COPPE
Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia
UFRJ

IDENTIFYING CLASSIFIER-RELEVANT REGIONS IN IMAGES THROUGH WEIGHTLESS LEARNING

Aluizio dos Santos de Lima Filho

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientadores: Priscila Machado Vieira Lima
Felipe Maia Galvão França

Rio de Janeiro
Julho de 2021

IDENTIFYING CLASSIFIER-RELEVANT REGIONS IN IMAGES THROUGH WEIGHTLESS LEARNING

Aluizio dos Santos de Lima Filho

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientadores: Priscila Machado Vieira Lima
Felipe Maia Galvão França

Aprovada por: Prof. Priscila Machado Vieira Lima
Prof. Claudio Miceli de Farias
Prof. Elias Silva de Oliveira

RIO DE JANEIRO, RJ – BRASIL
JULHO DE 2021

Lima Filho, Aluizio dos Santos de

Identifying Classifier-Relevant Regions in Images through Weightless Learning/Aluizio dos Santos de Lima Filho. – Rio de Janeiro: UFRJ/COPPE, 2021.

XI, 84 p.: il.; $29,7$cm.

Orientadores: Priscila Machado Vieira Lima
Felipe Maia Galvão França

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2021.

Referências Bibliográficas: p. 49 – 53.

1. XAI. 2. WiSARD. 3. Regression. 4. Fuzzy logic. I. Lima, Priscila Machado Vieira *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

O medo serve para identificar
algo perigoso e então assumir
uma postura de preservação da
vida, no entanto quando o medo
te paralisa, sua liberdade e sua
própria vida podem ser perdidas.

Acknowledgments

I discovered how much harder it is to work alone; for that reason, I sought help. The front line of this help are my advisers Priscila Lima and Felipe França, which guided me across this difficult road, many thanks to them for all the good and bad times we had together. We had a lot of experiences that marked our lives forever.

Also, the close friends, who always supported me in the bad times and gave me the hope to continue this road until the end. Special thanks to Amanda Braga, Gabriel Bhering, João Curcio, Christian Cardozo and Lívia Vasconcelos.

Furthermore, during this research, I discovered how important it is to make friends directly related to the research field, which helped me in several discussions, with ideas and discoveries, that would support this research. Thanks to Gabriel Guarisa, Luiz Fernando, Leopoldo Filho, Diego Souza and Juliana for all the time we spent together.

I would like to thank the secretary of the PESC, who helped and helps me a lot with the bureaucracy and more. Thanks to Gutierrez da Costa, Ricardo Cézar and the whole team at PESC, also thanks to Maria de Lourdes and Rosa who make my days easier with their joy.

Finally, The defence line for this work, who always to checks on me, ensuring everything was alright, my mother Maria Aparecida, my father Aluizio Lima and my sister Gisele Oliveira. Thank you all for always supporting me.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

IDENTIFICANDO REGIÕES RELEVANTES PARA OS CLASSIFICADORES EM IMAGENS ATRAVÉS DO APRENDIZADO SEM PESO

Aluizio dos Santos de Lima Filho

Julho/2021

Orientadores: Priscila Machado Vieira Lima
Felipe Maia Galvão França

Programa: Engenharia de Sistemas e Computação

A necessidade de Inteligência Artificial Explicável se torna aparente à medida que os modelos de aprendizagem profunda crescem em popularidade e a Inteligência Artificial é usada em cada vez mais áreas. Muitas técnicas foram propostas até agora para produzir explicações legíveis para os processos de decisão dos classificadores, cada um resolvendo uma pequena peça deste enorme quebra-cabeça. Uma parte disso são as explicações visuais, que buscam produzir imagens que possam destacar o que é o conteúdo relevante para o classificador e indique ao usuário se o modelo toma sua decisão em uma base sólida, ou ao acaso, ou mesmo em uma premissa errada. Portanto, soluções como LIME encontram maneiras de gerar essas explicações em diferentes modelos de aprendizagem, fornecendo uma ferramenta versátil para compreender melhor os modelos de classificação. Embora essas soluções geralmente tentem ser o mais agnóstico possível, a advertência natural é que eles são mais adequados para algumas classes de problemas e classificadores do que outros. Portanto, uma série de modelos explicáveis diferentes são necessários para cobrir o vasto espaço de modelos possíveis para explicar. Nós apresentamos um desses modelos, o Fuzzy Regression WiSARD Interpreter (FRWI), para tentar produzir explicações de alta qualidade a partir de modelos baseados em WiSARD. Além disso, como precisamos de uma maneira objetiva e quantificável de avaliar como os diferentes modelos se comparam, também apresentamos nosso próprio Interpretation Capacity Score (ICS), uma medida para julgar as explicações produzidas. Sob esta métrica além dos subjetivos testes qualitativos, esta nova abordagem FRWI teve resultados promissores, que podem superar o LIME nos cenários testados e fornecer explicações compreensíveis.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

IDENTIFYING CLASSIFIER-RELEVANT REGIONS IN IMAGES THROUGH WEIGHTLESS LEARNING

Aluizio dos Santos de Lima Filho

July/2021

Advisors: Priscila Machado Vieira Lima
Felipe Maia Galvão França

Department: Systems Engineering and Computer Science

The need for eXplainable Artificial Intelligence becomes apparent as deep learning models grow in popularity and Artificial Intelligence is used in more and more areas. Many techniques have been proposed thus far to produce human-legible explanations to the decision processes of classifiers, each solving a small piece of this enormous puzzle. One part of this, are the visual explanations, which seeks to produce images which can highlight what is the relevant content to the classifier, and clue the user in as to whether the model makes its decision on a sound basis, or at random, or even on a mistaken premise. Thus, solutions such as LIME find ways to generate theses explanations across different learning models, providing a versatile tool to better understand classification models. Although said solutions usually attempt to be as model agnostic as possible, the natural caveat is that they are better suited for some classes of problems and classifiers than others. Therefore, a number of different explainable models are needed in order to cover the vast space of possible models to explain. We introduce one such model, the Fuzzy Regression WiSARD Interpreter (FRWI), to attempt to produce higher quality explanations from WiSARD based models. Furthermore, as we need an objective, quantifiable way of gauging how different models compare, we also introduce our own Interpretation Capacity Score (ICS), a measurement process to judge the explanations produced. Under this metric as well as subjective, qualitative tests, this new FRWI approach had promising results, which could beat LIME in the tested scenarios and provide comprehensible explanations.

Contents

List of Figures

Chapter 1

Introduction

The research on eXplainable Artificial Intelligence (XAI) [2] grows with the demand for understanding the decision process made by learning models. It seeks to supply the final user or the designer of the model the explanations needed to comprehend the model's decision.

This demand arises from the need to answer certain questions. For instance, when law enforcement or legislators request the reasoning behind some incident involving an AI system from some company. The XAI is a tool that tries to provide humans such an explanation for the machine decision process. There are different solutions for different scenarios. In an image classification system, it could present some report of what is relevant to the classifier. In a deep explanation system, it could produce images from the concept learned by the deep model.

1.1 Motivation

There are some decisions processes where the explanation is fundamental. As a notable example, consider disease classification [42]. Doctors need to understand why the model suggests a patient has some disease, and in order to do so, he or she needs to know how the classifier reaches its conclusion. In a tumour classification, for example, it is necessary to know where is the tumour, so an explanation of relevant regions can determine that, and also inform to the doctor if the classifier has decided for the wrong reason. This explanation producing process can also significantly help the user to learn when to use or not to use the classifier.

One method to produce an explanation is to generate visualisations that highlight the relevant regions in the image. LIME [39] is one model that does so. It analyses the classifier locally to obtain answers about its behaviour. Despite LIME being model agnostic, it does not perform well in all scenarios, including WiSARD [1] classifier, where the explanation produced by LIME is hard to grasp.

Following are some motivations for this work:

- The first and most important one is the comprehension of black-boxes. The classifiers usually use a very complex model whose behaviour is hard to comprehend, because they use a very large amount of variables.

- Another reason is to have more assets to work with the great variety of learning models that exist, where each one may perform better for given specific scenarios.

- The initial reason for this work is to develop an explanation for the models based on the WiSARD classifier, which can help to improve the model and trust in it in real-life scenarios, as well as to improve the model architecture to reach new research fields.

These motivations are also the base for the contributions exposed in detail in the next section.

1.2 Contributions

The development of the research followed the path presented below.

While looking for existing methods to generate explanations to the WiSARD model, the explainer LIME was found, whose purpose was to produce explanations agnostic to the learning model. However, although LIME can guarantee agnostic explanations, it cannot universally guarantee explanation quality. That is to say, although it can explain every model, it cannot produce useful explanations to every model, and that was exactly the result encountered with WiSARD. The explanations LIME produced for WiSARD marked almost whole image as relevant, therefore failing to elucidate any aspect of the WiSARD's decision process. Extensive analysis of the working process of LIME highlighted how one of the its steps attempts to approximate the values through linear regression, using a line to approximate to a complex function or a discontinuous function, such as the ones WiSARD produces.

This made the problem clear, we needed to be able to handle data that is not easily approximated to a linear function, and that is the role of the Regression WiSARD (ReW) model, as it is specifically made to solve regression problems with a discontinuous approach. As for what data would be tested, the ReW takes a binary input. While it would have been possible to just feed ReW randomly generated images converted to binary, it was more useful to generate binary masks which would highlight portions of the image and focus on those instead.

The main question remains, what the ReW will learn? Since the core objective is to understand the decision WiSARD's process, ReW must learn what it is relevant to the classifier in a given classification. So the scope was narrowed to a single step classification (local observation) in order to minimise the complexity of the classifying function and consequently generate explanations that would more accurately reflect the reality of the classification process.

The next focus point was on how to determine what is relevant to the classifier within the random image generation and ReW learning process. At that point the winning idea was to use a set of fuzzy rules to determine in a logic way what it is relevant. Finally the last point, the set of fuzzy rules needed of input factors to which the rules are applied, these factors are the context that has to be evaluated, in our case they are the produced images and the responses given by the classifier, to transform this into factors the distances between images and responses were calculated, that is, the distances between the vectors and the corresponding "masked" ones.

Although the process of learning what it is relevant is complete, it was missing a minor important detail: the extraction of this information. There was however already a process to extract what was learned by the WiSARD, this process is called DRASiW [18] which produce a mental image. The final result achieved is the FRWI model[16].

The main contribution of this work is the new explainable model based on WiSARD (FRWI [16]), which is capable of producing visual explanations from the images classifiers. To highlight the relevant regions in the image of the classification and then provide some clues about its decision process in specific classifications. Some qualitative experiments were developed to analyse the behaviour of explainable models, where it is possible to see if the explanations are comprehensible or not, and also useful for some conclusion.

These explainable models need some metric to evaluate their performance, so a new metric – the interpretation capacity score – was developed to fill this necessity. Thus, this metric is also a new contribution from this work. Therefore some quantitative experiments with this metric were executed to analyse the performance of the models, and also a possible reason for the acquired results.

Additional contribution brought about by this research include the Regression WiSARD [17] developed in the Palm Oil Prediction competition on Kaggle [23], which had a regression problem to predict the amount of oil produced by a palm tree in a month. Thus, since the WiSARD model had no regression compatibility, the Regression WiSARD was adapted from the n-tuple regression [24] to achieve this goal. Furthermore, when research started there was no library to run and test the models based on WiSARD, so a library called wisardpkg [28] was created to fill this gap and become an unified tool of usage and experimentation with WiSARD model and others.

1.3 Structure of this Document

Chapter 2 presents the area of eXplainable Artificial Intelligence and problems it tries to solve, also how LIME's internal operations works. As the novel explainable model is a new way to solve these problems, it is important to present what it is necessary to explain its building blocks. So Chapter 3 presents the WiSARD model and how it works; a method to produce a global visualisation from the WiSARD model called mental image; Regression WiSARD which helps the new explainable model to aggregate information; and the fuzzy logic with is the core of the new model. Finally, Chapter 4 presents the new model and each of its steps in detail. To evaluate the new approach, plenty of experiments were made, as extensively presented in Chapter 5. Thus, the work concludes with Chapter 6, which presents what was achieved and possible future works for this research.

Chapter 2

eXplainable Artificial Intelligence

This chapter introduces the eXplainable Artificial Intelligence(XAI) [21], focusing in the problem with providing reasoning. Because of that, the field of eXplainable Artificial Intelligence arose to study the reasoning behind decisions made by a black-boxes. The goal is for humans to be able to understand the concept learned by the model and make better decisions based on that.

There are different situations where humans need an explanation from the model. Justification, for example, a bank needs to provide a reason for a credit denial. Discrimination is another possible situation, the model can discriminate people by their skin colour or the location where they live. In medicine, Doctors needs to know why a patient has been diagnosed with a certain disease determined by a model, in order to decide on a treatment and to double check the machine's findings [42].

Explainable models are a solution to generate such reasoning from a black-box model. Those models use different methods to produce explanations, as detailed in Section 2.2. One way to produce explanations is through counterfactual evidence [7] which can play an important role in XAI. Alongside with the decision made by a learning model, decisions that were not made are presented as well as some representation of the criteria for that discard, making it easier to identify reasons for the decision.

Moreover, the counterfacts allows users to make inferences with information that they would otherwise neglect or be reluctant to include. A counterfact could take shape as an explanation produced with images that try to present a reason for the classification, which could then be used in new conjectures.

One kind of explainable model which works with images is LIME [39]. It produces explanations in the domain of image classification. It evaluates a single classification made by a model generating many inputs to the model, and it calculates whether they are relevant for the model. Finally, LIME determines the explanation with a linear model, as shown in the Section 2.3.

2.1 Concepts of XAI

The first important thing to make clear is the difference between interpretability and explainability. This can cause some confusion. Based on the survey [2], the interpretability refers to the passive characteristic from the model to be human understandable, or in other words a given model makes sense for a human observer as is. On the other hand, explainability can be view as the active characteristic of a model, where a model takes action with the intent of clarifying its internal processes.

The survey [2] also gives us a definition for XAI:
"Given an audience, an explainable Artificial Intelligence is one that produces details or reasons to make its functioning clear or easy to understand."
This definition shows us that the explanation always depends on the audience, who is directing the explanation. Thus, the XAI model produces some reasons to clarify model decisions for certain audiences. The XAI aims to achieve a better understanding of the learning models to make them trustworthy to the business sector and users in general. Also, it has a second goal, which is to acquire knowledge through the machine learning model to deal with huge amounts of data, and hopefully lead to new knowledge.

There are various minor goals in which XAI can be useful as the survey [2] details: *trustworthiness*, as mentioned before; *causality*, referring to finding causes for problems in AI systems; *transferability* as XAI can elucidate the boundaries which might affect the model; *informativeness*, with the intent to support decision making; *confidence*, since there are many scenarios where confidence is expected and XAI can help evaluate it; *fairness*, explainability can help guarantee fairness in machine learning models; *accessibility*, allowing users to get more involved in the process of improving and developing the ML models; *interactivity*, increasing the ability of the model to interact with the user through an explainable model; *privacy awareness*, XAI can be used to discover when models store private data in its internal representation.

The next section introduces some approaches to reach some of these goals.

2.2 Types of approaches to produce explanations

There are different approaches to produce explanations, each one suitable for specific situations; a deep explanation for deep learning models; interpretable models for logical models and model induction to deal with any model as a black-box [21].

The deep explanation uses a modified deep learning model to learn features which help in the explanation process. The Grad-CAM [40], for example, tries to extract the relevant features on an image in a convolutional model, so the user can understand why the model takes a specific decision of classification.

The interpretable models use techniques to learn with more structured information in order to have a more comprehensive decision process of the model. The random forest model [22], for example, structures a decision tree based on the data learned in order to make the classification. Although the structure information generated is not always suitable for humans to understand, but the decision path is visible in the decision tree.

Lastly, the model induction, which tries to infer the decision process of a black-box. After this inference, it produces the explanation of the model in a simple way to humans. LIME [39] is an example of model induction in the classification problem domain as can be seen in the next section. The current work fits in a model induction type. It tries to infer the decision behind the classification without internal information of the model. There are also other approaches to tackle explanation production, such as the ones the survey [13] describes.

2.3 LIME

The Local Interpretable Model-agnostic Explanations (LIME) [39] is a technique to explain predictions of classifiers, through a learning process locally around the predictions. So this model analyses the classifier around a prediction and infers what is relevant.

The inference process has two parts: sampling the input data and learning from the produced data. The complete process is present below. This process produces many examples based on the input; therefore, the new examples are close to the given input. It gives each example to the classifier and gets its response. Each produced example has a weight calculated by an exponential kernel function over some distance equation. It then learns how the classifier responds in this situation. It minimises the distance between the responses of the classifier and the complexity of the explanation. It is a trade-off between fidelity to the responses and the complexity of the explanation. More complex explanations usually means higher fidelity explanations, but less readily interpretable. The result of LIME is a representation by a binary vector where each position identifies the presence or absence of a feature in the raw data. When LIME minimises the complexity as a consequence, it minimises the number of features appearing in the result.

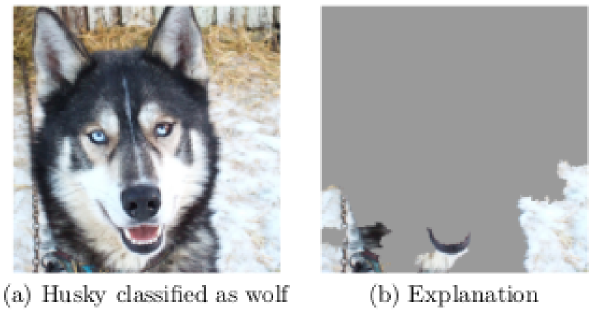
1. Produce examples;

2. Classify new examples;

3. Calculate the weight of each example;

4. Learns with the image and its weight;

5. Extract the info learned through the internal weights;



(a) Husky classified as wolf

(b) Explanation

Figure 2.1: Husky classified as wolf [39]

In Figure 2.1, there is an image of a husky which the classifier identifies as a wolf. In this case, the user asks the question why? LIME provides reasoning as can be seen on the right-hand side, making it clear the classifier selected the snow as relevant to the classification of wolves. The current work does in the same general line, but before discussing it further, the next chapter presents the building blocks of it.

Chapter 3

Weightless WiSARD Model and Fuzzy Logic

There are some essential concepts to understand before introducing the current work. For that reason, this chapter presents the WiSARD [1] model, which is the base to comprehend Mental Image [18] and Regression WiSARD [17], that the proposed model uses.

The WiSARD is a weightless neural net, which uses binary inputs to learn, with a speedy training process if compared to other models [36]. A new classification algorithm called Bleaching [19] was developed recently, which improves general accuracy. A mental image is a concept to gather the complete information learned by the WiSARD model. As a result of this process, a picture emerges where the distribution of the learned data through the features becomes visible.

Regression WiSARD came later to give WiSARD the possibility to perform regression learning. It no longer uses the concept of discriminators. Instead, it only uses RAMs which collect information to predict the output.

And finally, fuzzy logic [46] which gives us the possibility to have answers between 0 and 1. Which is to say, it is not crisp. Using a membership function, it can receive any input and give it a degree of pertinence to a set.

3.1 WiSARD

The weightless model does not use the matrices of weights as a conventional neural network feed-forward. This model works with the random memory access structure to perform the training and classification procedures. Thus it just needs to write to train, and to read to classify on this structure as defined in the n-tuple classifier [4].

WiSARD [1], a weightless model, works with pattern recognition for classification and supervised learning. It attributes each class to a discriminator, which is a concept to split the data learned from each class into a set of RAMs. Therefore, each discriminator only learns about its related label by frequency of access. The usage of RAMs grants fast training and classification to the model as an advantage.

This model uses a binary representation to learn, thus encoding the data to binary is required. For that propose, there are different techniques to process the input data [15, 38], like for example, the thermometer method which, given a value x, determines in which range x is and then the number of ones in sequence associated with that range. Another one would be the cut method, which simply represents values above a given x as one and otherwise as zero.

The model follows a specific structure to train and classify. It has two parts: the discriminators and the RAMs, where a discriminator is composed of a set of RAMs. The input data has a set of labels from the domain, so there is a discriminator to represent each. Furthermore, it only trains with the collection of data which represents it. The RAM is a matrix of addresses pointing to content values related to them, where the address is composed of the binary input. Thus a RAM connects the entry data with its address by a mapping tuple. A tuple is a group of positions in the entry to form an address to access the RAM. Despite the RAM being defined as a matrix, it is often not feasible to implement it as such, as in real scenarios a RAM matrix will occupy extensive memory space although most of its contents will be empty. Therefore, it is much more efficient to use a dictionary instead of a matrix, implemented through a hash structure.

The learning procedure determines the label to select which discriminator has to learn the data. Then it uses the binary input to compose a group of addresses to access the RAMs contents of the discriminator, and there add the value one in the content. This process is illustrated in Figure 3.1, where there are four rams, and each access is determined by the input image H.

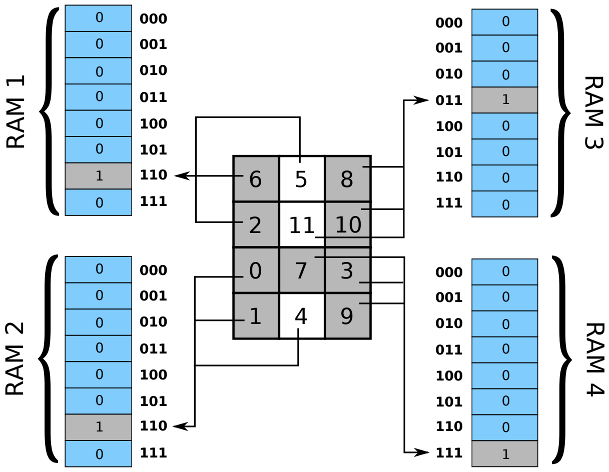


Figure 3.1: The structure of WiSARD

The classification procedure makes access to the RAM precisely like the training process, but instead of writing in the memory, it reads from them so each RAM will give you an answer. Hence, each discriminator will have a set of active RAMs, which are RAMs that answer one instead of zero. So the discriminator with the most significant number of active RAMs is the winner.

Moreover, the classification process through the Bleaching algorithm [19], which make uses of the counter inside the RAM content instead of just looking at whether it is active. With these counters the bleaching performs a process of cutting values returned by the RAMs; thus it starts with a value 1 for the cut, and then it determines how many RAMs are active verifying which are above the cut value. And so, it checks if there is a winner discriminator or not, if the answer is yes, so the algorithm stops; otherwise, it adds one to the value of the cut and makes all checks again until it finds a winner. The Figure 3.2 presents how this procedure performs its actions inside a discriminator. The Figure 3.3 shows the activation of each discriminator with the application of the cut.

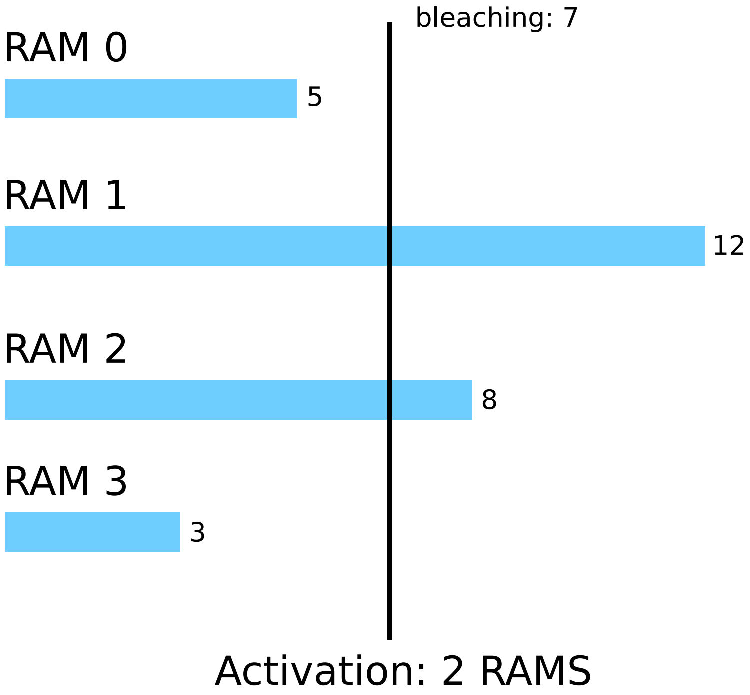


Figure 3.2: The bleaching procedure applied over one discriminator

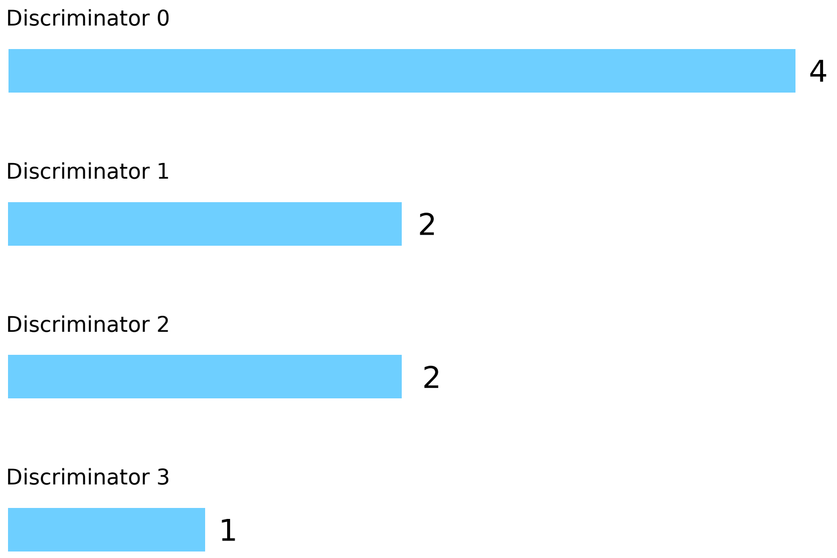


Figure 3.3: The discriminators activation after the bleaching step

Over the years, the application and research of the WiSARD model has increased, yielding important works such as [8–11, 14, 30, 31, 43]. The WiSARD structure performs a straightforward learning process, but there is more it can do. This structure is capable of supplying more information about how the WiSARD model sees the data which it learns. This information is what is called the Mental Image, which is the focus of the next section.

