3

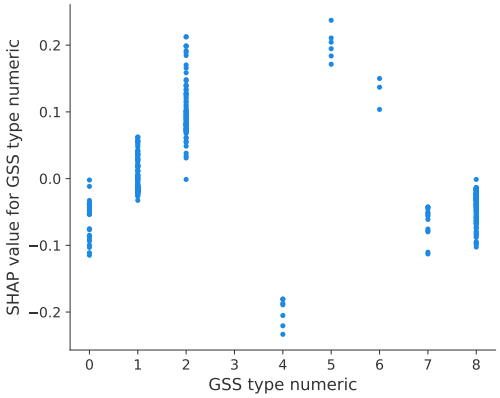


(d) RF *age in months* SHAP dependence plot for fold 4

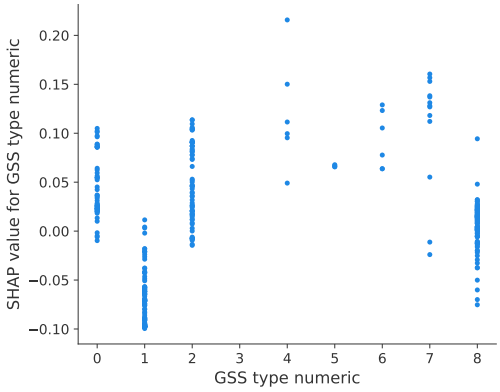
Figure 25: RF *age in months* SHAP dependence plots for the remaining folds



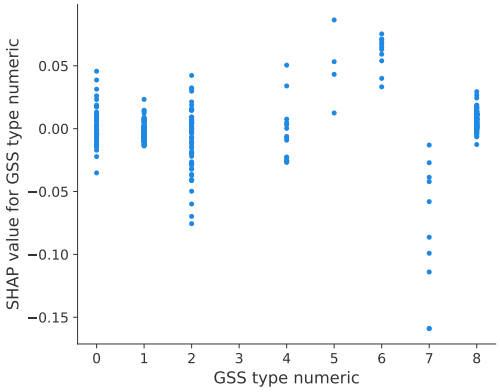
(a) FIS *GSS Type numeric* SHAP dependence plot for fold 1



(b) FIS *GSS Type numeric* SHAP dependence plot for fold 2

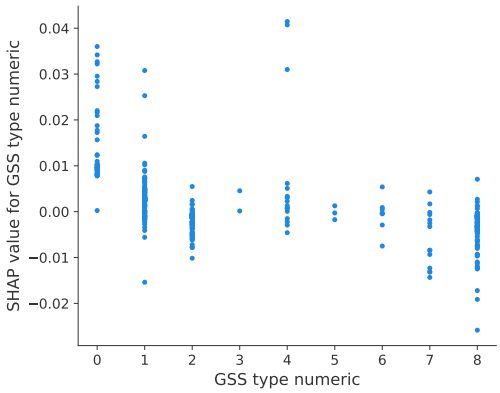


(c) FIS *GSS Type numeric* SHAP dependence plot for fold 3

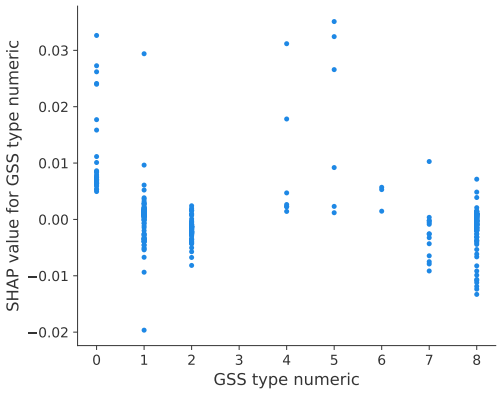


(d) FIS *GSS Type numeric* SHAP dependence plot for fold 4

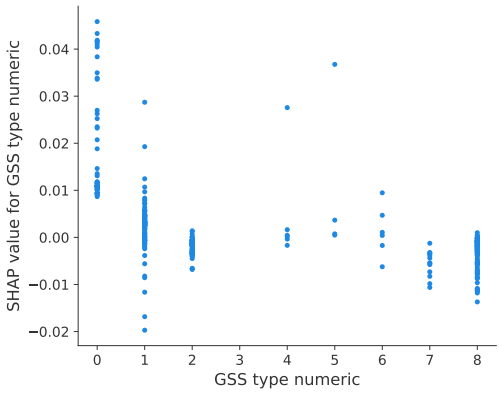
Figure 26: FIS *GSS Type numeric* SHAP dependence plots for the remaining folds



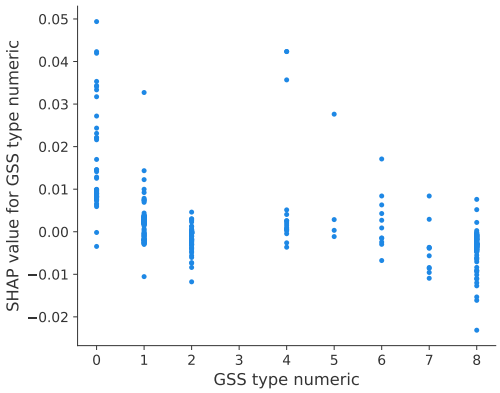
(a) RF *GSS Type numeric* SHAP dependence plot for fold 1



(b) RF *GSS Type numeric* SHAP dependence plot for fold 2



(c) RF *GSS Type numeric* SHAP dependence plot for fold 3



(d) RF *GSS Type numeric* SHAP dependence plot for fold 4

Figure 27: RF *GSS Type numeric* SHAP dependence plots for the remaining folds