3.2 Mental Image

The mental image was a concept developed by the Burattini [6] to generate some visual elements together with words to provide some degree of explanation of its system. Beyond its goal, it also gives us information about the data behaviour like a heatmap of all data aggregated as can be seen in this work [18]. Thus, it gives us some insights about the data and helps us to improve modelling of a problem.

It is possible to make some starting visual explanation from what the model learns. But it also is too premature to help us with a reason, and the process of generating the mental image removes essential information like the correlation between data which the model learns. Therefore the current work presents an algorithm to capture that information from WiSARD and give us a better explanation of what the classifier learns.

However, as an advantage, Mental Image is information that you can get directly from the model after it learns something, without high computational costs. Its procedure is the inverse of the training. Reading all data from the RAMs and writing in the input structure, but strictly following the rules of the mapping between input and RAMs. So each content value from RAM is written in all positions of input structure, where the value in the binary address is one. In cases where more than one RAM is mapping to the same location; it sums the contents. The Figure 3.4 shows this reverse process.

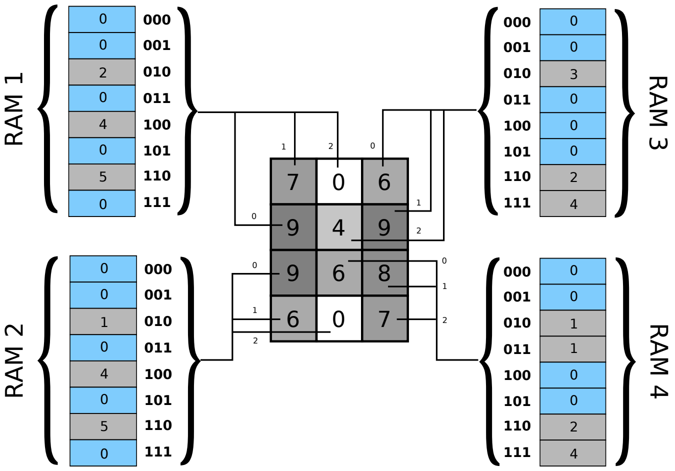


Figure 3.4: The structure of DRASiW

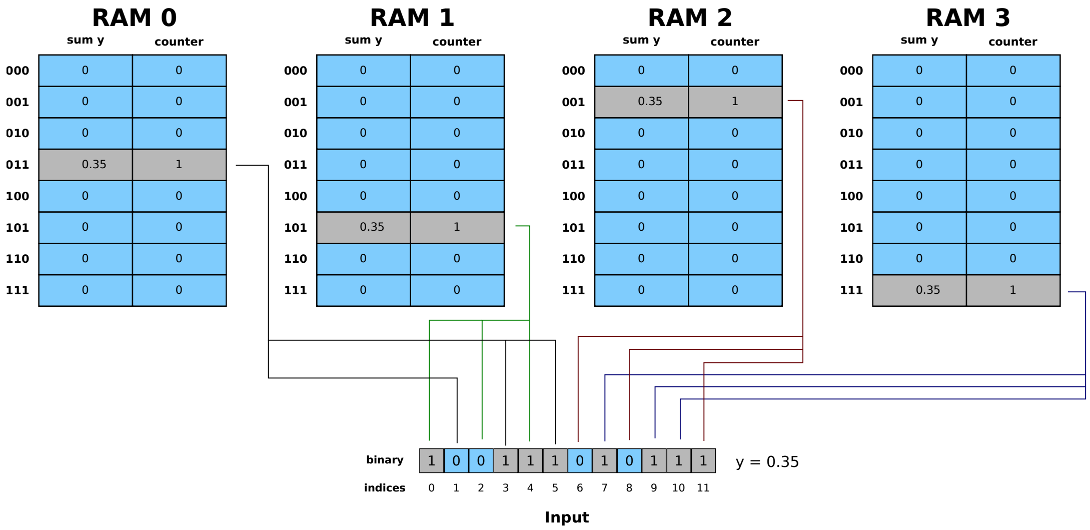
The current work uses the mental image concept in its algorithm, but it uses inside a recent model Regression WiSARD. The next section presents this model of regression and how it works.

3.3 Regression WiSARD

Kaggle's Palm Tree Oil [23] includes a regression problem. Namely in order to predict the amount of oil produced by a set of palm trees, it is possible to model the scenario as a regression problem. At that time we tackled it, the WiSARD model focused only on classification problems, so a new model was needed to participate in this competition and predicts the oil production and, ultimately to add to the WiSARD ecosystem; thus, the Regression WiSARD (ReW) model was conceived [17]. Despite this WiSARD shortcoming, there was already a model with a similar structure which could deal with regression models: the n-tuple regression. Developed and published in 1995 a paper by Kolcz [24], where this model performed regression on controlled scenarios as a simulated environment where the data was produced. In the palm oil tree competition, this model does not perform well with the measure of mean absolute error (MAE) [41]. Thus, to achieve better performance, this new model was created based on both models, n-tuple regression and WiSARD.

Its structure is quite similar to the regular WiSARD model but with the following differences. It is composed only of RAMs, there is no discriminator. Each RAM is responsible for learning the pattern of the associated data. This pattern has two values as input to training: y, the amount to be predicted, and the binary vector x, which is the input data transformed. The RAM as before is a matrix, but this time are three columns: address; counter; and sum_y. The counter works the same way as before, and it is responsible for learning the pattern. The sum of y is a new dimension to the RAM, and it stores the value y from the input added to previous values. There is also the mapping between the input and the RAMs to make access to the RAM possible.

To learn this model receives a binary input and the y value associated with it. The binary input creates several addresses to be accessed for each RAM and their counters to be sum up by one. Furthermore, each address accessed has its y dimension added to the y from the input. Thus, the model learns the pattern and the y associated. The Figure 3.5 shows the learning process of ReW.



RAM 0

	sum y	counter
000	0	0
001	0	0
010	0	0
011	0.35	1
100	0	0
101	0	0
110	0	0
111	0	0

RAM 1

	sum y	counter
000	0	0
001	0	0
010	0	0
011	0	0
100	0	0
101	0.35	1
110	0	0
111	0	0

RAM 2

	sum y	counter
000	0	0
001	0.35	1
010	0	0
011	0	0
100	0	0
101	0	0
110	0	0
111	0	0

RAM 3

	sum y	counter
000	0	0
001	0	0
010	0	0
011	0	0
100	0	0
101	0	0
110	0	0
111	0.35	1

binary	1	0	0	1	1	1	0	1	0	1	1	1
indices	0	1	2	3	4	5	6	7	8	9	10	11

Figure 3.5: The structure of Regression WiSARD

The procedure to make the prediction remains nearly the same, reading from memory, but with a specific step to be done before prediction stage. A simple mean is taken from each RAM - whose addresses are constructed from the binary input - simply by dividing the accumulator variable sum_y by the counters. The mean of means, which is the regression result, can be a simple mean, but other types will yield interesting results as well, such as power mean, harmonic mean, geometric mean and exponential mean. The Figure 3.6 presents this procedure in a simple view.

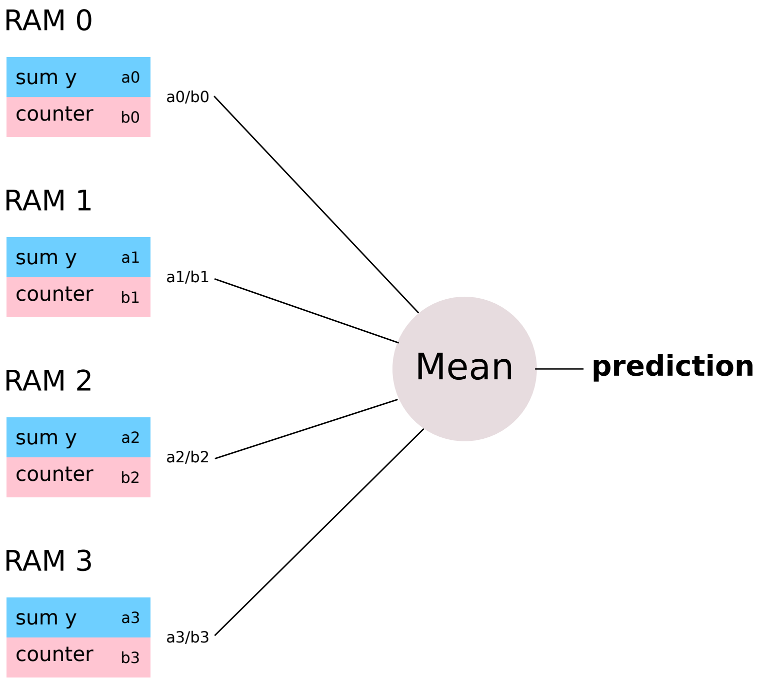


Figure 3.6: The prediction procedure of Regression WiSARD

This model performs well in the competition, but the winner was a model using XGBoost [12], which is popular in this kind of competition on Kaggle. The current work uses ReW's training procedure, where fuzzy rules produce part of the data used. Thus, the next section makes a brief introduction to fuzzy logic, which the next chapter develops in more detail.

3.4 Fuzzy Logic

Most forms of classic logic are limited to operating on binary, true or false, variables. Fuzzy logic [46] is a way to deal with continuous variables instead, usually ranging from 0 to 1. It also presents a more direct translation of the way humans face certain questions.

To illustrate it, consider attempting to posit the question in formal logic, for the sake of whether an air conditioning unit should turn on or not. For the sake of this example, let us imagine only having temperature and humidity sensors. Ideally we would like to describe ranges where it is considered hot, normal and cold, as well as dry, regular and wet and then establish rules by which the unit turns on.

If we adhere to classic binary logic, we would have to decide on steps for our two continuous variables - temperature and humidity - so we can work with them as discrete variables, and then exhaustively describe rules to each step, of which variable.

Fuzzy logic, while performing functionally almost the same, allows us to succinctly take the temperature and humidity as the continuous variables they are and describe continuous functions for arbitrary concepts such as "how hot is it" or "how wet is it". We can then take these attributes and describe the rules to combine them. So instead of writing an extensive list of propositions such as "if it is over 25 degrees, with a relative humidity over 10 percent, turn on the AC", we can write simply, "if it is hot and humid, turn on the AC".

In short, fuzzy logic gives us a way to *determine* how hot, dry and so on in a human-like perspective, while working with formal logic.

Generally, fuzzy problems have a set of input variables, a set of classes to each variable, and a set of rules to solve a problem. Each class has a membership function, which determines the degree of pertinence to it. The Greek letter μ is used to denote a membership function. In our example, μ_{hot} describes the membership function of the class *hot* as can be seen in the Figure 3.7. The fuzzy rules are where we formalise the logic to solve the problem, for example, if the temperature is hot and humidity is wet, then turn on the air conditioner. In this example, we have two variables temperature and humidity, and two membership functions μ_{hot} and μ_{wet}, but we can have as many membership functions as necessary. It all depends on the problem. If a given problem has more membership functions, it may be necessary to define all the combinations of rules to cover all the possibilities of system behaviour to be controlled.

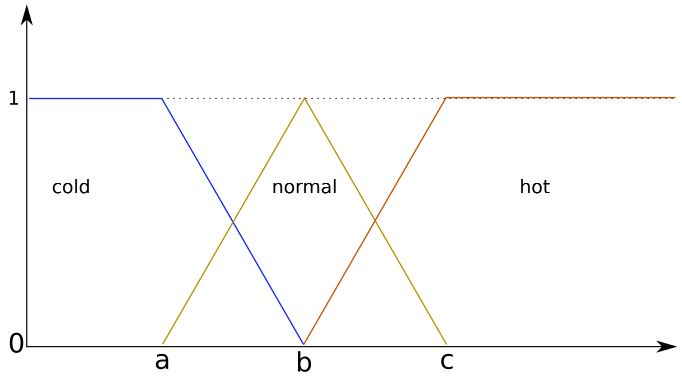


Figure 3.7: Membership functions

A fuzzy problem usually goes through three steps: fuzzification [3], when the membership functions are applied to the input variables; the fuzzy rules, to determine the system behaviour; and the defuzzification to translate the answer of the fuzzy rules to the system. The defuzzification is the inverse process of the membership function; the defuzzification converts the membership degree to be used as a system control variable. The current work does not use the defuzzification process.

To use membership function responses in fuzzy logic, we have the fuzzy logic operators And, Or and Not. These operators are functions to deal with the variation between 0 and 1 from the membership functions. There are multiple possible functions to each operator, for example, the And operator $min(a, b)$ and $a * b$, the Or operator could be $max(a, b)$ or $a + b - a * b$, and at last the Not could be $1 - a$. These functions must respect some restrictions to be considered an operator.

The fuzzy logic is used in this work but without the defuzzification procedure. The fuzzy rules analyse the data and measure the relevance of each data in the algorithm of this work. With all the build blocks presented, the next chapter presents the novel model in detail.

Chapter 4

FRWI

This chapter presents the current work in detail. This new methodology focuses on interpreting the answer from any image classifier, which determines the relevant regions in the image.

It presents the FRWI overview in Section 4.1, where it explains each step: the production of examples, evaluation and generation of a mental image. Each of these steps has an important role in the whole process, as shown in their respective sections. It also displays the process of producing examples and its algorithm and how it works in Section 4.2. After the production, it needs to be evaluated. So the Section 4.3 presents that process in detail. This evaluation is made with a set of rules from the fuzzy logic to determine what is relevant. And finally, Section 4.4 presents how to generate the local mental image. It shows how to use the structure of a Regression WiSARD to obtain this.

4.1 Overview

The Fuzzy Regression WiSARD Interpreter(FRWI) is a model to interpret the answers from classifiers and generates an image which reveals the relevant regions to the classifiers, so humans can have a better comprehension of what the classifiers took into consideration to make their decision. If, when provided an image of the number 7, a classifier decides it is in fact a 3, we would like to know why. To that effect, the explainable model marks what was relevant for that classification, allowing the user to have at least a clue as to what lead to that mistake.

The model has three steps: production of examples, evaluation and generation of a local mental image. The production step locally generates several examples around the input, so it is possible to see how the classifier answers in this environment. The evaluation step uses fuzzy rules to calculate the relevance of each generated example. The local mental image generation step then uses the Regression WiSARD to aggregate all the information calculated before, and at last extracts a mental image from the model. The Figure 4.1 shows how this procedure works and the Algorithm 1 details how to apply it.

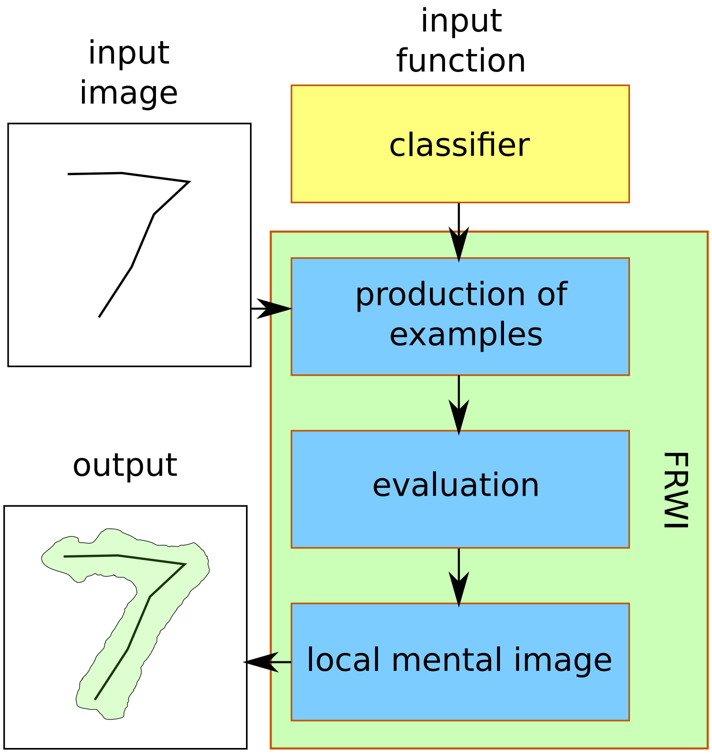


Figure 4.1: FRWI process

Algorithm 1: The general algorithm of FRWI

```
input  : $C$ – classifier
input  : $image$ – input image
input  : $fs$ – feature size
input  : $S$ – total examples
output: $lmi$ – local mental image
begin
    $examples, predictions, binaries \leftarrow$
      $production\_examples(C, image, fs, S)$
    $y \leftarrow fuzzyEvaluation(examples, predictions)$
    $lmi \leftarrow generationOfLocalMentalImage(binaries, y)$
end
```

The production of examples uses random binary masks to get different features from the input. The binary mask represents what is kept from the original image and what is not. Each example is in reality a different portion of the original input with the rest of the image greyed out. An arbitrary number of examples are generated in this format.

The next stage evaluates each image generated in the previous step with the fuzzy rules, which output a value between 0 and 1, where zero is not relevant, and one is entirely relevant. The evaluation uses two variables: the distance between images and the distance between answers. Thus we have two sets; the set of binary masks and the set of fuzzy outputs. These two sets are the input to the next phase, the generation of a local mental image.

The Regression WiSARD learns from these two sets. After it aggregates all this information, the ReW is ready to generate the local mental image. So, the reverse process as defined by a mental image in WiSARD 3.2 is the one to generate the mental image from ReW, where it reads from memory and writes to the input structure. The memory content has two dimensions, so it produces the mental image with these two dimensions. Finally, the result of this process is the local mental image.

The output of this process can be a positive or negative local mental image, depending on which fuzzy rules set is applied to the examples. Positive highlights what is relevant to the classifier to make its decision. So, it is possible to see the regions in the image that are relevant to the classifier decision. Therefore, we can understand the classification process and help us to answer how the classifier makes the right or the wrong decision.

The negative image is to help to identify if there is something in the image that is not part of the class learned by the classifier, so it is something that leads the classifier to a different answer. Therefore, the negative local mental image presents the regions which cause interference in the classifier. This image helps us question whether there is something that can cause a mistake by the classifier.

4.2 Production of examples

This procedure aims to observe how the classifier responds in different scenarios around the input. Conceptually, we could imagine an image of the number seven getting sliced in different ways so we can observe how the classifier responds to each line and the angle between them. So, to achieve this, the process generates several local examples based on the input.

This procedure generates two types of examples to observe the responses of classifier: the random example and its complement. A random binary mask applied over the input produces the random example, and the complement of this mask produces the complement example.

The algorithm produces the random binary mask in the following way. First, it receives a feature window size. So, a square window is selected in a random position in the image and marks it as ones in the same position in a binary mask. It repeats this until it iterates over a random portion of the image size. The feature size determines the size of detail you are trying to highlight. The Algorithm 2 specifies how this process happens.

Algorithm 2: Algorithm of production of examples

```
input  : $C$ – classifier
input  : $image$ – input image
input  : $fs$ – feature size
input  : $S$ – total examples
output: $examples$ – the resulting examples
output: $predictions$ – the predictions applied over the examples
output: $binaries$ – the binaries mask produced
begin
    $examples \leftarrow \{\}$
    $predictions \leftarrow \{\}$
    for $i$ from 1 to $S$ do
        $mask, complement \leftarrow generateMasks(height(image), width(image), fs)$
        $binaries \leftarrow binaries \cup \{mask, complement\}$
        $example1 \leftarrow applyMask(mask, image)$
        $example2 \leftarrow applyMask(complement, image)$
        $examples \leftarrow examples \cup \{example1, example2\}$
        $prediction1 \leftarrow C(example1)$
        $prediction2 \leftarrow C(example2)$
        $predictions \leftarrow predictions \cup \{prediction1, prediction2\}$
    end
end
```

Finally, after it completes the binary mask, then the mask is applied over the original image where the ones represent pixels unchanged, and the zeros represent a "grey" pixel, as the Algorithm 4 presents.

The value of "grey" depends on the scenario and the colour space of the images being studied. In this work, where the images of interest are grey-scale, we chose the value zero - totally black - to colour the grey pixels. In a full RGB space, a good choice could be 50% grey (128 red, 128 green and 128 blue).

Through the previously described mask, the algorithm produces the complement binary mask, where the zeros and ones are flipped. So, it generates the complement example after the application of the complement mask over the entry. The Figure 4.2 illustrates all these processes to make them explicit.

Algorithm 3: Algorithm of generation of masks

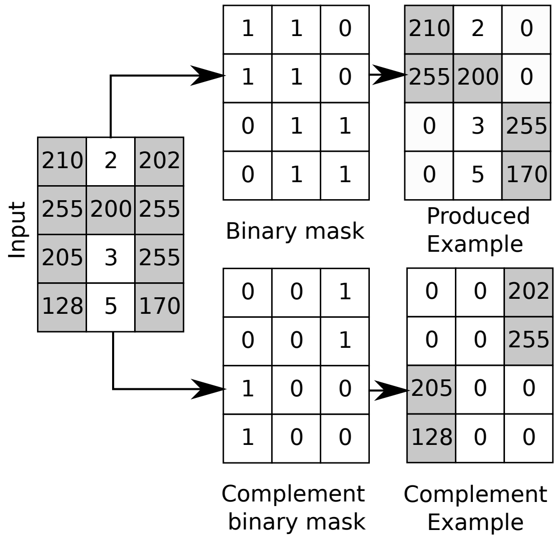
```
input  : $h$ – height
input  : $w$ – width
input  : $fs$ – feature size
output: $mask$ – the resulting mask
output: $complement$ – the complement of the resulting mask

begin
    $mask \leftarrow \{0_0, 0_1, 0_2, ..., 0_s\}$                    /* $s \leftarrow h * w$ */
    $complement \leftarrow \{1_0, 1_1, 1_2, ..., 1_s\}$
    $numberOfFeatures \leftarrow randomBetween(0,1) * ((h * w)/(fs * fs))$
    for $k$ from 1 to $numberOfFeatures$ do
        $l \leftarrow randomBetween(0, h - fs)$
        $c \leftarrow randomBetween(0, w - fs)$
        for $i$ from 1 to $fs$ do
            $pos \leftarrow (l + i) * w + (c + j)$
            $mask[pos] \leftarrow 1$
            $complement[pos] \leftarrow 0$
        end
    end
end
```

Algorithm 4: Algorithm of apply a mask

```
input  : $M$ – mask
input  : $image$ – input image
output: $example$ – the resulting example

begin
    $grey \leftarrow 0$
    $example \leftarrow \{\}$
    for $i$ from 1 to $size(image)$ do
        if $M[i]$ is equal to 1 then
            $example \leftarrow example \cup \{image[i]\}$
        else
            $examples \leftarrow example \cup \{grey\}$
        end
    end
end
```



Input

210	2	202
255	200	255
205	3	255
128	5	170

Binary mask

1	1	0
1	1	0
0	1	1
0	1	1

Produced Example

210	2	0
255	200	0
0	3	255
0	5	170

Complement binary mask

0	0	1
0	0	1
1	0	0
1	0	0

Complement Example

0	0	202
0	0	255
205	0	0
128	0	0

Figure 4.2: FRWI production of examples

With the examples generated, the classifier evaluates them, thus producing a data set of responses. The algorithm expects the classifier to respond with a vector which indicates the relevance of each class. The next section presents the evaluation procedure which uses these vectors.

4.3 Evaluation of examples

This procedure calculates the relevance of each example, which in the next step of the model will be aggregated. Each example has a piece of information about what is relevant, and this information is repeated several times over the whole data set. Through the frequency of those repetitions, the algorithm discovers the relevant regions to the classifier.

The fuzzy rules use four factors as input. These factors are calculated with the distance equation through two dots in a space of n dimensions, each factor uses different vectors to represent the dots in the space. Each of factors are:

- F_I, the distance between the original image vector and the generated image vector;

- F_{Ic}, the distance between the original image vector and the complement of the generated image vector;

- F_R, the distance between the response vector of the classifier over the original image and over the generated image;

- F_{Rc}, the distance between the response vector of the classifier over the original image and over the complement of the generated image;

Each of these distances can be interpreted as how similar each vector is. Once again, in this context, the generated example is a section of the original image selected by a binary mask and the complement is the section selected by the complement of the mask.

$$factor(a,b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \tag{4.1}$$

The next stage adjusts the membership functions to each of the four factors. Each factor has three membership functions: low distance, middle distance and high distance. So, low distance means the group of distances factors which are very similar to the original value. The high distance means the group of distances factors which are very different from the original one, and the middle is naturally somewhere in between. So, the closer the factor is to zero, the more similar it is, and conversely, the furthest from zero, the more different it is. These three types of membership function can be seen in the equations 4.2, 4.3, 4.4.

$$\mu_H(x) = \begin{cases} 0, & \text{if } x < b \\ (x-b)/(c-b), & \text{if } x \geq b \text{ and } x \leq c \\ 1, & \text{if } x > c \end{cases} \tag{4.2}$$

$$\mu_M(x) = \begin{cases} 0, & \text{if } x < a \\ (x-a)/(b-a), & \text{if } x \geq a \text{ and } x < b \\ (c-x)/(c-b), & \text{if } x \geq b \text{ and } x \leq c \\ 0, & \text{if } x > c \end{cases} \tag{4.3}$$

$$\mu_L(x) = \begin{cases} 1, & \text{if } x < a \\ (b-x)/(b-a), & \text{if } x \geq a \text{ and } x \leq b \\ 0, & \text{if } x > b \end{cases} \tag{4.4}$$

These membership functions together have three variables a, b and c. The Algorithm 5 describes how to calculate these variables through the data of the factors. The algorithm is the K-means algorithm [25] with a single step, and with three clusters.

So, each factor will have three variables determined by the above algorithm to be placed in the membership functions, as shown in the Figure 4.3.

Algorithm 5: Algorithm to calculate variables a,b and c

```
input  : data – vector of values
output: clusters – the resulting variables a,b and c respectively
begin
    $sort(data)$
    $piece \leftarrow size(data)/3$
    $clusters \leftarrow \{0, 0, 0\}$
    $c \leftarrow 1$
    for $i$ from 1 to $size(data)$ do
        $clusters[c] \leftarrow clusters[c] + data[i]$
        if $i$ mod $piece$ is 0 then
            $c \leftarrow c + 1$
        end
    end
    for $j$ from 1 to 3 do
        $clusters[j] \leftarrow clusters[j]/piece$
    end
end
```

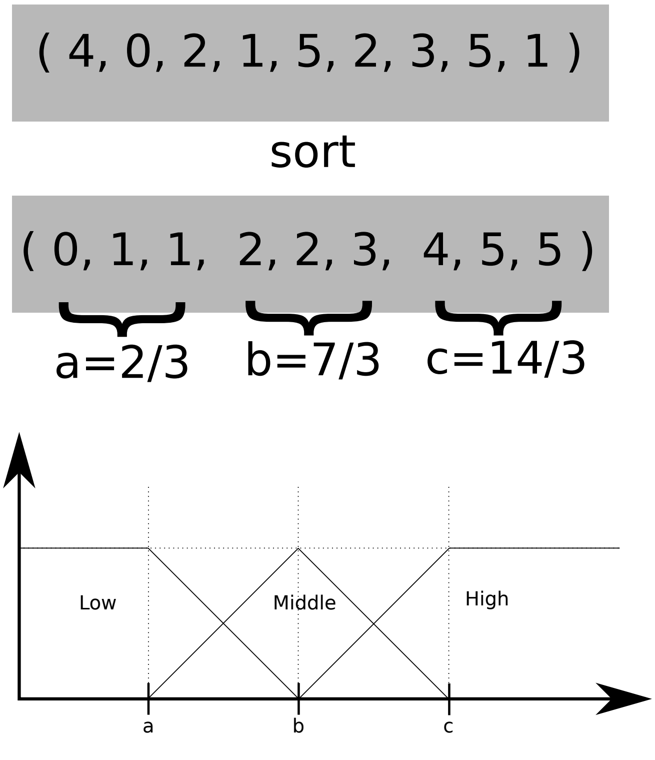


Figure 4.3: FRWI adjust of membership function

With all the membership functions determined, the fuzzy rules can play their role. First of all, the equivalence of the fuzzy operators used in this model are defined in the Equation 4.5.

$$\begin{aligned} a \land b &\equiv a * b \\ a \lor b &\equiv a + b - a * b \\ \neg a &\equiv 1 - a \end{aligned} \tag{4.5}$$

There are the rules for positive and negative relevance. The positive set of fuzzy rules p has two parts: one to determine what contributes to the relevance r; and what does not contribute nr. These two parts help to distinguish the relevant regions. The positive relevance regions help to understand what features the classifier uses to make its decision. The set of fuzzy rules used are found in the Equation 4.8.

$$r(x_1, x_2, y_1) = \begin{cases} \mu_{Lx_1}(x_1) \land \neg\mu_{Lx_2}(x_2) \land \mu_{Ly_1}(y_1) \lor \\ \mu_{Lx_1}(x_1) \land \neg\mu_{Lx_2}(x_2) \land \mu_{My_1}(y_1) \lor \\ \mu_{Lx_1}(x_1) \land \neg\mu_{Lx_2}(x_2) \land \mu_{Hy_1}(y_1) \end{cases} \tag{4.6}$$

$$nr(x_1, x_2, y_1) = \begin{cases} \mu_{Hx_1}(x_1) \land \neg\mu_{Hx_2}(x_2) \land \mu_{Ly_1}(y_1) \lor \\ \mu_{Hx_1}(x_1) \land \neg\mu_{Hx_2}(x_2) \land \mu_{My_1}(y_1) \lor \\ \mu_{Hx_1}(x_1) \land \neg\mu_{Hx_2}(x_2) \land \mu_{Hy_1}(y_1) \end{cases} \tag{4.7}$$

$$p(x_1, x_2, y_1) = r(x_1, x_2, y_1) \land \neg nr(x_1, x_2, y_1) \tag{4.8}$$

Similarly to before, x_1 and x_2 here are distance factors between the original image the generated example and its complement. Meanwhile y_1 is specifically the distance between the response vectors from the classifier over the original image and the generated example. The membership function μ_{Lx_1} is the function μ_L defined in 4.4 and adjusted by the factor x_1. As previously defined, p is the positive function which evaluates the factors of one produced example. The same function can be used for the complement of the example, in which case y_1 should be the distance between the classifier responses for the original image and the example complement, and x_1 and x_2 should be swapped. The resulting equations can be seen in 4.9, and its returned values are used in the next stage where the information is aggregated.

Thus, it result in two equations one applied over the produced example and other in the complement of the produced example as can be seen in the Equation 4.9, these resulting values are used in the next stage where all the information is aggregated.

$$\begin{aligned} p_1 &= p(F_I, F_{Ic}, F_R) \\ p_2 &= p(F_{Ic}, F_I, F_{Rc}) \end{aligned} \tag{4.9}$$

The negative relevance regions help to understand what features could make the classifier make the wrong decision. In other words, the elements that are hindering the classifier in making its decision. The Equation 4.10 describes the fuzzy rules of negative relevance.

$$n(x_1, x_2, y_1) = \neg \begin{cases} \mu_{Lx_1}(x_1) \wedge \mu_{Lx_2}(x_2) \wedge \mu_{Ly_1}(y_1) \vee \\ \mu_{Lx_1}(x_1) \wedge \mu_{Lx_2}(x_2) \wedge \mu_{My_1}(y_1) \vee \\ \mu_{Lx_1}(x_1) \wedge \mu_{Lx_2}(x_2) \wedge \mu_{Hy_1}(y_1) \vee \\ \\ \neg\mu_{Lx_1}(x_1) \wedge \neg\mu_{Lx_2}(x_2) \wedge \mu_{Ly_1}(y_1) \vee \\ \neg\mu_{Lx_1}(x_1) \wedge \neg\mu_{Lx_2}(x_2) \wedge \mu_{My_1}(y_1) \vee \\ \neg\mu_{Lx_1}(x_1) \wedge \neg\mu_{Lx_2}(x_2) \wedge \mu_{Hy_1}(y_1) \end{cases} \tag{4.10}$$

Where x_1,x_2 and y_1 are defined the same as in the positive function. The membership functions also has the same meaning as in the positive function, but now n is the negative function to evaluate one produced example. As before, to evaluate the complement example, the same can be done. Thus, these resulting in two equations as can be seen in 4.11, one applied over the produced image, and other applied over the produced complement image. Theses results are used in the next stage, to produce the negative relevance.

$$\begin{aligned} n_1 &= n(F_I, F_{Ic}, F_R) \\ n_2 &= n(F_{Ic}, F_I, F_{Rc}) \end{aligned} \tag{4.11}$$

With that values are calculated in the Equations 4.9, 4.11, the local mental images for the positive and negative relevance can play their central role. The next section describes how this process happens in detail.

4.4 Local Mental Image

The local mental image aggregates all the previous information calculated to produce a visual explanation of what is happening in the decision process of the classifier in one classification. The trained ReW is the one to provide this.

The procedure to produce the mental image has the following steps:

1. generating the neighbour mapping;

2. instantiating the ReW;

3. training the ReW;
4. extracting of the mental image from ReW;
5. finally normalising the mental image;

First, it needs to calculate the neighbour mapping to be able to instantiate the ReW. The neighbour mapping has the intention to create a RAM for each pixel of the image. Thus, the tuple uses the central pixel and its direct neighbours totalling nine pixels, making it a nine-tuple. This mapping allows the ReW to spread the information to its neighbourhood of RAMs, and then it creates a smoother image than without it. The Algorithm 6 describes in detail how to execute this process.

Algorithm 6: Algorithm to calculate the neighbor mapping

```
input  : $width$ – width from the image
input  : $height$ – height from the image
input  : $n$ – neighbor size
output: $mapping$ – the resulting mapping
begin
    $tuple\_size \leftarrow (2 * n + 1)^2$
    $s \leftarrow width * height * tuple\_size$
    $mapping \leftarrow \{0_1, 0_{2,3}, ..., 0_s\}$
    $pos \leftarrow 0$
    for $l$ from 1 to $height$ do
        for $c$ from 1 to $width$ do
            for $ll$ from $l - n$ to $l + n$ do
                for $cc$ from $c - n$ to $c + n$ do
                    if $ll < 0$ or $cc < 0$ or $ll > height$ or $cc > width$ then
                        $mapping[pos] \leftarrow l * width + c$
                    else
                        $mapping[pos] \leftarrow ll * width + cc$
                    end
                    $pos \leftarrow pos + 1$
                end
            end
        end
    end
end
```

Given the previous information, the algorithm instantiates the ReW with the tuple size of nine and using the neighbour mapping. It creates two ReW, one to the positive relevance and other to the negative. After the instantiation, it trains each ReW with the binary mask data set generated in the production of examples step and the y value, which is the output from the fuzzy rules, where the evaluation of positive relevance is the input to the positive ReW and the negative relevance to the negative ReW.

When it completes the training process, the stage of extraction of the mental image from the ReW can start. The mental image is a heat map of what the WiSARD learns. In the case of Regression WiSARD the concept is the same, but the process is different, as for the purposes of this work it requires to use the y value learned too. It reads from memory, both values in this case, and writes the values in the input format. The Algorithm 7 describes how to apply this procedure, where it produces the raw local mental image.

The final step, the raw output from the Algorithm 7 is not useful yet, because the relevance regions are not visible and all values in the output are too high. In order to make the information it produces clearer, it applies a simple normalisation. In this normalisation, the biggest value is represesnted by one, and the lowest value represents zero; and all other values are normalised on this range. The Algorithm 8 shows how to proceed with this process.

Algorithm 8: Algorithm to normalise vector with the lowest and biggest value as range

```
input  : $data$
output: $data$
begin
    $min\_value \leftarrow min(data)$
    $range \leftarrow max(data) - min\_value$
    for $i \leftarrow 1, size(data)$ do
        $data[i] \leftarrow (data[i] - min\_value)/range$
    end
end
```

This chapter described the whole process of the model The next chapter describes the tests performed on the model in different cases to help us understand its behaviour in certain environments. It also shows the performance of the model compared to LIME in some scenarios.

Algorithm 7: Algorithm to extract the mental image from Regression WiSARD

```
input  : $mapping$
input  : $s$ – the size of the image
input  : $tuple\_size$
input  : $memory$ – list of dictionaries from the RAMs contents
output: $mi$ – the resulting mental image
begin
    $mi \leftarrow \{0_1, 0_2, 0_3, ..., 0_s\}$
    for $r$ from 1 to $size(memory)$ do
        $y \leftarrow 0_1, 0_2, 0_3, ..., 0_n$                /* $n \leftarrow tuple\_size * 2$ */
        for $p$ from 1 to $size(memory[r])$ do
            $addr \leftarrow getAddress(memory[r][p])$
            $counter, sum\_y \leftarrow getRamContent(memory[r][p])$
            for $i$ from 0 to $tuple\_size - 1$ do
                /* / is a integer division; & is and operation bit
                a bit; % is module operation; << is a shift left
                operation of bits */

                $bit \leftarrow addr[i/8]\&(1 << (i\%8))$
                if $bit$ is not 0 then
                    $y[i * 2] \leftarrow y[i * 2] + counter$
                    $y[i * 2 + 1] \leftarrow y[i * 2 + 1] + sum\_y$
                end
            end
        end
        for $j$ from 1 to $tuple\_size$ do
            if $y[j * 2]$ is not 0 then
                $pos \leftarrow mapping[r * tuple\_size + 1]$
                                /* here is not a interget division */

                $mi[pos] \leftarrow y[j * 2 + 1]/y[j * 2]$
            end
        end
    end
end
```

Chapter 5

Experiments

After presenting the methodology, it is finally the moment to evaluate the model in specific scenarios, to comprehend its functionality and to learn when to use it and when not to use it. So, this chapter discusses the experiments conducted to evaluate the performance of the explainable models in different scenarios.

Firstly, it presents the environment, and the data sets used to apply the experiments. After, it shows the new equation to evaluate the explainable models, and possibilities to compare their performance of explainability in a quantitative fashion.

The last two sections are about the analysis of the experiments. The first one, evaluates them in a qualitative manner, through the images generated by them. That is, it reports on plenty of pictures from different classes, to see what is useful to the user and not. Therefore, it is possible to see the explainers supplying information which can help us to understand the decision process of the classifier. The last section presents the quantitative evaluation with the new equation, and so to see the capability of the explainers to generate precise explanations.

5.1 Experiments Base

The first subsection presents how to apply the quantitative evaluation through the equation that it defines. So, it describes the new score in detail and how each part works. Later, it shows the data sets, the distribution of the train set and the test set, and also the classes that each one contains. The last subsection shows how each classifier executes its training and classification.

5.1.1 Methodology of the experiments for quantitative evaluation

Both the new model and LIME produces images as a form of visual explanation. Although we can discuss whether it is useful or not, we can not measure its explanation performance with any degree of precision. Therefore we created a new score to measure the explainable model performances which use images to highlight the relevant regions in pictures to the classifiers. Thus with this metric, it is possible to compare and evaluate explainable model performances. We do not measure the quality of the explanation from a human perspective, but measuring the capacity of interpretation from the explainable model in some scenario.

The interpretation capacity score (ICS) uses three factors to make its value. First, *the keeps response*, this one verifies if the classifier keeps its response after applying the relevant regions as a mask over the original image and it keeps just the relevant region erasing the remaining portion of the image. Thus, it is possible to check if the relevant regions are indeed relevant. But only this is not enough, because we can remove this relevant region and the classifier keeps its response, and therefore these regions are not relevant. So there is the second factor, *the change response.* This one as above applies the relevant regions as a mask over the original image. But this time it removes the relevant regions and verify if the classifier changes its response, and therefore more one time checking if the relevant regions are indeed relevant. Again these two factors are not enough, because the explainable model can actually select the entire image as relevant and this does not give us useful information, so it is more useful if the explainable model selects small areas. For that reason, the last factor, *the interpretation complexity* measures the proportion of the relevant region over the image. Thus if it is bigger than the complexity is greater, otherwise will be lesser. The two first factors are joined in a harmonic mean and weighted by the complement of interpretation complexity as the Equation 5.1 shows forming the ICS.

$$ics = (1 - c) * 2 * \frac{p * n}{p + n} \tag{5.1}$$

Where p is the mean of the keeps response, n is the mean of changes response, and c is the mean of interpretation complexity. It calculates each of these means over some test data set.

Thus, it is possible to measure the explanation performance from the explainable models. The next section presents the data sets used to compare them.

5.1.2 Data Sets

The first data set is the MNIST of handwritten digits [26], which is a subset from a larger set NIST [20]. All images have a normalisation in a fixed size and centred. It only contains greyscale images of 28 pixels of height and width, and the classes are from the number zero to the nine as shown in the Figure 5.1.

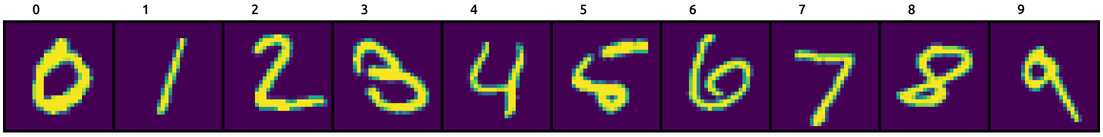


Figure 5.1: MNIST data set

This data set has two groups: one for training and the other one to test the performance of the classifiers. The training set has 60000 images, and the test set has 10000 images, both sets are balanced data through the classes.
This one is a very controlled data set to test the viability of the algorithm proposed in this work. Where it is elementary to see the difference between the classes, and to the classifiers achieve remarkably high accuracy. Without high accuracy from the classifier, it will not make sense to analyse the performance of the proposed algorithm, because the classifier itself is flawed and incapable of telling us which class each image belongs with certainty.

The second data set has the same configuration as the previews one. The Fashion MNIST [44] has greyscale images also of size 28 by 28 pixels, but this time it has images of clothes instead of numbers, where each number is related to an article of clothing: 0 T-shirt/top; 1 Trouser; 2 Pullover; 3 Dress; 4 Coat; 5 Sandal; 6 Shirt; 7 Sneaker; 8 Bag; 9 Ankle boot; The Figures 5.2 shows each class of image. It also has a training set of 60000 images and test set of 10000 images.

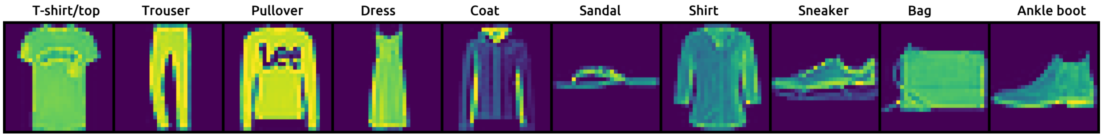


Figure 5.2: Fashioin MNIST data set

The Fashion MNIST is also a controlled dataset with clearly distinguishable classes from a human perspective, if not that simple to the classifiers. Thus, this data set brings some level of difficulty to the classifier to evaluate the FRWI and LIME performance in a different classification scenario.

5.1.3 Organisation of learning models to train with MNIST and Fashion MNIST

Our work has three learning models for its experiments, to evaluate the proposed algorithm and compare it to LIME. The first one is the WiSARD model which learns by frequency. The second one is the Ridge Regression which is a regression model applied to classification. And the last one is the Random Forest which builds a decision tree during the training.

Both data sets have the same setup despite being a different scenario, so the same configuration was applied to train and classify the data. The training set has a total of 60000 samples, and the test set has 10000. The models train on the training set, and are then evaluated based on their responses to the test set.

To the WiSARD model, the image was processed with the mean threshold algorithm to transform into a binary vector. This algorithm calculates the mean value from the image and after it checks if the pixel value if above the mean value. Suppose it is above so it encodes to 1 otherwise to zero. Beyond that, it uses replication of 4 times, this means instead of to encode the pixel to only one bit it will encode it to 4 equals bits, so the one becomes 1111, and the zero becomes 0000. The model uses a tuple size of 40 to learn. The classes vector remain the same with values between zero and nine, which represents the real classes. To supply suitable information to FRWI and LIME, a predict function was developed to make the classification through a vector that has the relevance of each class. Thus, the last step from the bleaching algorithm, where each class has the total of active RAMs, is taken and normalised between zero and one.

For the Ridge Regression, it transforms each image to vector with the original values. Each value from the classes vector is transformed into a vector by one hot encode algorithm, to make it feasible to the model applies a regression. The model uses a regularisation value of zero.

The last one is the Random Forest, which transforms each image to vector as before, and the L2-norm normalises each vector. The classes vector remain the original numbers from zero to nine to represent the classes.

The next section presents the setup of the experiments of the explainable models, which uses the learning models shown above, whose configuration the current section details.

5.2 Qualitative experiments

In this section, the visual experiments will be made to see how the explainable models produce their results in some specifics scenarios. First, how the experiment was conducted is explained in the next subsection, there the implementation and parameters used are discussed. There are two types of experiments in this section: the comparison of positive explanations, to understand how the explainable model deals with the given situation; and the comparison of negative explanations to see how it responds to some noise data and help suggesting future improvements. The next subsections will discourse about them.

5.2.1 Organisation of the experiment

These experiments will produce local mental images with the variation of data set, learning models and classes. To achieve this, first, the learning models must be trained as the Subsection 5.1.3 specifies. After the training, for each class from each data set the explanation is produced from each explainable model.

The FRWI model is set up with the window size of 4, because the features are small, and the number of examples of 10000 to achieve smoother results.

The LIME use the example of 10000 to be equivalent to the FRWI. LIME uses the segmentation algorithm from scikit-learn [5] with the following parameters: $kernel\_size = 1$; $max\_dist = 200$; $ratio = 0.2$.

In the case of negatives explanations, the number of examples produced was from 100000 to both explainable models, because the FRWI was incapable of producing local mental images that have significant information on it with examples smaller than this.

5.2.2 Comparison of the positive explanation

MNIST

First, the MNIST data set is analysed. Thus, it is possible to see how the explanation from each class in this data set was produced. Both explainable models produce their explanations, and thus it is possible to compare them visually. Not only classes, but also the learning models were varied to see the behaviour of the explainable models in scenarios of different complexity levels of learning models.

The WiSARD model is the first subject of the experiment. It was trained as mention before in Section 5.1.3. And then, the explainable models are applied, resulting in the Figure 5.3.

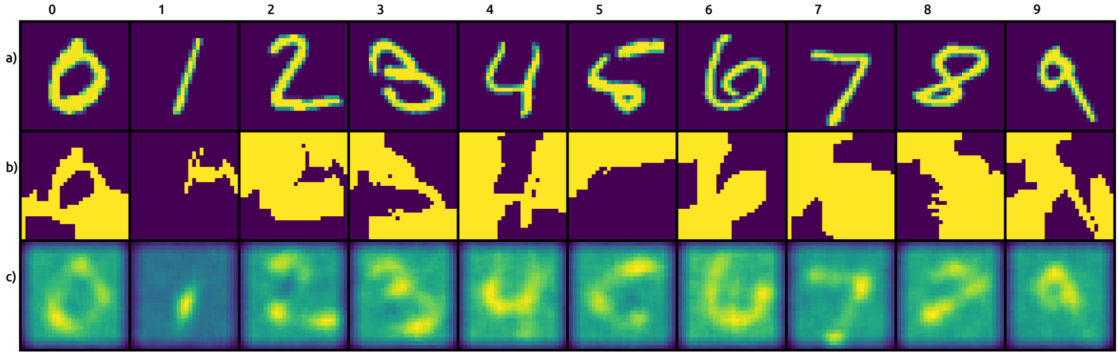


Figure 5.3: MNIST on WiSARD qualitative experiment: a) original images ; b) LIME explanations; c) FRWI explanations

The first line contains the original images as they come from the data set. The third one is the FRWI explanations from these original images over the WiSARD, and the second line has explanations from LIME in the same conditions. In the Figure 5.3, FRWI shows explanations covering the almost entire number for the classes 0, 3 and 4, where these are examples have low noises in the data set. The class 7 also can fit in this description. But other examples have even less noises, such as 1, 2, 6. The explanations do not cover the entire number, indicating some preferential regions to learning model. Interestingly, disturbances such as small rotations, incomplete figures or twisted images, such as seen in 9, 5 and 8, causes bad behaviour. The explanation covers only a few parts of the numbers, which teaches us that the learning model can not deal well with this kind of situation. Furthermore, these explanations from FRWI provide us with some precious information about how the WiSARD acts in this scenario, therefore this new approach begins to show some utility. In the case of LIME, it is hard to collect some useful information about the model in this scenario, but the classes 0, 3, 4 and 6 is possible to see the explanation drawing the mould of the number.

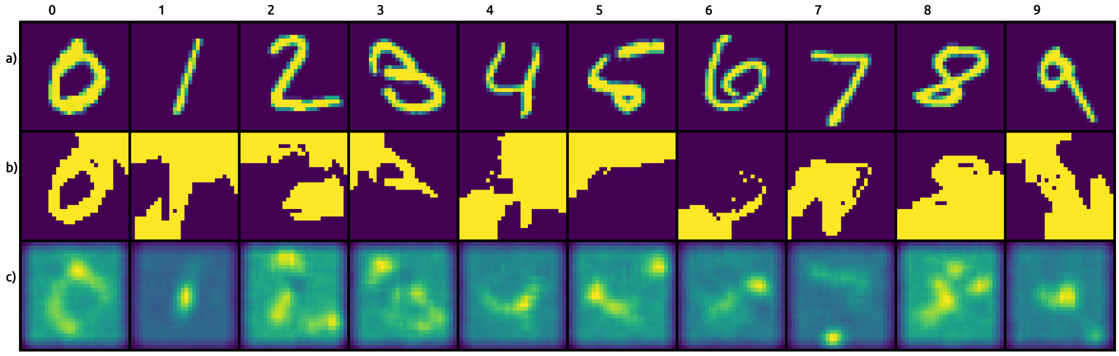


Figure 5.4: MNIST on Ridge Regression qualitative experiment: a) original images; b) LIME explanations; c) FRWI explanations

The next scenario of the experiment is the Ridge Regression as can be seen in the Figure 5.4. Same as before, it was trained as can detailed in the Subsection 5.1.3. And the lines mean the same as before. This time FRWI explanations cover lesser regions than before, probably indicating that the model focus on small regions. The explanations from FRWI in this experiment have one or two dots with a high degree of relevance, which demonstrates the behaviour of the model. And again, LIME is very hard to make it clear. The only case that it is possible to understand is class 0 where LIME distinguishes the shape; also the class 3 it has a piece of the number.

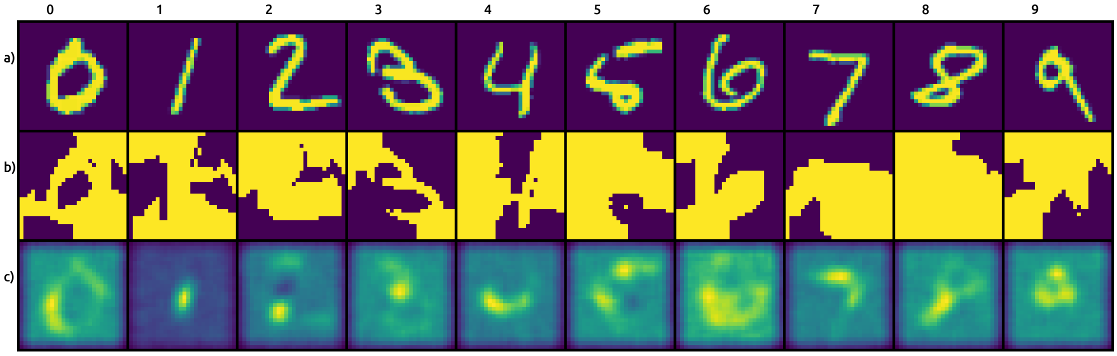


Figure 5.5: MNIST on Random Forest qualitative experiment: a) original images; b) LIME explanations; c) FRWI explanations

The learning model of the experiment in the Figure 5.5 is the Random Forest. Its training is described in Subsection 5.1.3. The FRWI explanations from this model have some similarities with the Ridge Regression, where they focus on some dots, but this case has some small areas also. This way, it is possible to see some concepts learned by the model, like the small dot in the middle of the image to indicate the class 1. In this scenario is where LIME is more confused, covering almost the entire image as relevant in all cases.

The numbers are simple concepts to the human perspective, so this scenario helps us to understand how the explainable models act and where they can be useful. A general aspect that happens in the FRWI explanations is the less relevant pixels are all too much similar. The main reason to cause this, it is the calculation of membership functions described in the Section 4.3, which restricts the range of values to the range of the data generated. Despite this drawback, this restrictions is important to make the FRWI adapt to different scenarios and not become a biased model. LIME has some tendency to focus on the shape in these scenarios, but in most of the cases, the relevant regions are too big to be comprehensible.

Fashion MNIST

Now for the Fashion MNIST. The classes are not as simple as the previous data set, but it still somewhat clean data. All models were trained as described in Section 5.1.3 and the explainable models in Section 5.2.1.

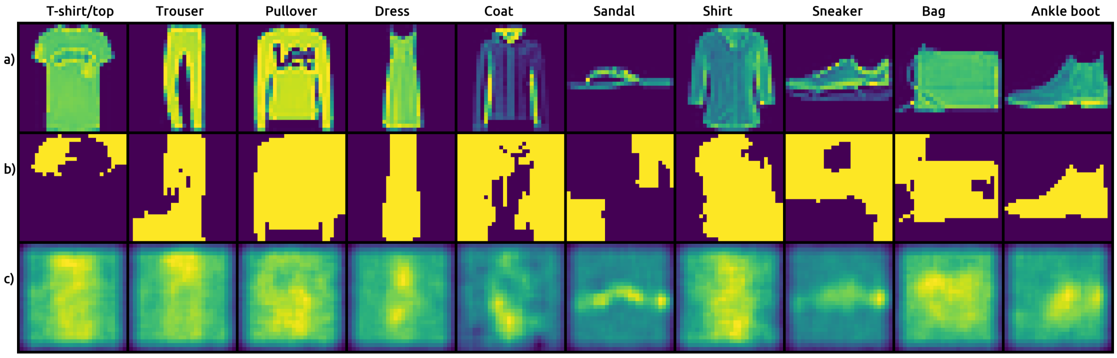


Figure 5.6: Fashion MNIST on WiSARD qualitative experiment: a) original images; b) LIME explanations; c) FRWI explanations

This time, the explanations from FRWI in the second line were not so clear, as shown in the Figure 5.6. It produces some clouds of relevant regions, but it shows us how WiSARD focuses on shapes. Especially in the case of the sandal, where it is possible to see the mould more clearly. LIME performs better in the classes dress and ankle boot, where it captures the shape very well. In the class pullover, also, it is possible to see the shape, but not so clear in other cases. LIME, still, selects the most of image as relevant, but its ability to capture shapes, is highlighted in this scenario a little more. At least both explainable models help us to understand that WiSARD is focusing on shape. Furthermore, the FRWI explanations show us that it is not often clear how WiSARD differentiates classes.

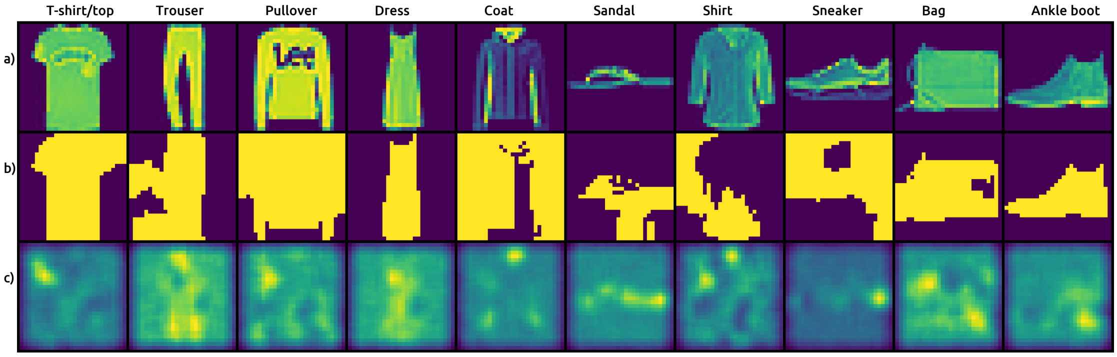


Figure 5.7: Fashion MNIST on Ridge Regression qualitative experiment: a) original images; b) LIME explanations; c) FRWI explanations

The Figure 5.7 is the experiment over the Ridge Regression. This scenario is one of the more interesting ones because it shows clearly how LIME from FRWI diverge. Here, LIME is focusing on the shape of the classes learned by the model, but FRWI is only focusing on what is relevant to the model, where it shows some dots with the highest degree of relevance in the image.

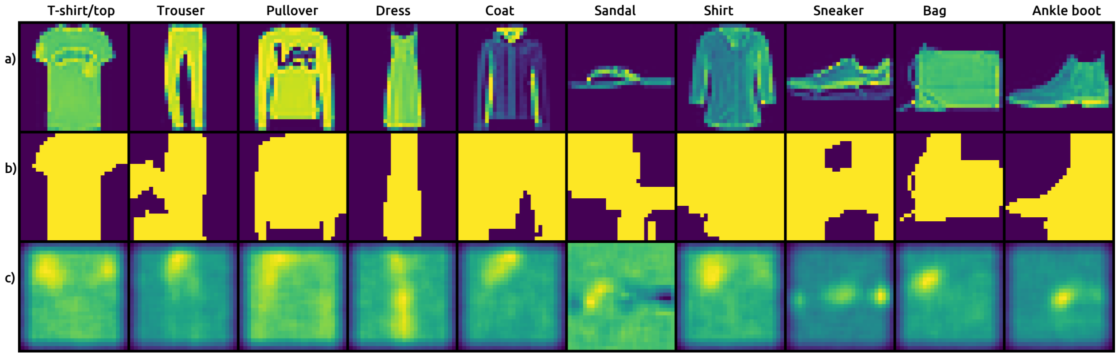


Figure 5.8: Fashion MNIST on Random Forest qualitative experiment: a) original images; b) LIME explanations; c) FRWI explanations

Finally, the last experiment over the Random Forest - in the Figure 5.8 - shows us a similar situation as the previous, of the Ridge Regression experiment, with LIME focing on shape. This time, FRWI is not focusing on dots, but on some small areas. We can see how FRWI explanations expose the small areas that weigh heavily on classifications made by models.

This data set reveals more information about the behaviour of the explainable models. This time, LIME has a good explanation based on mould, but this not teach us too much about the learning model.

Here only the positive explanation was analysed, but there is another view on the explanation scenario useful, then it is the negative explanation. In the next section, we show an introductory version of what such analysis could encompass.

5.2.3 Comparison of the negative explanation

The negative explanation is on a premature stage. So a special case was constructed to make some qualitative analysis on it. Furthermore, the positive explanations were generated as well, to give us a complete view of this scenario.
In this scenario, one image from each class in the MNIST data set was select to compose a new set. This is necessary to create information that confuses the learning

models in the classification process, and so see if the explainable models are capable of identifying these regions. Then, for each image had three types of noise inserted:

- The point noise, where a dot is drawn on the corner of the image, not affecting the main information and being a little disturbance (the letter ‘p’ is used to represent this modification).
- The external line, where a line is drawn around the number shape, as before not affecting the main information, but being a big disturbance (the letter ‘e’ is used to represent this modification).
- The cut line, where the number shape is cut in the middle, this one affects the main information and probably is the most difficult situation to the learning models (the letter ‘c’ is used to represent this modification).

For each noise, a new image is created, so some up the noise images and the original images, this new set has 40 examples.

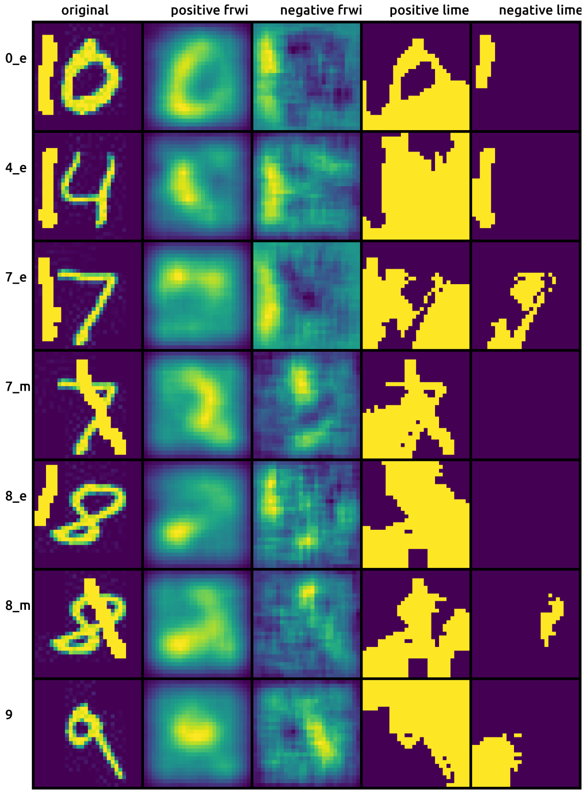


Figure 5.9: MNIST on WiSARD qualitative negative experiment

The learning models were trained with the MNIST data set to evaluate them with these noised images. The Figure 5.9 shows us some examples of negatives explanations from both explainable models over the WiSARD. The complete result of this experiment, including other models, can be seen in the Appendix B.1.2.

This experiment shows that FRWI is capable of identifying some added noise and also removing them from positive explanations. In the case of LIME, there are few cases where it finds some noise as a negative explanation. Still, it is very consistent in separating positive and negative explanations, as opposed to FRWI. FRWI has the drawback that it is needed to generate at least a 100000 examples before clear regions become visible, less than this only shows clouds of pixels, often meaning nothing to a human perspective.

In the previous subsection, each experiments goes over a small amount of cases, so to ratify these results, a quantitative experiment is essential. Therefore the next section details the quantitative experiments.

5.3 Quantitative experiments

In this section the experiments evaluated with the interpretation capacity score, as defined in Equation 5.1, are described, where the explainable models will be exposed to three different learning models in two data set as specified in Subsection 5.2.1. In these scenarios, it will be possible to see the performance of the models in controlled scenarios where it has a wide range of difficulty of explanations for the models.

5.3.1 Organisation of the experiment

As the qualitative experiment, the local mental images are generated, but this time these images are evaluated with the interpretation capacity score. To achieve that, some setup is required. In these experiments, only the positive explanations are evaluated, because the negative explanations are premature, and need more development.

The FRWI and LIME have the same configuration as in the qualitative experiments, but in the case of the number of examples produced was tested with two values as the next subsection shows.

First, it needs to calculate the mean of each parameter from the ICS, to do that, the basic evaluation from ICS is applied over the entire test set. Thus, it gets one value from ICS. Furthermore, to evaluate the consistency of this result, this experiment is repeated ten times, where each time the learning models are trained again. It is important to point out that the LIME and the FRWI are tested over the same trained learning model each time the ICS is calculated.

Another minor step needed for these experiments is to transform the output from FRWI to apply it as a filter of relevant regions because its output varies between zero and one. Thus, a cut based on the mean plus the standard deviation is applied by each output, where when it is over this value, it becomes one and zero otherwise. For the case of the application of the explanation as a filter, in both explainable models, it keeps the original value when one occurs, and it writes zero otherwise. The results of these experiments are detailed in the next section.

The environment used to run the experiments had the operational system Ubuntu version 18.04, and Python version 3.6. The following Python libraries were using: Scikit-Learn version 0.23.0, to apply the learning models, LIME version 0.2.0.0, to apply explanations of LIME, Pillow, version 7.0.0, to produce the images, and finally, Seaborn version 0.10.0, to produce the graphics. The WiSARD and Regression WiSARD were implemented in C++ with a Python wrapper as described in the sections 3.1,3.3 respectively. The source code of the model FRWI was built in C++ with a Python wrapper, so to take advantage of the C++ optimisations while retaining the clarity offered by Python scripting. The experiments were executed in a machine with a Ryzen 7 processor (16 cores) and 32GB of RAM memory. The complete execution of all experiments took 2 weeks.

5.3.2 Comparison of the positive explanation

In this comparison, the ICS was applied over the MNIST to compare the performance of the explainable models in a quantitative view, and also over the Fashion MNIST. Two values of examples produced were tested: 1000 and 10000, and these two cases were combined with the three learning models: WiSARD, Ridge Regression, Random Forest. Thus, it has two graphics for each data set: one for the number of examples 1000; and the other 10000.

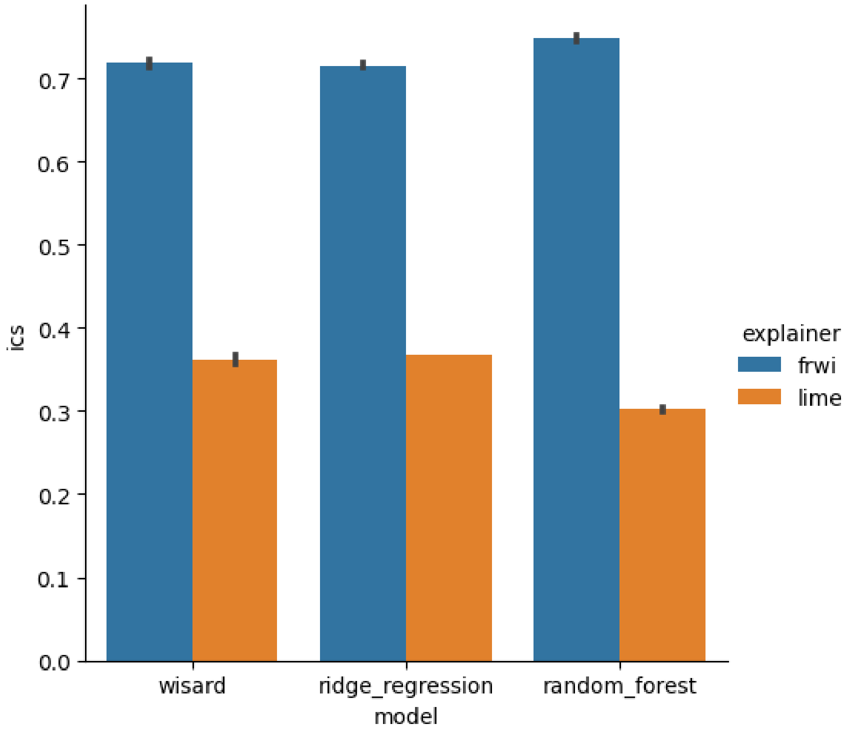


Figure 5.10: MNIST – ICS; 1k examples on explainable models

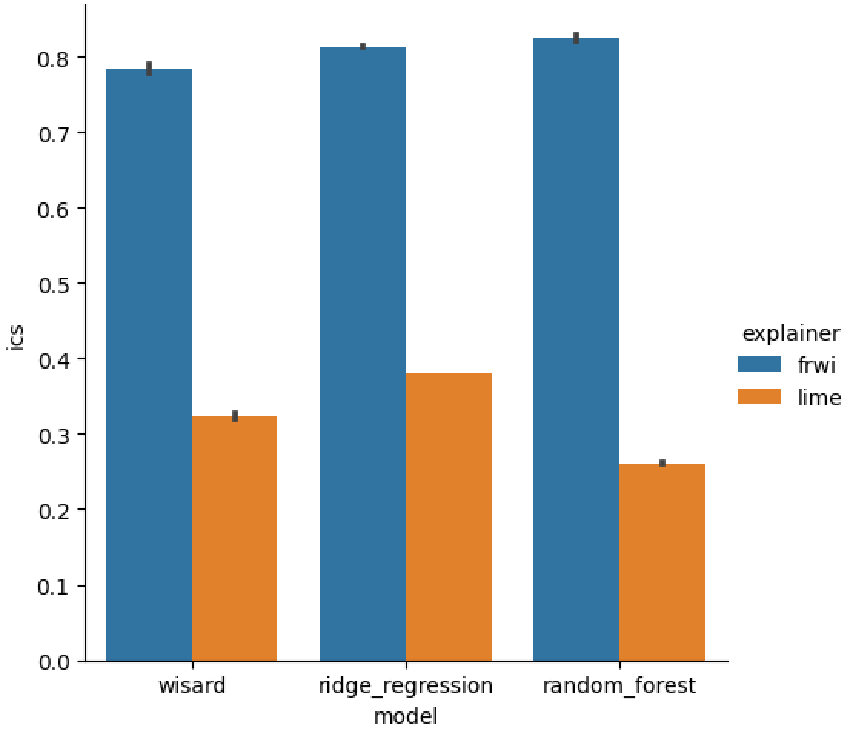


Figure 5.11: MNIST – ICS; 10k examples on explainable models

The Figure 5.10 shows the case of 1000 examples on the MNIST data set, where it is possible to see the FRWI winning in all cases. Also, the same occurs in the

Figure 5.11 of the case with 10000 examples. This shows the consistency with the previous analysis in qualitative experiments. The likely main reason for LIME's worse performance is its propensity to select big regions as relevant in the image. In both figures, the black vertical line shows the size standard deviation.

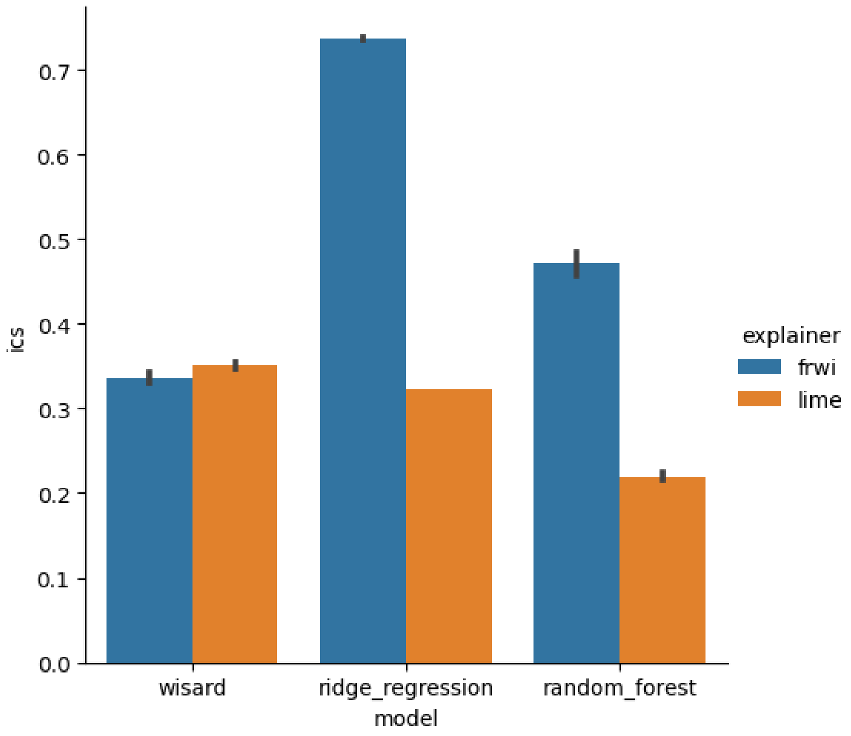


Figure 5.12: Fashion MNIST – ICS; 1k examples on explainable models

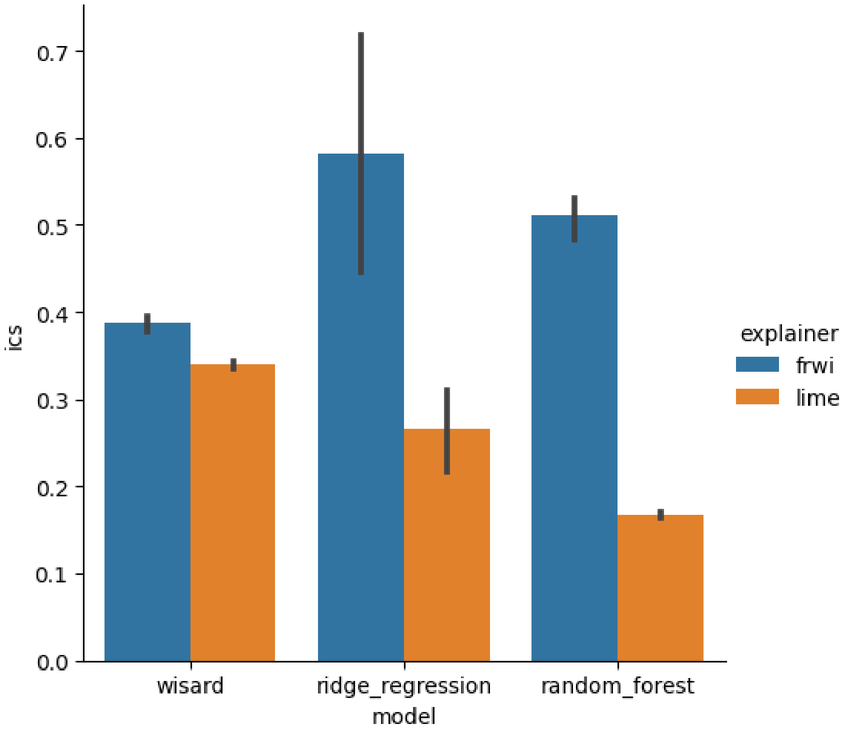


Figure 5.13: Fashion MNIST – ICS; 10k examples on explainable models

Here in the Figure 5.12 the case of 1000 examples on the Fashion MNIST data set, there is a statistic draw between LIME and FRWI, but the new approach is winning in other cases. This draw is probably because of the cloud regions determined by FRWI, where it was not very precise in its definition of what is relevant. In this data set, the WiSARD model has more colliding patterns between classes, which causes this situation. In the case of 10000 examples as the Figure 5.13 shows, the FRWI models wins all, probably because it produces a more precise explanation with more information. Despite this, the standard deviation is too high as the black vertical line shows. This is the drawback in being more precise because this can lead the explainable models to some mistakes.

After all these experiments, this work comes to its conclusions in next chapter. Reviewing the entire work and defining the next steps in this research.

Chapter 6

Final Considerations

This concluding chapter summarises the work developed and how it achieved results, and suggests next steps for this line of research.

As AI based decision systems are more frequently employed and in ever many areas, the need for eXplainable Artificial Intelligence grows, making this a rich research area. There are an abundance of methods of achieving explanations, depending on scenarios. The visualisation of relevant regions, as presented in earlier sections, is one of many of those methods. It can help us understand what the classifier considers important, and in the best cases can produce general views of the concepts learned by the models or a local view that focuses on one classification. This work was focused on the local view, as the LIME model is.

LIME was one of the first proposals to solve this problem of finding relevant regions. Its technique is to produce random examples using a specific approach, in order to optimise a linear model and thus find relevant regions. It is a good strategy for some scenarios, but it fails to achieve useful results in many others. For that reason it is important to develop and improve different methods.

The new approach, FRWI, can help fill the gap, producing explanations to models such as the WiSARD, or rule-based models like the Random Forest. FRWI also produces several examples to analyse the relevance to classifiers. By using fuzzy logic, it produces clouds of relevance, as opposed to LIME's crisp, discreet outlines, which is useful in yet other scenarios.

We reiterate, however, that although both of these explainers are *agnostic* and will attempt to produce explanations to any model, there is no guarantee they will produce useful explanations in every scenario. Analogous to the fact that not any single classifier model solves every problem, and are better suited for certain scenarios and data sets, each explainer model also suits specific classifiers and data sets.

The qualitative experiments showed the model is capable of producing reasonable explanations, which can help understand and improve classifiers. Also, in the scenarios examined, in comparison to LIME, its explanations are more human friendly and useful. As these experiments only show a small portion of the chosen datasets, the quantitative experiments make a more global evaluation of the models.

These quantitative evaluations use the new Interpretation Capacity Score, also introduced in this work, as an objective way to compare and assess the performance of different explainers. The ICS evaluations have shown some of the model's tendencies, such as LIME often selecting nearly the entire image as relevant in the experiments ran. Such a tendency can seem obvious, but with this we have a clear indicator and metric to highlight and show it. Moreover, that FRWI performed better by a large margin strongly suggests it is a noteworthy explainable model that could be the focus of future research.

This work present as well the new FRWI approach to produce explanations to the WiSARD classifier, which also shows promising results. It is a method that is compatible with the model structure and capable of translating WiSARD decision processes in a simple visual way.

Among the many possible points for further research, we highlight the need to continue testing FRWI in more scenarios, such as using colouring images instead of just greyscale. In order continue to further the understanding of in which cases these explainers are viable, more experiments in different datasets should be conducted. Another possibility is to extend FRWI to also work with tabular data, in which case a new set of fuzzy rules would likely be needed as well as a new way to present the explanations. It could also be interesting to extend it to work with regression models, though that can be even more complex in the same two points, as well as other adjustments that will inevitably be necessary to refine the explanations produced.

There is also a more specific point for future works, such as the application of new fuzzy rules to capture different information. For instance, looking for regions which confuse the classifier. It would be a variation of negative rules, which could help in the improvement of the learning models, with the use of the produced explanations. As can be easily seen, it should be possible to capture any desired information as long as it is possible to express it in fuzzy logic. Also, it is possible to automate the production of rules in order to generate explanations with rules, which would be more suitable to the context of tabular data or textual data. One last point of possible research is to research substitution of the step of evaluation of images to mechanisms other than fuzzy logic.

References

[1] ALEKSANDER, I., THOMAS, W. V., BOWDEN, P. A., 1984, "WISARD, a radical step forward in image recognition", *Sensor review*, v. 4, n. 3, pp. 120–124.

[2] ARRIETA, A. B., DÍAZ-RODRÍGUEZ, N., DEL SER, J., et al., 2020, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI", *Information Fusion*, v. 58, pp. 82–115.

[3] BHARGAVA, A. K., 2013, *Fuzzy set theory fuzzy logic and their applications*. S. Chand Publishing.

[4] BLEDSOE, W. W., BROWNING, I., 1959, "Pattern Recognition and Reading by Machine". In: *Eastern Joint IRE-AIEE-ACM Computer Conference*, p. 225–232, New York, NY, USA, 03.

[5] BUITINCK, L., LOUPPE, G., BLONDEL, M., et al., 2013, "API design for machine learning software: experiences from the scikit-learn project". In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 03.

[6] BURATTINI, E., DE GREGORIO, M., TAMBURRINI, G., 2000, "Mental imagery in explanations of visual object classification". In: *Proceedings. Vol. 1. Sixth Brazilian Symposium on Neural Networks*, pp. 137–143. IEEE, 3.

[7] BYRNE, R. M., 2019, "Counterfactuals in Explainable Artificial Intelligence (XAI): Evidence from Human Reasoning." In: *IJCAI*, pp. 6276–6282, 3.

[8] CARDOSO, D., DE GREGORIO, M., LIMA, P., et al., 2012, "A weightless neural network-based approach for stream data clustering". In: *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 328–335. Springer, 03.

[9] CARDOSO, D. D. O., LIMA, P. M. V., DE GREGORIO, M., et al., 2011, "Clustering data streams with weightless neural networks". In: *ESANN*

2011, 19th European Symposium on Artificial Neural Networks, pp. 201 – 206, Bruges, Belgium, 03.

[10] CARDOSO, D. O., CARVALHO, D. S., ALVES, D. S. F., et al., 2016, "Financial Credit Analysis via a Clustering Weightless Neural Classifier", *Neurocomputing*, v. 183, n. C (mar.), pp. 70–78. ISSN: 0925-2312. doi: 10.1016/j.neucom.2015.06.105. Disponível em: <`https://doi.org/10.1016/j.neucom.2015.06.105`>.

[11] CARNEIRO, H. C. C., FRANÇA, F. M. G., LIMA, P. M. V., 2015, "Multilingual part-of-speech tagging with weightless neural networks", *Neural Networks*, v. 66, pp. 11–21.

[12] CHEN, T., GUESTRIN, C., 2016, "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, pp. 785–794, 03.

[13] DAS, A., RAD, P., 2020, "Opportunities and challenges in explainable artificial intelligence (xai): A survey", *arXiv preprint arXiv:2006.11371*.

[14] DE SOUZA, D. F. P., CARNEIRO, H. C. C., FRANÇA, F. M. G., et al., 2013, "Rock-paper-scissors WiSARD". In: *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI 2013 & CBIC 2013)*, pp. 178–182, 03.

[15] DE SOUZA, D. F. P., FRANÇA, F. M. G., LIMA, P. M. V., 2014, "Spatio-temporal pattern classification with KernelCanvas and WiSARD". In: *2014 Brazilian Conference on Intelligent Systems (BRACIS 2014)*, pp. 228–233. IEEE, 03.

[16] FILHO, A. L., GUARISA, G. P., FILHO, L. A. D. L., et al., 2020, "Interpretation of Model Agnostic Classifiers via Local Mental Images", *European Symposium on Artificial Neural Networks*.

[17] FILHO, L. A. D. L., OLIVEIRA, L. F. R., FILHO, A. L., et al., 2019, "Prediction of palm oil production with an enhanced *n*-tuple regression network". In: *Proceedings of the 27th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 301–306, 3.

[18] GRIECO, B. P. A., LIMA, P. M. V., DE GREGORIO, M., et al., 2008, "Extracting fuzzy rules from "mental" images generated by modified WiSARD perceptrons". In: *Proc. E*, v. 26, pp. 101–773, 03.

[19] GRIECO, B. P. A., LIMA, P. M. V., GREGORIO, M., et al., 2010, "Producing pattern examples from "mental" images", *Neurocomputing*, v. 73, n. 7-9, pp. 1057–1064.

[20] GROTHER, P., HANAOKA, K., 2016, "NIST special database 19 handprinted forms and characters 2nd Edition", *National Institute of Standards and Technology, Tech. Rep.*

[21] GUNNING, D., 2017, "Explainable artificial intelligence (xai)", *Defense Advanced Research Projects Agency (DARPA), nd Web*, v. 2, n. 2.

[22] HO, T. K., 1995, "Random decision forests". In: *Proceedings of 3rd international conference on document analysis and recognition*, v. 1, pp. 278–282. IEEE, 03.

[23] IN DATABASES COMPETITION, B. K. D., 2020. "KDD BR Competition 2018". Disponível em: <`https://www.kaggle.com/c/kddbr-2018/`>. Last accessed 19 october 2020.

[24] KOLCZ, A., ALLINSON, N. M., 1996, "*n*-tuple Regression Network", *Neural Networks*, v. 9, n. 5, pp. 855–869.

[25] KRISHNA, K., MURTY, M. N., 1999, "Genetic K-means algorithm", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, v. 29, n. 3, pp. 433–439.

[26] LECUN, Y., CORTES, C., 1998, "The MNIST database of handwritten digits", *http://yann. lecun. com/exdb/mnist/*.

[27] LIMA FILHO, A., LUSQUINO FILHO, L. A. D., FRANÇA, F. M., et al., 2019, ""What are you thinking?"-Explanation and interpretation by an artificial consciousness system", *Creativity 2019*.

[28] LIMA FILHO, A. S., GUARISA, G. P., LUSQUINO FILHO, L. A., et al., 2020, "wisardpkg–A library for WiSARD-based models", *arXiv*, pp. arXiv–2005.

[29] LUSQUINO FILHO, L. A. D., LIMA FILHO, A., FRANÇA, F. M. G., et al., 2020, "A weightless emotion-driven architecture for planning tasks", *26th IEEE International Symposium on High-Performance Computer Architecture*.

[30] LUSQUINO FILHO, L. A. D., FRANÇA, F. M. G., LIMA, P. M. V., 2018, "Near-optimal facial emotion classification using WiSARD-based weightless system". In: *Proceedings of the 26th European Symposium on Artificial*

Neural Networks, Computational Intelligence and Machine Learning, pp. 85–90, 03.

[31] LUSQUINO FILHO, L. A. D., GUARISA, G. P., LIMA FILHO, A., et al., 2019, "Classifying Actions Units with ClusWiSARD". In: *Proceedings of the 28th International Conference on Artificial Neural Networks*, 03.

[32] LUSQUINO FILHO, L. A. D., GUARISA, G. P., OLIVEIRA, L. F. R., et al., 2019, "Action Units Classification Using ClusWiSARD". In: *International Conference on Artificial Neural Networks*, pp. 409–420. Springer, .

[33] LUSQUINO FILHO, L. A. D., OLIVEIRA, L. F. R., CARNEIRO, H. C. C., et al., 2020, "A weightless regression system for predicting multi-modal empathy". In: *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)(FG)*, pp. 554–558, .

[34] LUSQUINO FILHO, L. A. D., OLIVEIRA, L. F. R., LIMA FILHO, A., et al., 2020, "Extending the Weightless WiSARD Classifier for Regression", *Neurocomputing.*

[35] MOLNAR, C., 2018, "A guide for making black box models explainable", *URL: https://christophm. github. io/interpretable-ml-book.*

[36] OLIVEIRA, L. F. D. R. D., 2017, *Comparação de desempenho entre os modelos neurais ágeis ELM e WiSARD.* Tese de Mestrado, Universidade Federal do Rio de Janeiro, 3.

[37] OM, A. B. O., 2001, "Ridge regression and inverse problems", *Stockholm University, Department of Mathematics.*

[38] RANGEL, F. M., DE FARIA, F. F., LIMA, P. M. V., et al., 2016, "Semi-Supervised Classification of Social Textual Data Using WiSARD". In: *Proceedings of the 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 165–170, 03.

[39] RIBEIRO, M. T., SINGH, S., GUESTRIN, C., 2016, "Why should i trust you?: Explaining the predictions of any classifier". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144. ACM, 03.

[40] SELVARAJU, R. R., COGSWELL, M., DAS, A., et al., 2017, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *ICCV*, pp. 618–626, 03.

[41] SHCHERBAKOV, M. V., BREBELS, A., SHCHERBAKOVA, N. L., et al., 2013, “A survey of forecast error measures”, *World Applied Sciences Journal*, v. 24, n. 24, pp. 171–176.

[42] TJOA, E., GUAN, C., 2019, “A survey on explainable artificial intelligence (XAI): towards medical XAI”, *arXiv preprint arXiv:1907.07374*.

[43] VIDAL, F. S., CARNEIRO, H. C. C., ROSA, P. F. F., et al., 2013, “Identificação de emoções a partir de expressões faciais com redes neurais sem peso”. In: *Proceedings of XI SBAI – Simpósio Brasileiro de Automação Inteligente (In Portuguese)*, 03.

[44] XIAO, H., RASUL, K., VOLLGRAF, R., 2017, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms”, *arXiv preprint arXiv:1708.07747*.

[45] ZADEH, L. A., 1979, “Fuzzy sets and information granularity”, *Advances in fuzzy set theory and applications*, v. 11, pp. 3–18.

[46] ZADEH, L. A., 1965, “Fuzzy sets”, *Information and control*, v. 8, n. 3, pp. 338–353.

Appendix A

Works accepted

Following is the list of papers accepted for publication during the development of this work:

A.1 Journal Articles

1. LUSQUINO FILHO, L. A. D.; OLIVEIRA, L. F. R. ; LIMA FILHO, A. ; GUARISA, G. ; FELIX, L. M. ; LIMA, P. M. V. ; FRANÇA, F. M. G. . Extending the Weightless WiSARD Classifier for Regression. NEUROCOMPUTING, 2020.

A.2 Book chapters

1. LUSQUINO FILHO, L. A. D.; OLIVEIRA, L. F. R.; CARNEIRO, H. C. C., GUARISA G. P.; LIMA FILHO, A. S.; FRANÇA, F. M. G. ; LIMA, P. M. V., A Weightless Neural System for Empathy Prediction - Accepted for OMG-Challenges Book, Knowledge Technology Group, Springer, 2020.

A.3 Complete works published in proceedings of conferences

1. LIMA FILHO, A. ; GUARISA, G. ; LUSQUINO FILHO, L. A. D. ; OLIVEIRA, L. F. R. ; COSENZA, C. ; FRANÇA, F. M. G. ; LIMA, P. M. V. . Interpretation of Model Agnostic Classifiers via Local Mental Images. In: European Symposium on Artificial Neural Networks, 2020, Brugge. Proc. of ESANN 2020, 2020.

2. LUSQUINO FILHO, L. A. D.; OLIVEIRA, L. F. R. ; LIMA FILHO, A. ;

GUARISA, G. ; LIMA, P. M. V. ; FRANÇA, F. M. G. . Prediction of palm oil production with an enhanced n-Tuple Regression Network. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2019, Bruges. Proc of ESANN 2019, 2019. v. 27. p. 301-306.

3. LUSQUINO FILHO, L. A. D.; GUARISA, G. ; OLIVEIRA, L. F. R. ; LIMA FILHO, A. ; FRANÇA, F. M. G. ; LIMA, P. M. V. Action Units Classification Using ClusWiSARD. In: International Conference on Artificial Neural Networks, 2019, Munich. Artificial Neural Networks and Machine Learning, ICANN 2019: Image Processing, 2019. p. 409-420.

A.4 Extended abstracts published in proceedings of conferences

1. LUSQUINO FILHO, L. A. D.; OLIVEIRA, L. F. R.; CARNEIRO, H. C. C., GUARISA G. P.; LIMA FILHO, A. S.; FRANÇA, F. M. G. ; LIMA, P. M. V., *A weightless regression system for predicting multi-modal empathy*, Workshop Affective Behavior Analysis in-the-wild, Proceedings of the 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020), Buenos Aires, Argentina, 2020.

2. LIMA FILHO, A.; LUSQUINO FILHO, L. A. D. ; FRANÇA, F. M. G. ; LIMA, P. M. V., ’What are you thinking?’ - Explanation and interpretation by an artificial consciousness system. In: Creativity 2019, 2019, Rio de Janeiro. Creativity 2019, 2019.

3. LUSQUINO FILHO, L. A. D. ; LIMA FILHO, A. ; FRANÇA, F. M. G. ; LIMA, P. M. V., A weightless emotion-driven architecture for planning tasks. In: CogArch 2020, 2020, San Diego, CA. 26th IEEE International Symposium on High-Performance Computer Architecture, 2020.

A.5 First page of each published works

A weightless regression system for predicting multi-modal empathy

Leopoldo A. D. Lusquino Filho[1], Luiz F. R. Oliveira[1], Hugo C. C. Carneiro[1], Gabriel P. Guarisa[1], Aluízio Lima Filho[1], Felipe M. G. França[1] and Priscila M. V. Lima[1] [2]
[1] PESC/COPPE, [2] NCE - Universidade Federal do Rio de Janeiro, RJ, Brazil

***Abstract*— This work takes into account the benefits of machine learning in order to estimate the valence of emotions on the OMG Empathy dataset, considering the information obtained from face expressions and dialogue of interlocutors. RegressionWiSARD and ClusRegressionWiSARD n-tuple regressors and its ensembles were employed to this end. The best performance achieved among all the combinations of weightless neural models considered (evaluated using the CCC metric) was 0.25 in validation set of the Personalized Track .**

I. Introduction

Since emotional states are a fundamental part of the core of human psychology, often exceeding the intellect itself in psychological hierarchy, it is natural that Affective Computing[11] occupies a prominent place in the study of the human-machine interface. Along with the apogee of machine learning, Affective Computing has experienced great growth in recent years, but it still has many open questions. Some of the main ones involve the prediction of emotions based on information from many different sources and the identification of subtle emotional states in real time. Specifically, many advances have been made recently in the area of affective prediction[25][26][27][28][29][30][31][32].

In order to offer a significant contribution in the area, this paper discusses the use of weightless neural network ensembles in predicting the affective valence of individuals in conversation videos, since this type of model has computational simplicity, great computational agility and ease of being parallelized.

The structure of this work is as follows: in Section 2 the WiSARD weightless model, some of its extensions and ensembles will be described, Section 3 deals with the preprocessing of data from different sources, Section 4 describes the experiments carried out with different types of ensembles using uni and multimodal data and discusses their results, and Section 5 is the conclusion of this work, summarizing all the previous discussion and also offering the main ongoing works.

II. n-Tuple models

The n-Tuple classifier is a boolean node pattern classifier [3], which distances itself from models derived from perceptron because it do not use synaptic weights between their neurons, thus avoiding all training time required for their convergence. n-Tuple classifier does not need any parameter fine tuning, nor does it use any error minimization technique to obtain generalization in pattern learning [17]. The family of models derived from the n-Tuple classifier is known as Weightless Artificial Neural Network (WANN).

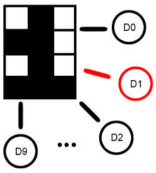


Fig. 1. The WiSARD model multidiscriminator structure. For digits recognition task there are ten discriminators. In the training phase, only the corresponding discriminator is accessed.

A. WiSARD

WiSARD is a neural model based on the n-tuple classifier, where each neuron is equivalent to a piece of memory [1]. This model is class discriminator-oriented, where all discriminators are formed by N RAM-neurons, whose memory addresses are addressed by n-bit tuples. Each neuron has 2^n memory locations.

WiSARD works with binary standards, requiring the use of some preprocessing technique to form data suitable to the model before the training and classification process. The training process consists of using the binary input to access specific memory positions of the corresponding discriminator and increment the counter that constitutes its content. During the classification, all discriminators are accessed and they are assigned a score formed by the number of non-null positions accessed. The discriminator with the highest score will determine the class of the entry and in case of a tie, a threshold called bleaching, which is initialized to zero, is increased and the classification is repeated, considering for the score only memory locations whose counter has higher value than bleaching. This procedure is repeated until there is a winning discriminator or until the bleaching value exceeds the highest counter among the memory locations accessed, in which case a default class is chosen for the entry. The structure and the training process in WiSARD are illustrated in Figs. 1 and 2. WiSARD can be used to accelerate the training of deep models, and can be used as a starting layer for such neural networks in a hybrid hierarchy[33].

This work was not supported by any organization

Action Units Classification using ClusWiSARD

Leopoldo A. D. Lusquino Filho[1]*, Gabriel P. Guarisa[1], Luiz F. R. Oliveira[1],
Aluizio Lima Filho[1], Felipe M. G. França[1], and Priscila M. V. Lima[2]

1- PESC/COPPE 2- NCE
Universidade Federal do Rio de Janeiro, RJ, Brazil **
lusquino@cos.ufrj.br

Abstract. This paper presents the use of WiSARD and ClusWiSARD weightless neural networks models for the classification of the contraction and extension of *Action Units*, the facial muscles involved in emotive expressions. This is a complex problem due to the large number of very similar classes, and because it is a multi-label classification task, where the positive expression of one class can modify the response of the others. WiSARD and ClusWiSARD solutions are proposed and validated using the CK+ dataset, producing responses with accuracy of 89.66%. Some of the major works in the field are cited here, but a proper comparison is not possible due to a lack of appropriate information about such solutions, such as the subset of classes used and the time of training/testing. The contribution of this paper is in the pioneering use of weightless neural networks in an AUs classification task, in the unpublished application of the WiSARD and ClusWiSARD models in multi-label tasks and in the new unsupervised expansion of ClusWiSARD proposed here.

Keywords: Action Units, WiSARD, ClusWISARD, weightless neural network

1 Introduction

Ekman and Friesen [1] cataloged a set of muscles known as *Action Units* (AUs) – which would be responsible for all facial expressiveness – while attempting to obtain a set of universal emotions present in any human. The automatic identification of these AUs has been developed since the mid-1990s and has several applications: forensics, psychological treatment, physical therapy support and advertising feedback, among others. AUs have also been used in the development of adaptive digital avatars [2].

Some of the great difficulties in automatic detection of AUs are the large number of classes and the wide variety of forms how AUs express themselves, besides the fact that they usually manifest together, making this a hard multi-label task. In this way, the approaches that are emerging in the literature usually

* Corresponding author

** This work was partially supported by CAPES, CNPq, FAPERJ and FINEP, Brazilian research agencies.

Neurocomputing 416 (2020) 280–291

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Extending the weightless WiSARD classifier for regression

Leopoldo A.D. Lusquino Filho[a,1,*], Luiz F.R. Oliveira[a,1], Aluizio Lima Filho[a], Gabriel P. Guarisa[a], Lucca M. Felix[b], Priscila M.V. Lima[a,c], Felipe M.G. França[a]

[a] *PESC/COPPE, Universidade Federal do Rio de Janeiro, RJ, Brazil*
[b] *DCC, Universidade Federal do Rio de Janeiro, RJ, Brazil*
[c] *NCE, Universidade Federal do Rio de Janeiro, RJ, Brazil*

ARTICLE INFO

Article history:
Received 15 July 2019
Revised 11 December 2019
Accepted 12 December 2019
Available online 8 April 2020

Communicated by Dr. Oneto Luca

Keywords:
Regression WiSARD
WiSARD
ClusWiSARD
n-Tuple classifier
n-Tuple Regression Network
Ensemble
Online learning

ABSTRACT

This paper explores two new weightless neural network models, Regression WiSARD and ClusRegression WiSARD, in the challenging task of predicting the total palm oil production of a set of 28 (twenty eight) differently located sites under different climate and soil profiles. Both models were derived from Kolcz and Allinson's *n*-Tuple Regression weightless neural model and obtained mean absolute error (MAE) rates of 0.09097 and 0.09173, respectively. Such results are very competitive with the state-of-the-art (0.07983), whilst being four orders of magnitude faster during the training phase. Additionally the models have been tested on three classic regression datasets, also presenting competitive performance with respect to other models often used in this type of task.

1. Introduction

Regression is a traditional and important machine learning task, since there is a wide range of practical situations in the real world where it is necessary to predict values in a continuous space. In a precision agriculture scenario, it would be desirable that simple devices, such as small sensors, could perform regression. Weightless Artificial Neural Networks (WANNs), due to its lean, RAM-based architecture, seem to be a suitable computational intelligence model for this type of task.

This paper explores the use of WANNs in the KDD18 competition [3], a challenge which goal is to predict the palm oil harvest productivity of a set of 28 (twenty eight) different production fields using data provided by an agribusiness company. The dataset contains information about palm trees varieties, harvest dates, atmospheric data during the development of the trees, and soil characteristics of the fields where the trees are located in. The WANN models explored in this work are based on the *n*-Tuple Regression Network [2], which was proved to be successful when compared to other classical regression approaches in non-linear plant approximation [33] and Mackey-Glass chaotic time series prediction tasks [32]. These WANN models were introduced in [5]. Here, a wider theoretical background is presented, alongside a broader exploration of their parameters and how the models perform when combined as ensembles.

The remainder of this text is organized as follows. Section 2 presents the basic models that inspired the new weightless regression ones: *n*-Tuple Classifier, WiSARD [1], ClusWISARD [8], and *n*-Tuple Regression Network [2]. Section 3 presents the two weightless models proposed for regression, and the ensemble techniques explored. Section 4 discusses the various approaches used in the KDD18 competition, as well as a comparison with state-of-the-art methods. This section also contains the description of experiments using the new models in the House Prices, CalCOFI, and Parkinson datasets. Concluding remarks and ongoing work are presented in Section 5.

2. *n*-Tuple Classifier and family

2.1. n-*Tuple Classifier*

The *n*-Tuple Classifier is a binary pattern classifier [4] based on Random Access Memories (RAMs), requiring no parameter fine tuning or any error minimization technique to achieve generalized learning patterns [34,35]. The basis of its operation is to use the

[*] Corresponding author.
E-mail address: lusquino@cos.ufrj.br (L.A.D. Lusquino Filho).
[1] Both authors had equal participation and are first author.

https://doi.org/10.1016/j.neucom.2019.12.134
0925-2312/© 2020 Elsevier B.V. All rights reserved.

“What are you thinking?” - Explanation and interpretation by an artificial consciousness system

Aluizio Lima Filho, Leopoldo A.D. Lusquino Filho, Felipe M. G. França, and Priscila M. V. Lima
PESC/COPPE
Universidade Federal do Rio de Janeiro – Brazil

One key requirement for complex intelligent systems is the ability to explain their decisions/actions, which constitutes the focus of eXplainable Artificial Intelligence (XAI) (Gunning, 2017). In the core of XAI lies the need to generate some interpretation of the AI system’s behaviour (Ribeiro et al., 2016). Explanation constitutes a process that may involve argumentational and emotional responses, aiming at persuading the person who receives the arguments (Molnar, 2019). In such a process there are two kinds of interpretations: (i) the first one is the interpretation made by the explainer in order to create arguments; (ii) the second one is the interpretation coined by the one who receives the argumentation to understand what was said.

In a conscious system, one can assume consciousness of its existence, consciousness of its perceptions, and so on (Aleksander, 1995). Explanation helps consciousness of oneself. So, the ability to create explanations is needed to make the system aware of itself and to help the system to create “logical thoughts”. The consciousness of itself could be made possible when the system generates interpretations of what it perceives, and with that create its argumentation to start the process of explanation. Therefore, it could create a line of reasoning that one could call “thinking”.

The first step endow a system with explanatory abilities is to create tools capable of producing interpretation as resources to the artificial consciousness system. In this direction, the ongoing research of interpretations with the WiSARD model (Aleksander et al., 1984) provides a tool capable of produce “mental” images as interpretational resources that could further be used by a conscious system.

Bibliography

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

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "Why should i trust you?: Explaining the predictions of any classifier." *Proceedings of*

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Interpretation of Model Agnostic Classifiers via Local Mental Images

Aluizio Lima Filho[1], Gabriel P. Guarisa[1], Leopoldo A.D. Lusquino Filho[1],
Luiz F. R. Oliveira[1], Carlos A. N. Cosenza[3],
Felipe M. G. França[1] and Priscila M. V. Lima[1,2] *

1–PESC/COPPE, 2–NCE, 3–PEP/COPPE
Universidade Federal do Rio de Janeiro, RJ, Brazil

Abstract. Although successful black-box learning models have been created, understanding what happens when a machine produces a classification response is still a challenge. This work introduces FRWI – Fuzzy Regression WiSARD Interpreter, a novel fuzzy rules-based algorithm that is capable of interpreting the responses of black-box classifiers via the production of local mental images from a WiSARD n-tuple classifier. FRWI is compared with LIME – Local Interpretable Model-Agnostic Explanations, a pioneering agnostic classification interpreter model. To make a quantitative evaluation of interpretable models, a new metric – Interpretation Capacity Score – is proposed. Using this metric, it is shown that FRWI surpasses LIME in producing coherent interpretations.

1 Introduction

The need to interpret responses from learning models gets higher in different situations [1]. Questions arise such as: how the models make the decision in the classification, or when to trust its process, and when not to do so. One way to answer the first question is to show what is relevant to the model. LIME [2] – Local Interpretable Model-Agnostic Explanations – was developed with the motivation to clarify such relevance. There are other interpreter models focused on DNNs, like Gran-Cam [3], that were later introduced in the literature. How ever, LIME does not have feasible interpretation capacity for all learning models, due to interpretable models have scenarios where they work better as learning models. Experimental tests were performed utilizing LIME to explain decisions made by following classifiers: WiSARD [4], Linear model [5] and Random Forest model [6] trained with images data sets. It will be shown that results will select too much in the image as relevant, and it will not let it clear what is happening inside the classifier.For that reason, the idea of creating a degree of relevance for each pixel in the image came as an alternative to interpret the responses of black-box classifiers more feasible. This work introduces FRWI – Fuzzy Regression WiSARD Interpreter, a WiSARD n-tuple classifier that produces local mental images, via a fuzzy rules-based algorithm, as an interpretation of the responses of black-box classifiers. To compare the interpretation capacity of both LIME and FRWI models, the Interpretation Capacity Score metric is defined.

*This work was partially supported by NGD Systems, Inc./COPPETEC grant PESC21713; CAPES, CNPq and FAPERJ, Brazilian research agencies.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

Prediction of Palm Oil Production with an Enhanced n-Tuple Regression Network

Leopoldo A. D. Lusquino Filho[1], Luiz F. R. Oliveira[1], Aluizio L. Filho[1],
Gabriel P. Guarisa[1], Priscila M. V. Lima[2], Felipe M. G. França[1] *

1- PESC/COPPE 2- NCE
Universidade Federal do Rio de Janeiro, RJ, Brazil

Abstract. This paper introduces Regression WiSARD and ClusRegression WiSARD, two new weightless neural network models that were applied in the challenging task of predicting the total palm oil production of a set of 28 differently located sites under different climate and soil profiles. Both models were derived from the n-tuple regression weightless neural model and obtained error (MAE) rates of 0.08737% and 0.08938%, respectively, which are very competitive with the state-of-art (0.07569), whilst being four (4) orders of magnitude faster during the training phase.

1 Introduction

Regression is one of the most important machine learning tasks, given the wide range of practical situations in the real world where it is necessary to predict values in a given continuum space. Due to its great utility, it is desirable that simple devices, such as small sensors, could perform regression with online training. Weightless artificial neural networks (WANNs), due to its lean, RAM-based architecture, seems to be ideal for this type of task.

This paper presents and explores the use of WANNs in the KDD18 competition [5], a challenge which goal is to predict the palm oil harvest productivity of a set of 28 different production fields using data provided by an agribusiness company. The dataset contains information about palm trees varieties, harvest dates, atmospheric data during the development of the trees, and soil characteristics of the fields where the trees are located in. The novel WANN models are based on the n-tuple Regression Network [3], which has been proved successful when compared to other classical regression approaches in non-linear plant approximation, and Mackey-Glass chaotic time series prediction tasks.

The remainder of this text is organized as follows: Section 2 presents the two weightless models proposed for regression, as well as the basic concepts behind the models that inspired it: WiSARD [1] and n-tuple Regression Network. Section 3 discusses the various approaches used in the KDD18 competition, as well as a comparison with state-of-the-art methods and other relevant results. Conclusion and future work are presented in Section 4.

*This work was partially supported by CAPES, CNPq, FAPERJ and FINEP, Brazilian research agencies.

WiSEMAN: A weightless emotion-driven neural architecture for planning-related tasks

Leopoldo A.D. Lusquino Filho, Aluizio Lima Filho,
Felipe Maia Galvão França, Priscila Machado Vieira Lima

PESC/COPPE
Universidade Federal do Rio de Janeiro – Brazil

ABSTRACT

Planning and management systems via multi-agent have become an increasingly demanding solution for many tasks involving heterogeneous data. Research into conscious agents has also shown solid progress. Here we propose an agent-based cognitive system with conscious-like behavior using Weightless Artificial Neural Networks. WiSEMAN is easy to embed on a wide range of devices due to low computational cost.

1. INTRODUCTION

The increasing number of devices collecting data and connecting to each other, exchanging information, makes concepts like IoT and BigData tangible. The emergence of efficient ways to deal with such huge heterogeneous mass of data in real time has become extremely necessary.In this scenario, distributed solutions, such as multi-agent planning, have been more and more highlighted. The ability of agents to be adaptive, agile in their negotiations and able to handle multiple types of tasks simultaneously is of fundamental importance in this type of system. Because of this, different types of cognitive architecture have been designed to structure the learning and action taking capabilities of such agents[1].

Another area of computing that has gradually developed is Machine Consciousness (MC)[2], which attempts to build systems that have subjectivity. One of MC's partial goals is to construct agents who behave indistinguishably from that of a truly conscious agent, even if unintentionally. In all of these systems, emotion modeling plays a vital role as a tool for improving agent training. This is to so expected, because all major theories of consciousness place great emphasis on the role of emotions.

Substantial, though gradual, advances have been made in this type of system, and some prototypes have already been embodied in robots, with motion and visual control systems, sound, touch and pain receptors[3]. A naval dispatching system[4] was also designed. Based on correlates to consciousness principles it incorporated natural language processing, database interaction, and resource management.

A common limitation of different architectures for conscious agents is the large number of partially independent modules for performing complex tasks involving multi-modal learning. Therefore, we propose a multi-agent task planning system whose architecture is inspired by a cognitive system with conscious-like emotion-driven behavior using only weightless artificial neural networks (WANN): WiSEMAN (WiSARD Emotional Multi-Agent Network). WANNs are especially computationally inexpensive. While the WiSEMAN itself is still theoretical and not yet tested, its different modules have been developed separately and tested in different domains.

Section 02 describes the WiSARD, WANN model used here, and Section 03 details the WiSEMAN architecture and refers to the basis on the technology needed to build its modules.

2. WISARD

WANNs are RAM-based neural networks where each neuron is a simple memory table. Learning on these models consists on memory writes, and classification on memory reads. A traditional and simple model of WANN is WiSARD[5], a class discriminator-oriented architecture that has an input retina, responsible for performing a pseudo-random mapping of n-tuples of a binary input into specific RAM locations. Since N neurons compose each WiSARD discriminator, the length of a binary input is roughly $N * n$ bits. Each memory location stores an integer that represents the number of hits in the memory position addressed by the n bits in question.

The WiSARD training phase consists of feeding an input data to the model and increment the counter inside each accessed RAM location of a class discriminator. To classify a sample, all discriminators access their RAM locations and return the counters that represent the scores of each class. The discriminator with the highest score is the predicted class of the model. If a score tie occurs, a threshold technique called *bleaching* is applied. The *bleaching* value is initialized to zero and

Appendix B

Extended experiments results

B.1 Qualitative experiments

B.1.1 Positives

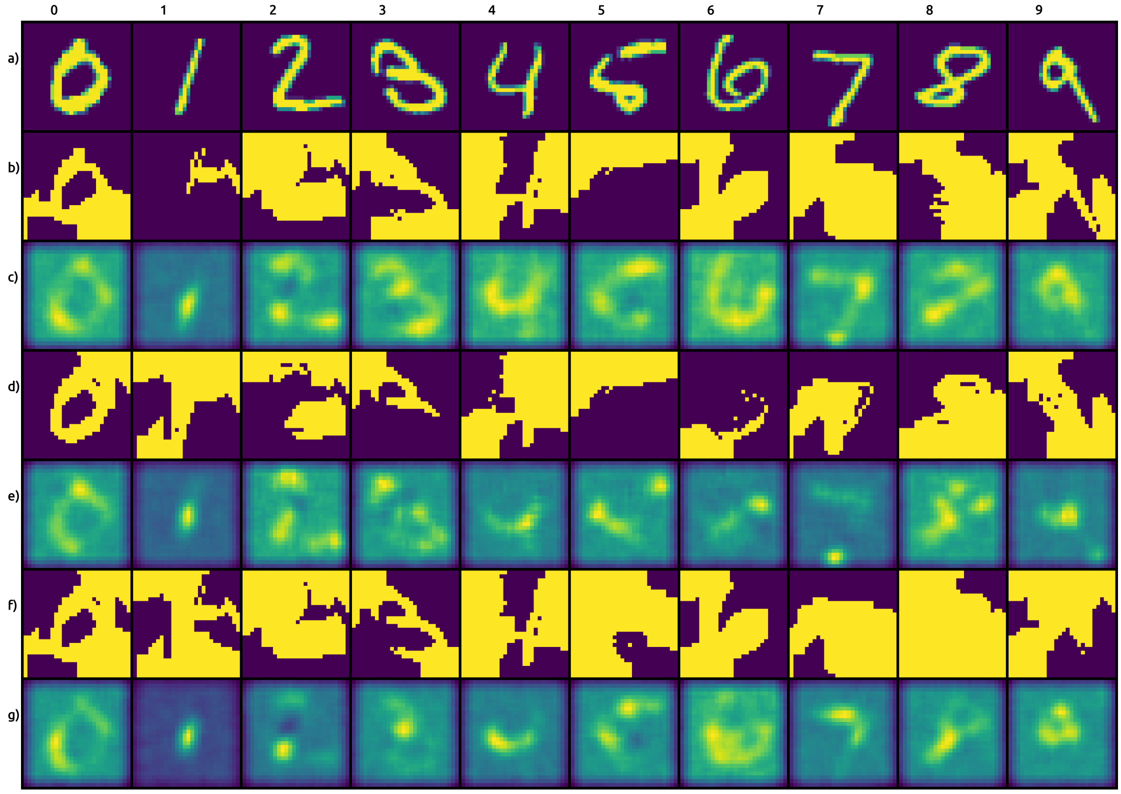


Figure B.1: MNIST qualitative experiment: a) original; b) LIME – WiSARD; c) FRWI – WiSARD; d) LIME – Ridge Regression; e) FRWI – Ridge Regression; f) LIME – Random Forest; g) FRWI – Random Forest

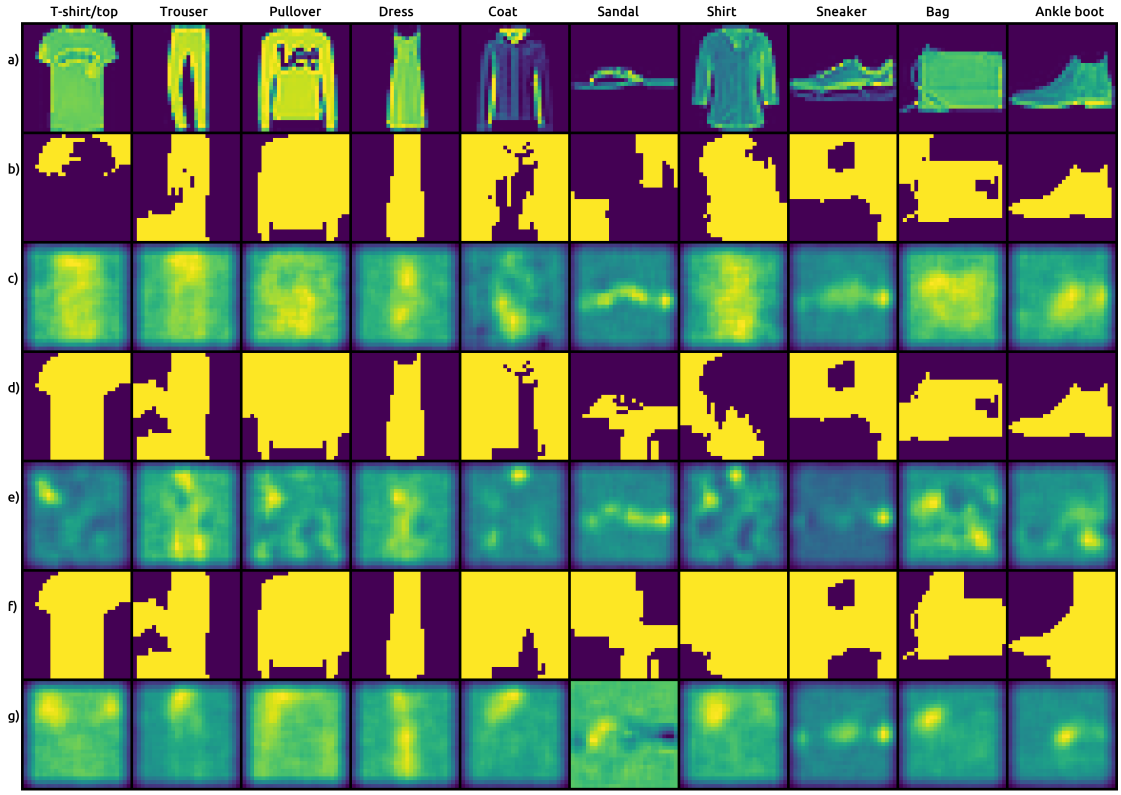


Figure B.2: Fashion MNIST qualitative experiment: a) original; b) LIME – WiSARD; c) FRWI – WiSARD; d) LIME – Ridge Regression; e) FRWI – Ridge Regression; f) LIME – Random Forest; g) FRWI – Random Forest

B.1.2 Negatives

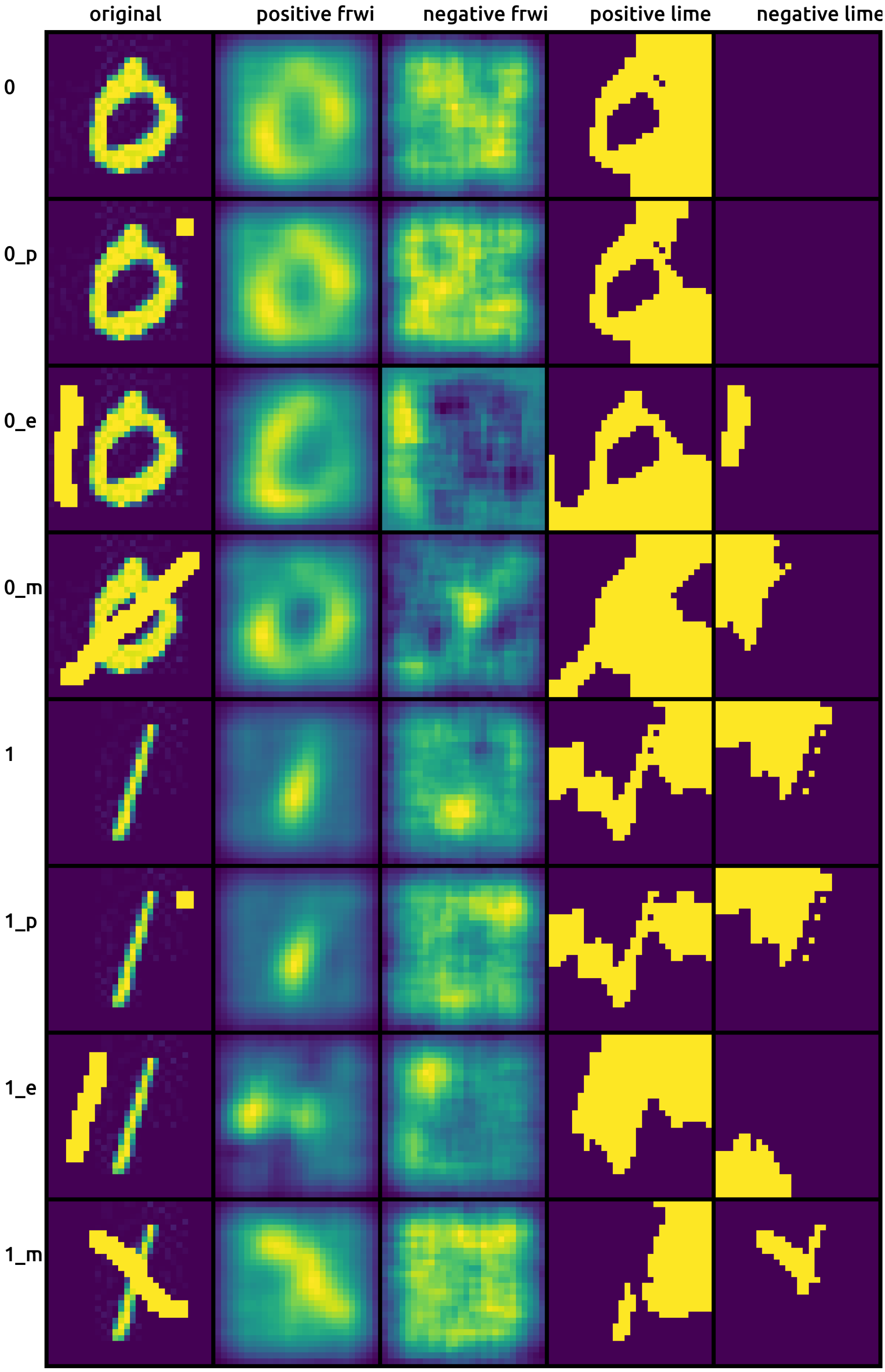


Figure B.3: MNIST negative qualitative experiment: 0 and 1

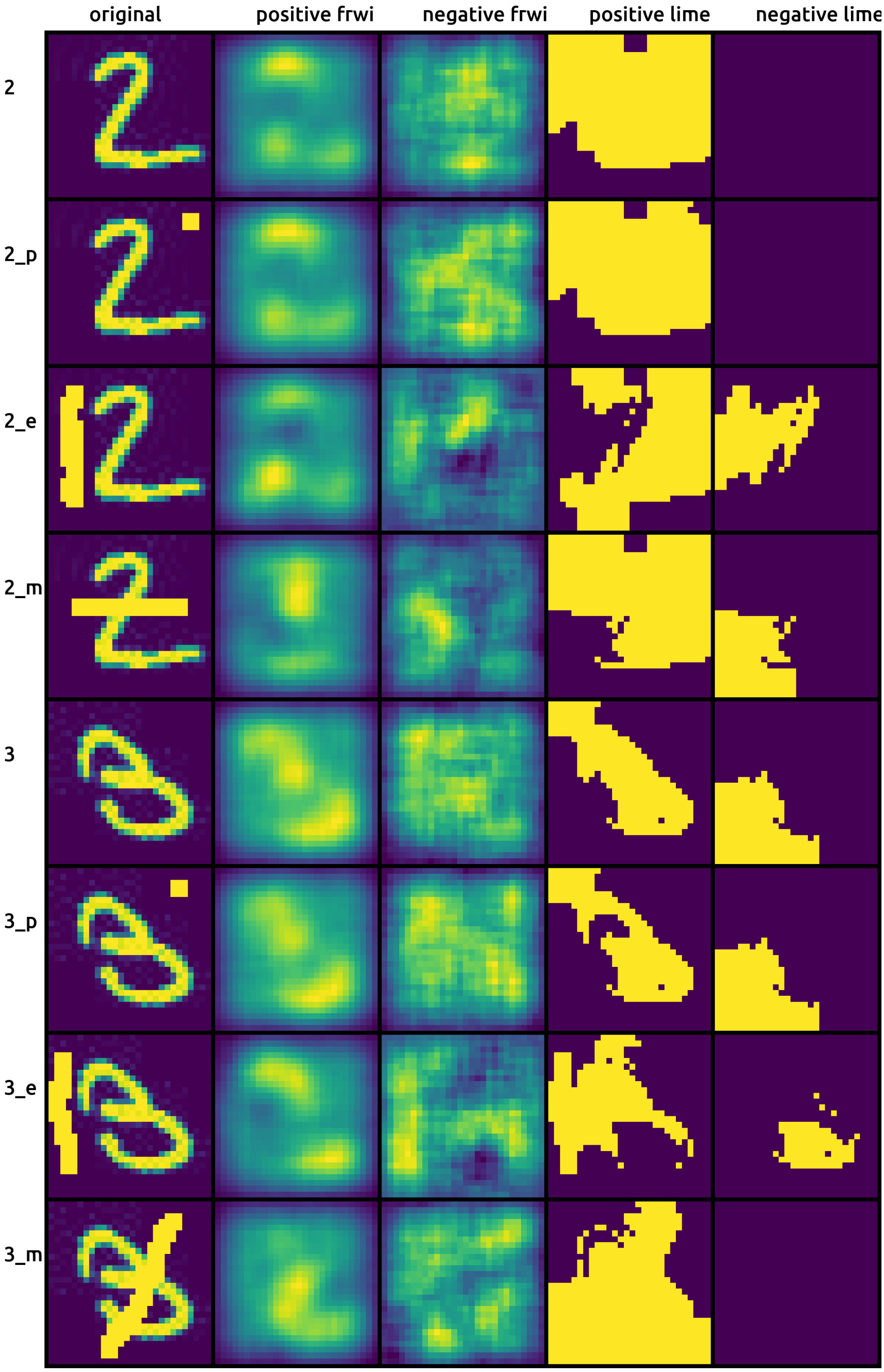


Figure B.4: MNIST negative qualitative experiment: 2 and 3

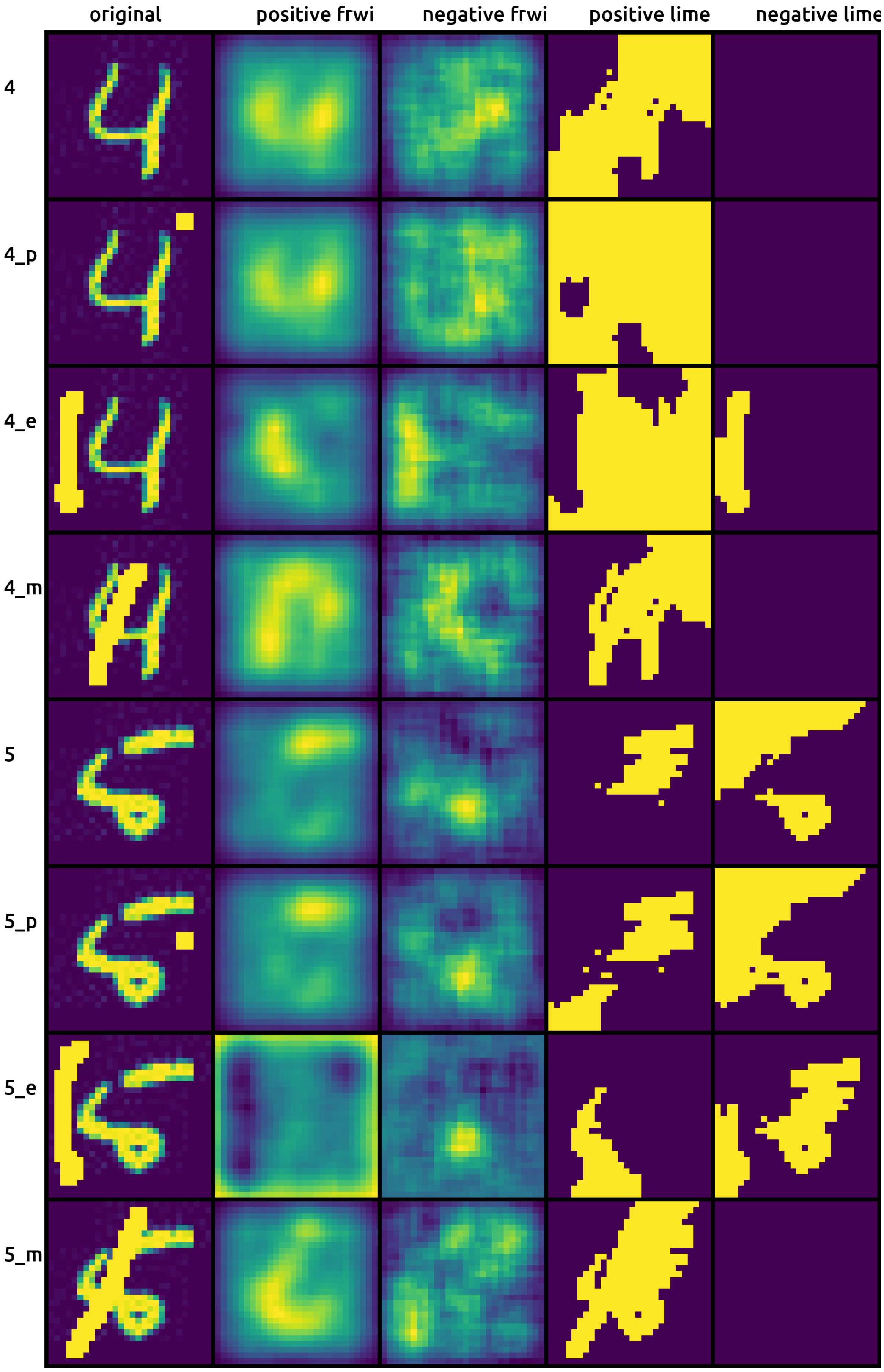


Figure B.5: MNIST negative qualitative experiment: 4 and 5

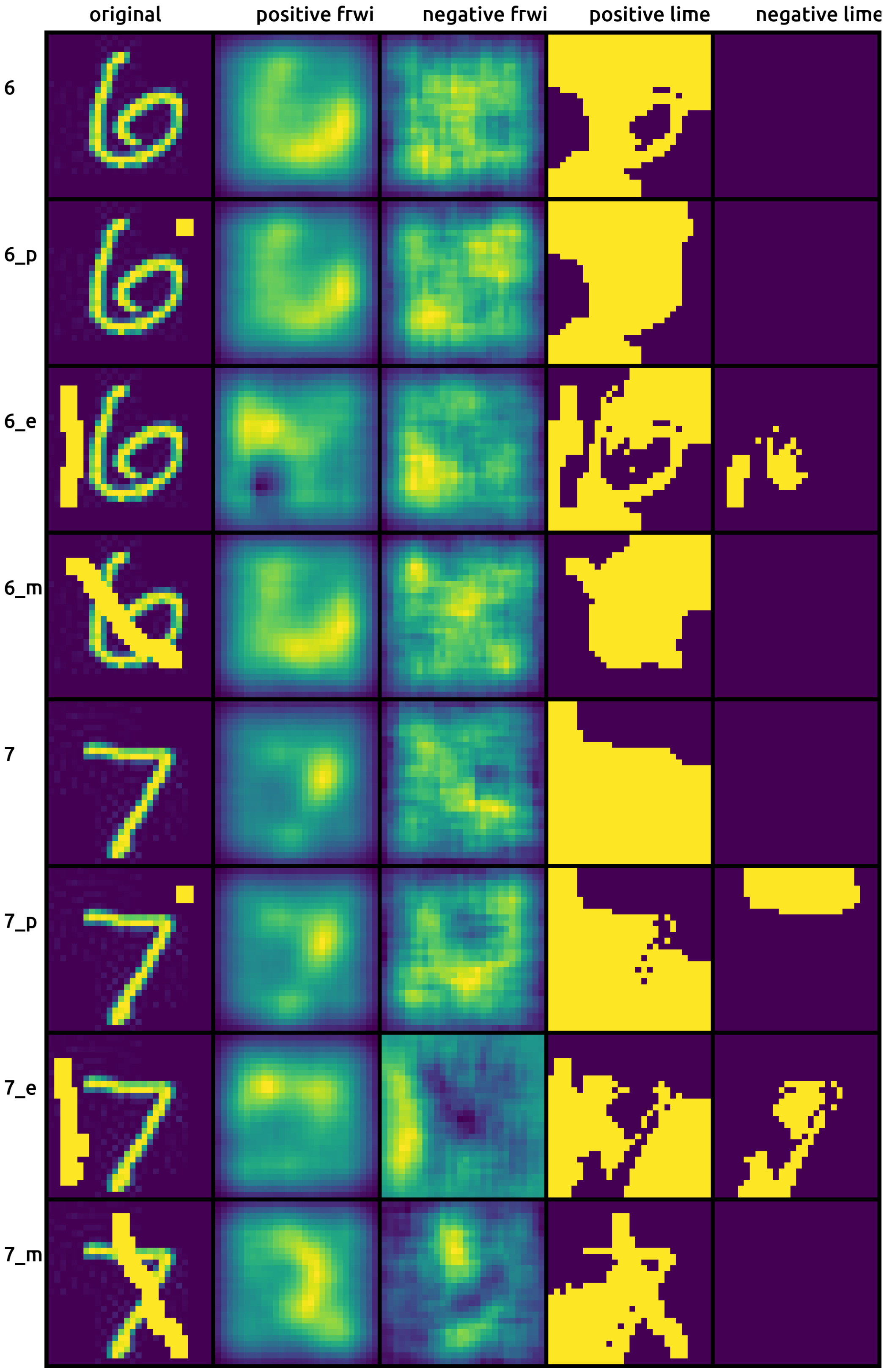


Figure B.6: MNIST negative qualitative experiment: 6 and 7

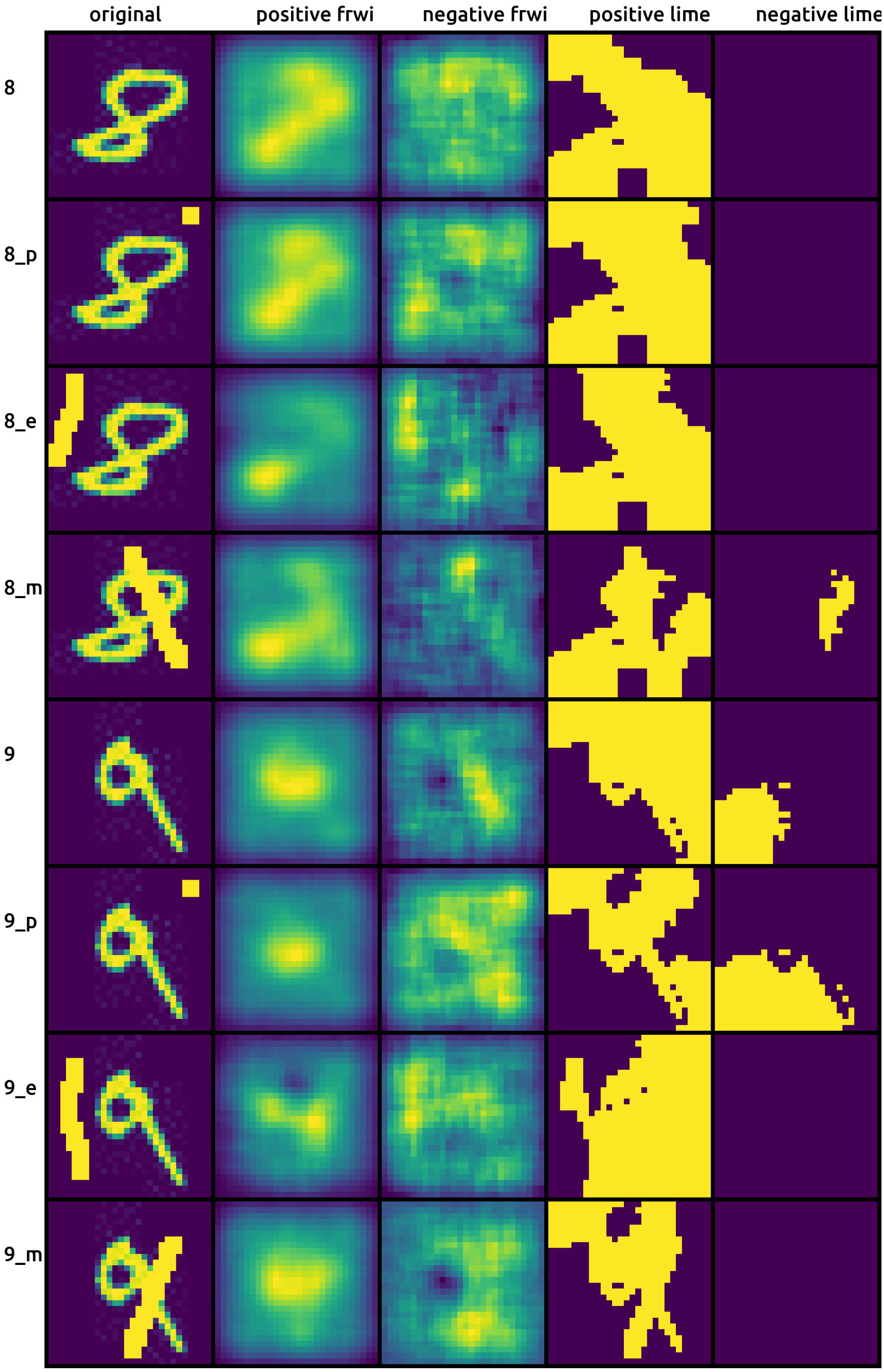


Figure B.7: MNIST negative qualitative experiment: 8 and 9

B.2 Parameters variation experiments

In this section are presented some of the experiments made varying parameters of the proposed explainable model, aiming to produce an overview of the corresponding behaviour changes. First, the parameter feature size was tested in the following values: 1; 2; 4; 6; 8; 10, which correspond to the size of the feature drawn by the model to determine relevant regions. Second, the parameter number of examples was tested in the following values: 1000; 5000; 10000; 50000; 100000, which mainly serves to induce a convergence of the explanations with more examples determining the same relevant regions.

In the images below, each line has a feature size value in pixels, and each column has a number of examples. These experiments were ran only in the MNIST data set to train the learning models, with WiSARD, Ridge regression and Random forest as the section 5.1.3 details, and only with the first image of the data set of test, which is the number seven as can be seen in the figure 5.1, to extract the explanation with the proposed explainable model.

This shows how increasing the number of examples causes the resulting image to grow smoother, as the increasing iterations cause a convergence on relevant regions. As for the second parameter, as the feature size grows, information is shared with neighbouring pixels leading a less sharp image, but with very highlighted or concentrated relevant regions.

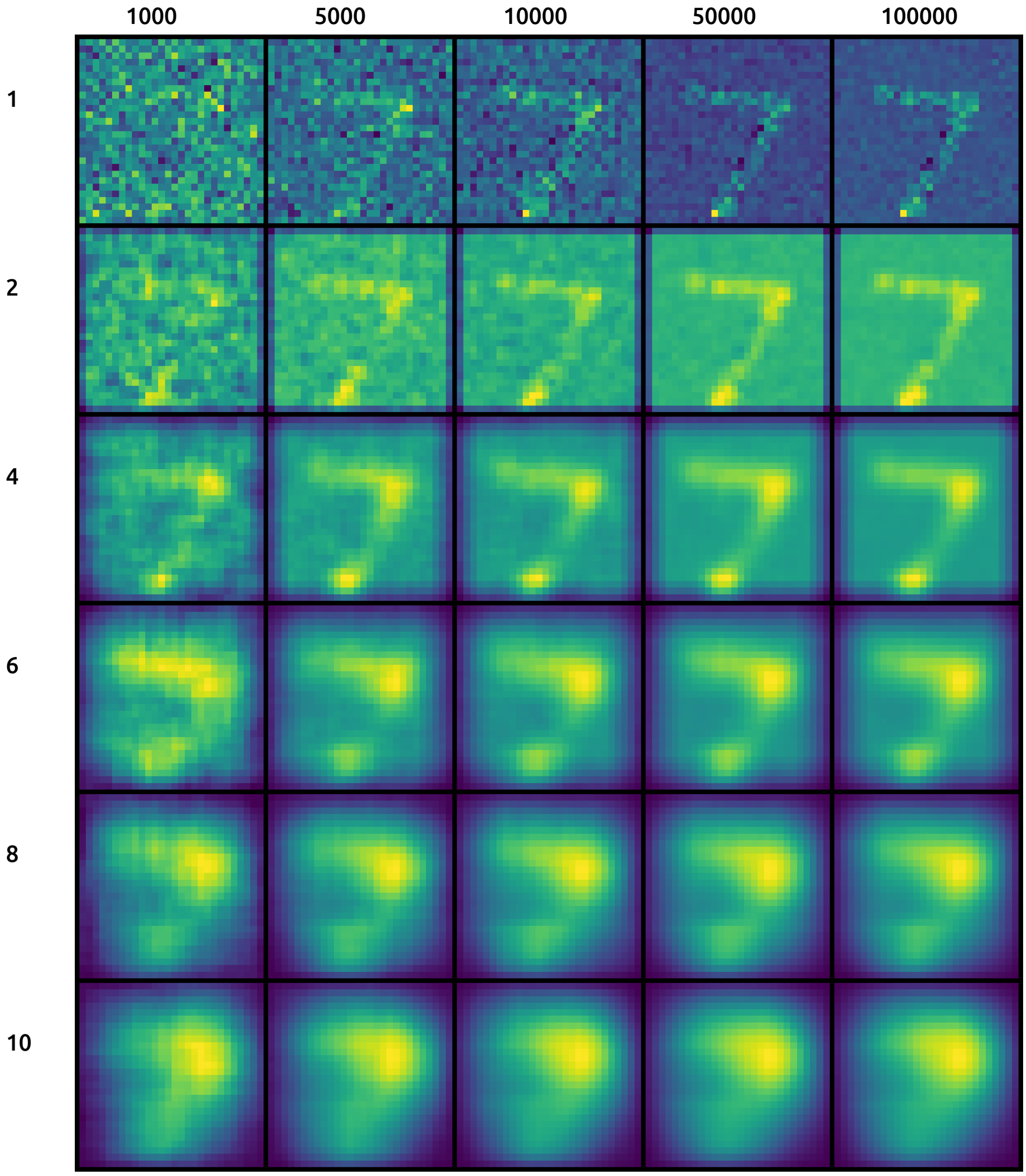


Figure B.8: FRWI parameters variation experiment on MNIST and WiSARD

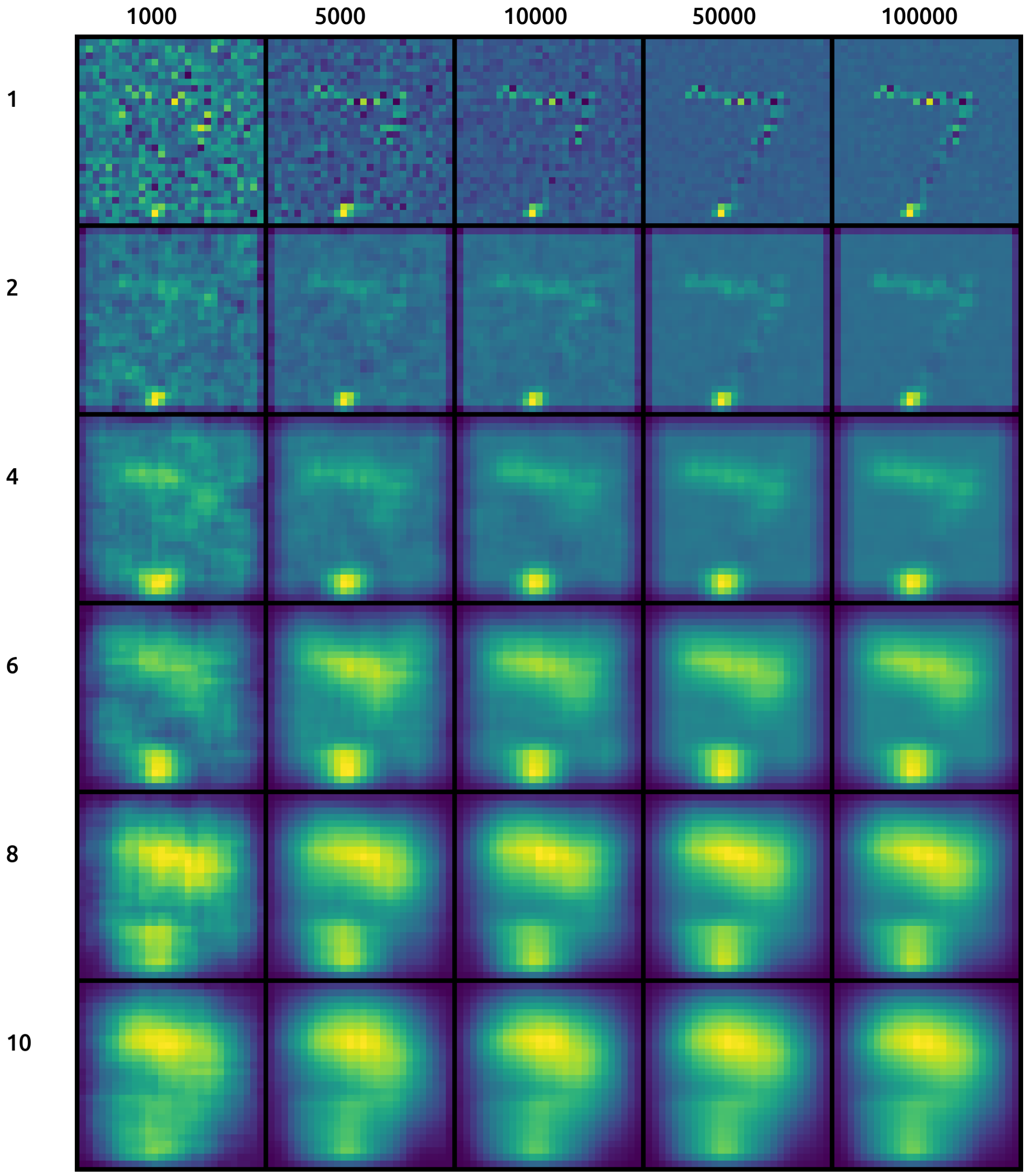


Figure B.9: FRWI parameters variation experiment on MNIST and Ridge Regression

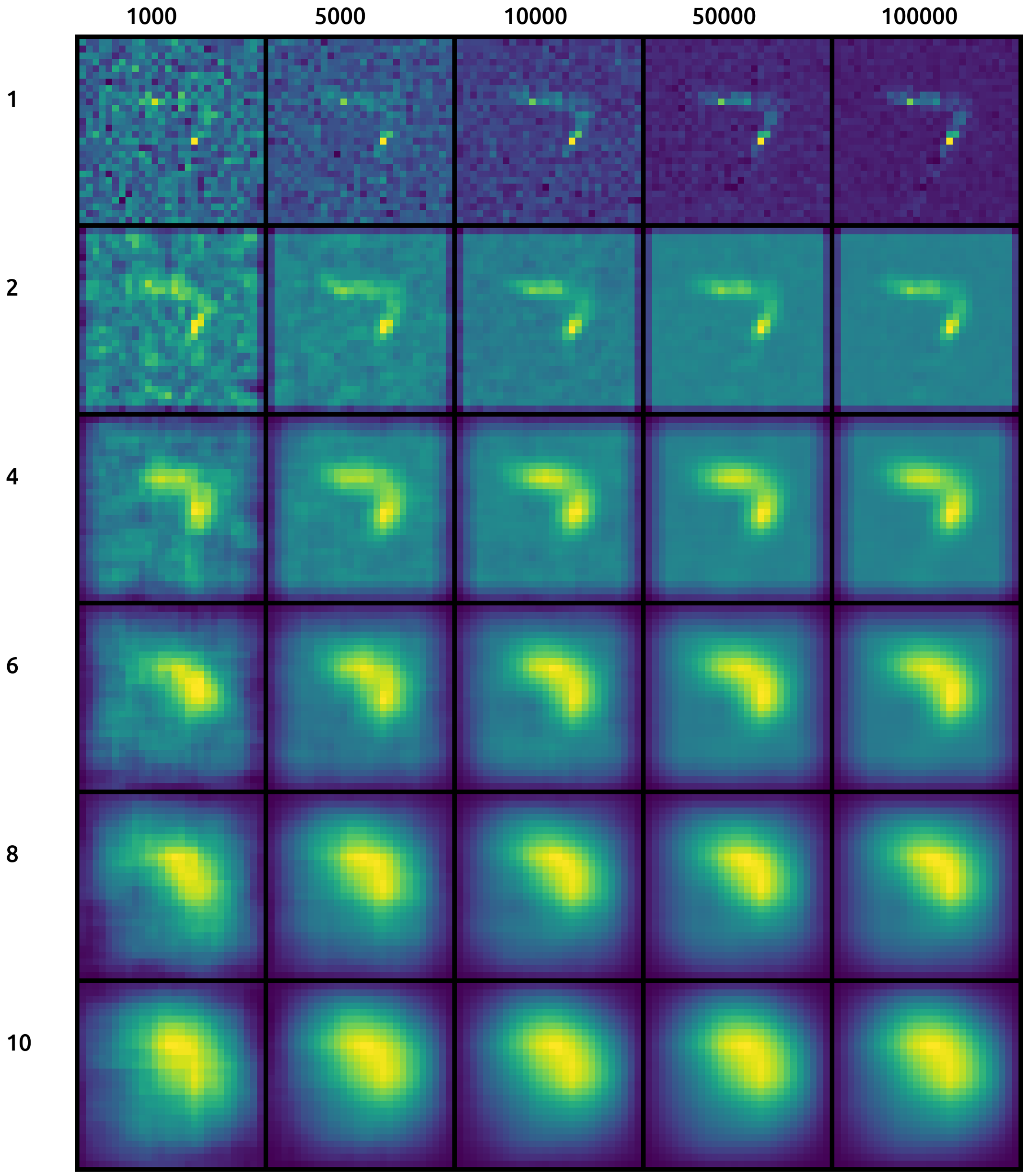


Figure B.10: FRWI parameters variation experiment on MNIST and Random Forest

B.3 Quantitatives experiments

B.3.1 MNIST 1k

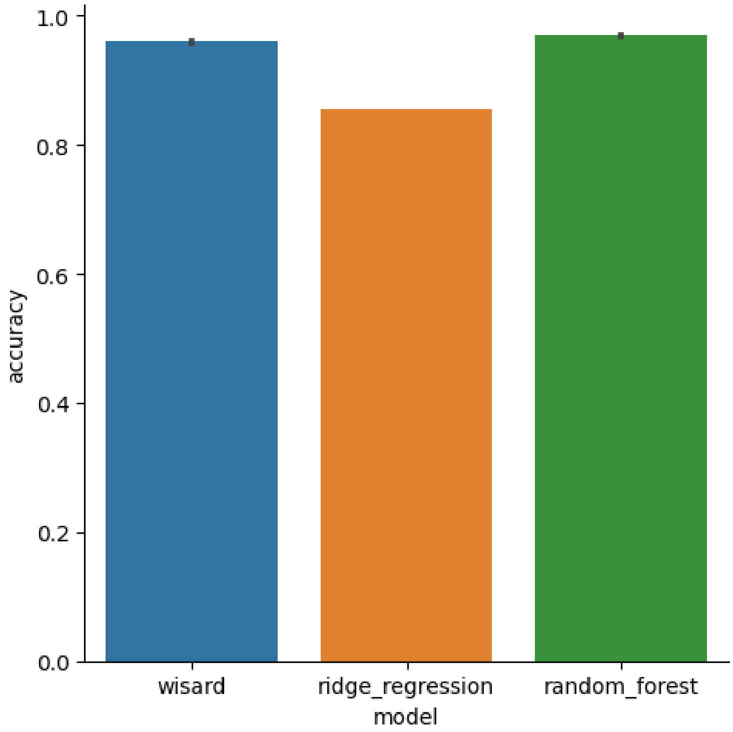


Figure B.11: MNIST models accuracy

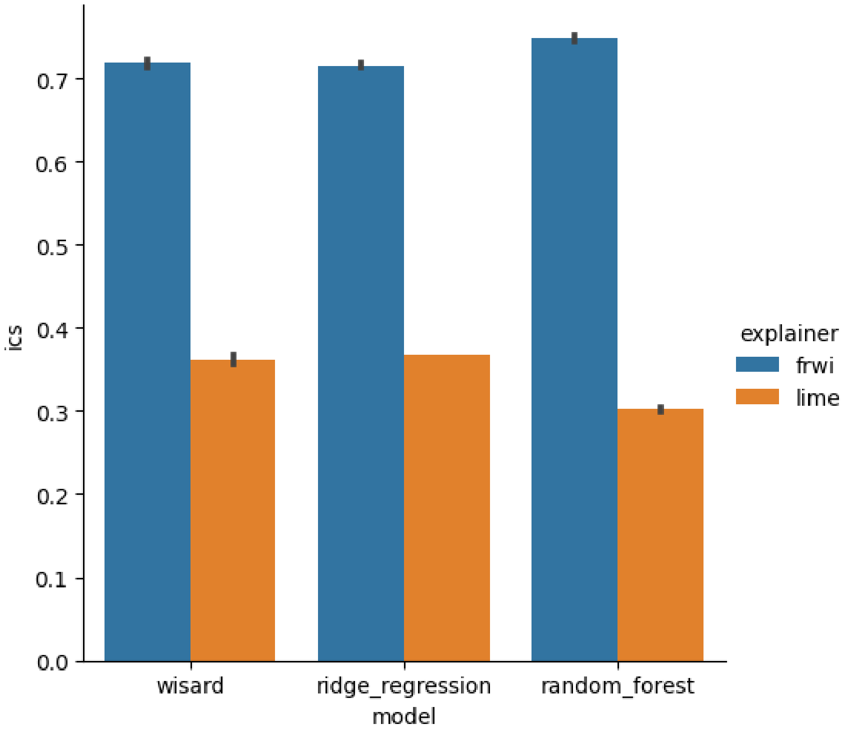


Figure B.12: MNIST models ICS

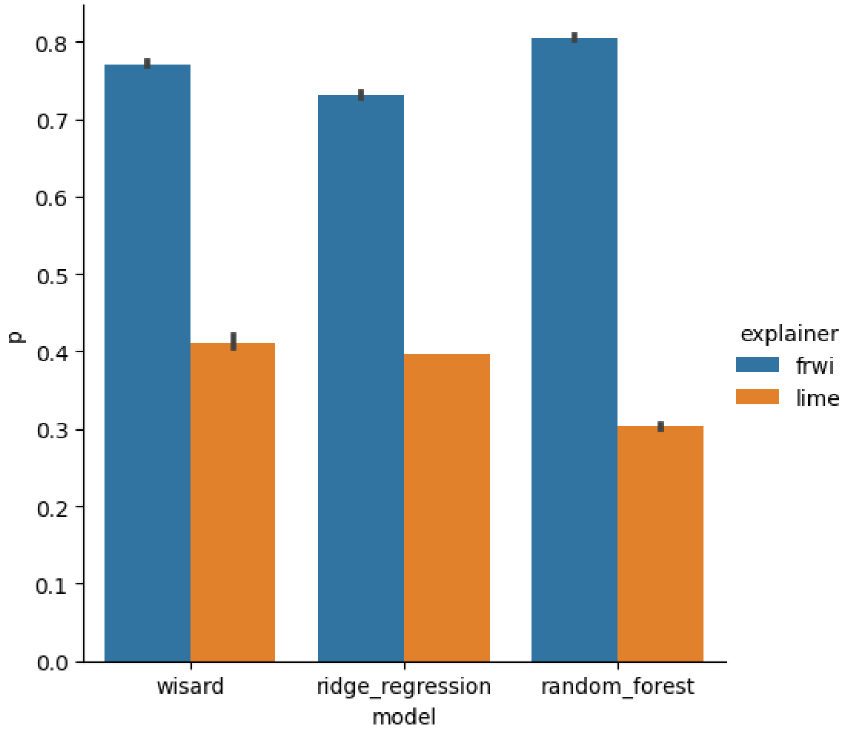


Figure B.13: MNIST models; mean from p variable

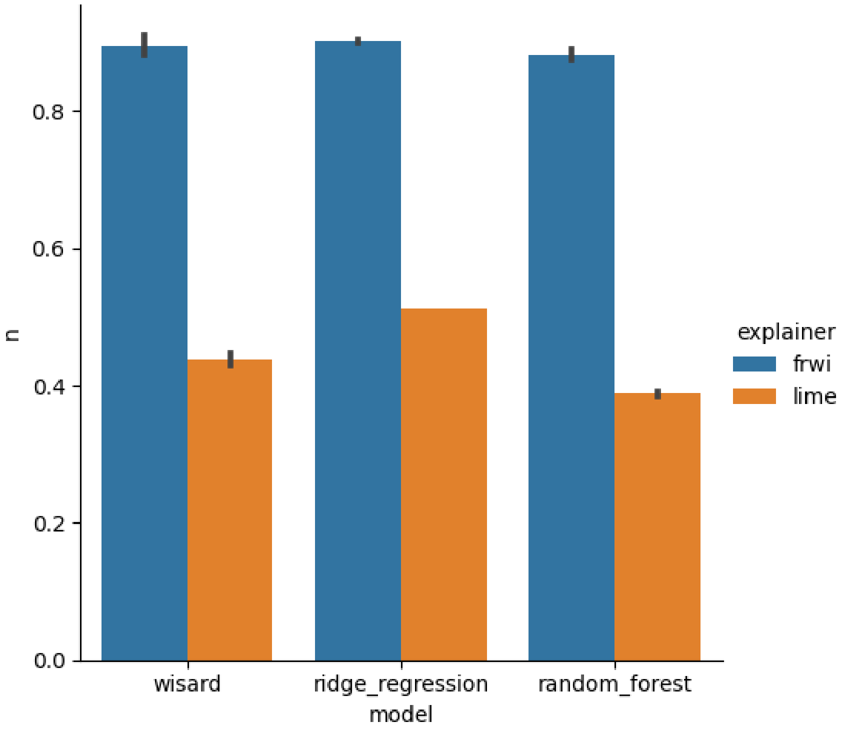


Figure B.14: MNIST models; mean from n variable

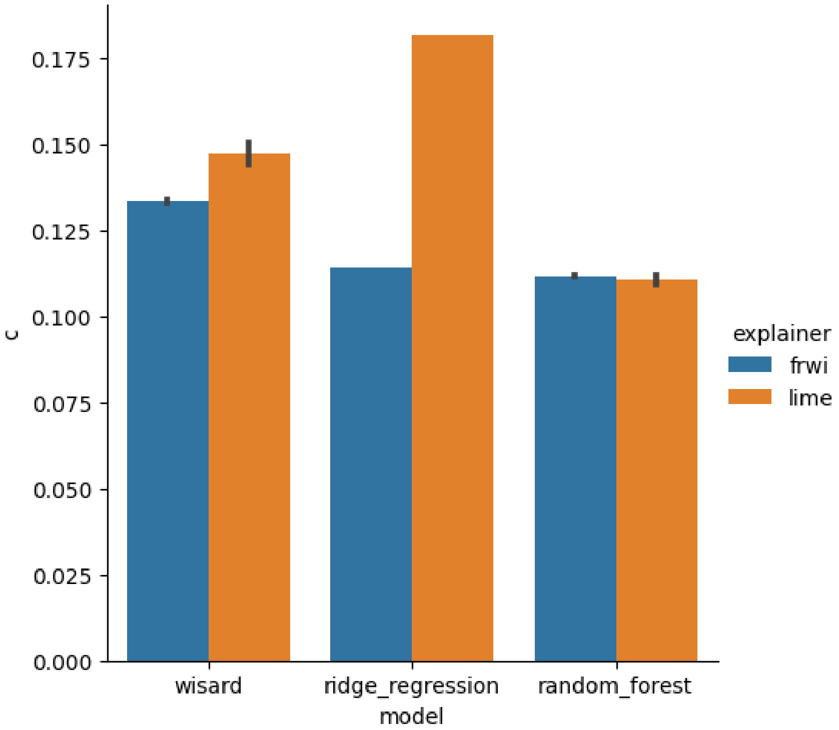


Figure B.15: MNIST models; mean from c variable

B.3.2 MNIST 10k

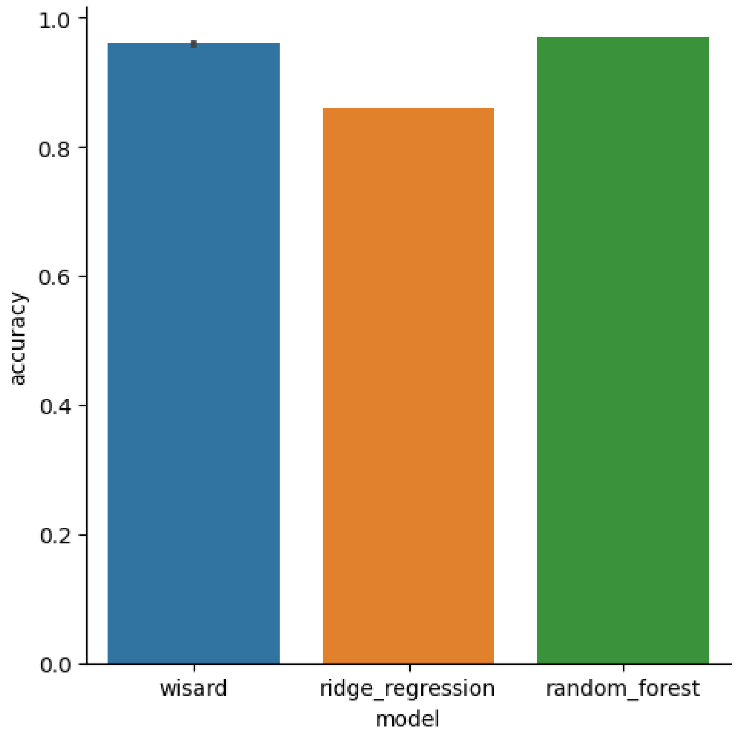


Figure B.16: MNIST models accuracy

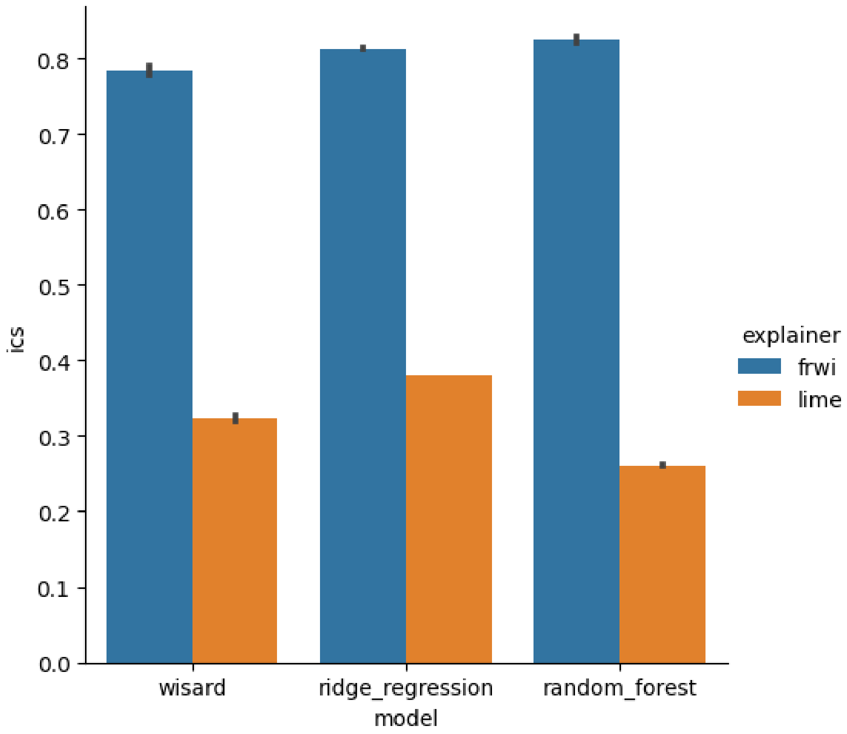


Figure B.17: MNIST models ICS

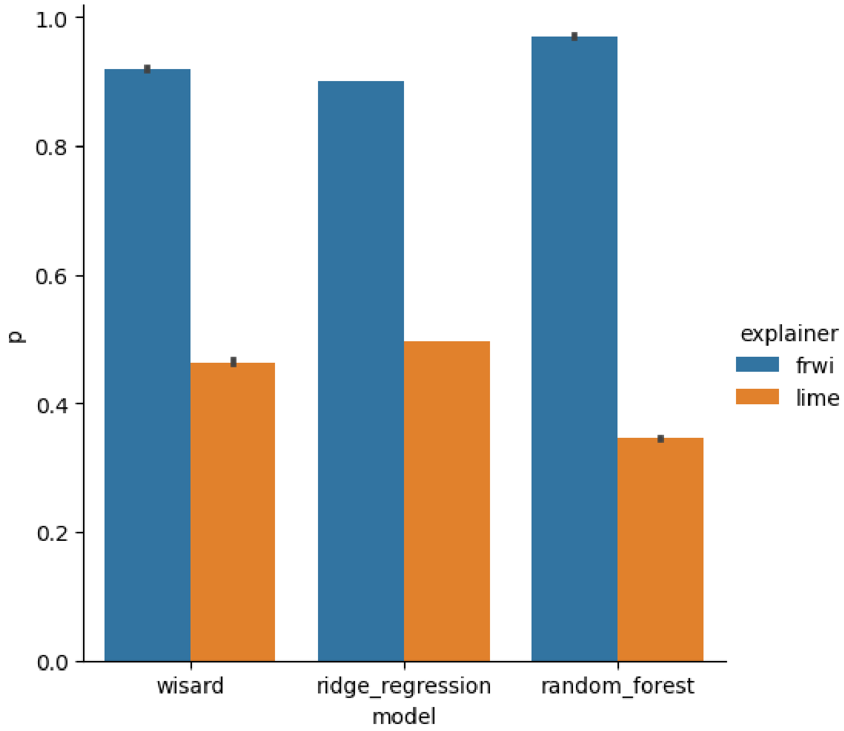


Figure B.18: MNIST models; mean from p variable

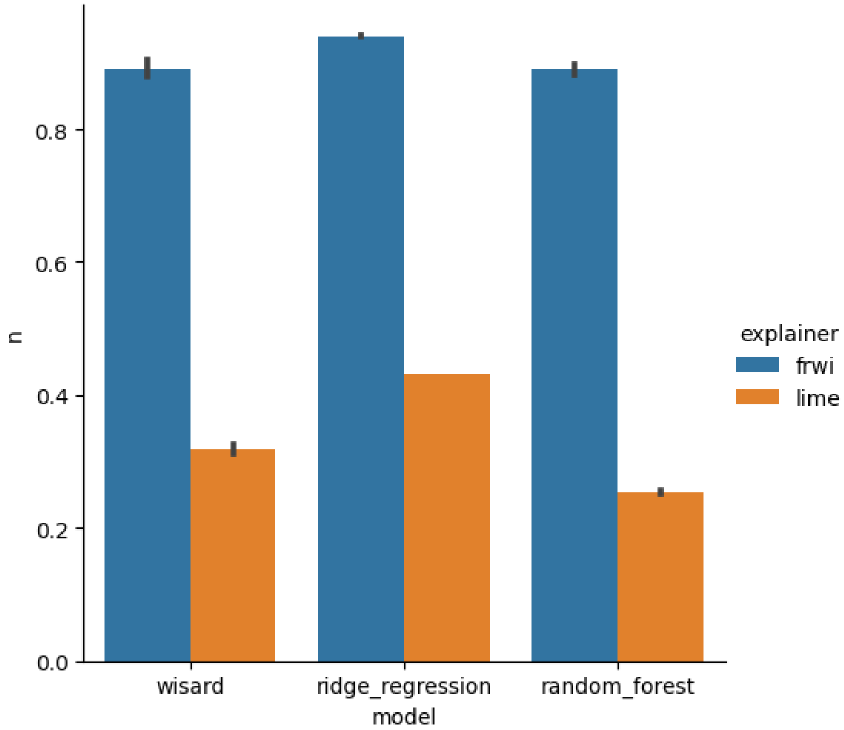


Figure B.19: MNIST models; mean from n variable

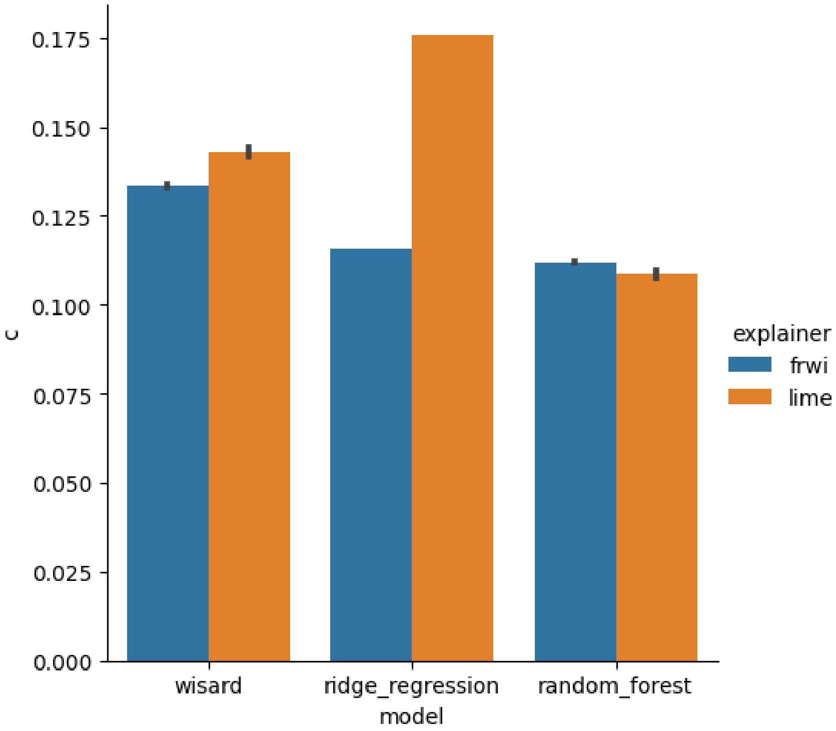


Figure B.20: MNIST models; mean from c variable

B.3.3 Fashion MNIST 1k

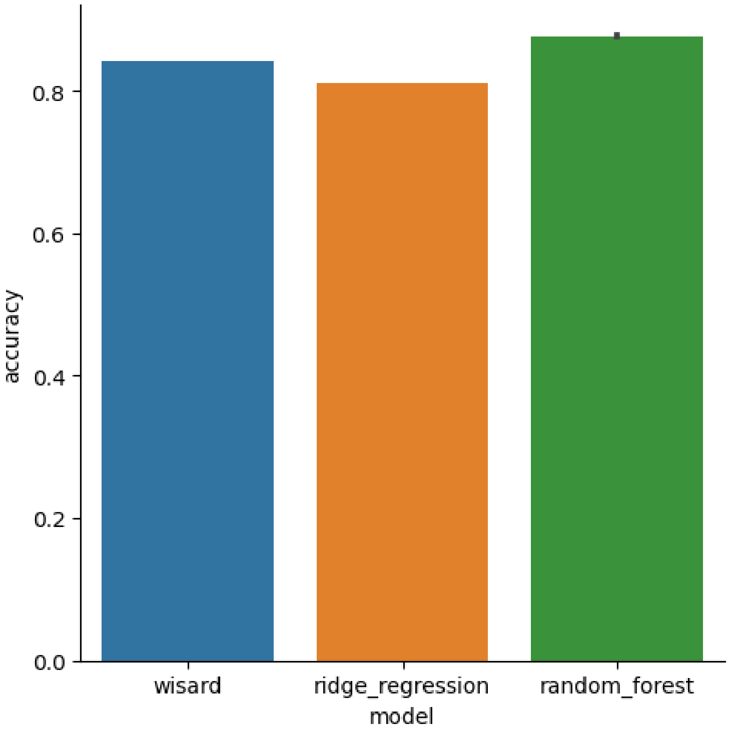


Figure B.21: Fashion MNIST models accuracy

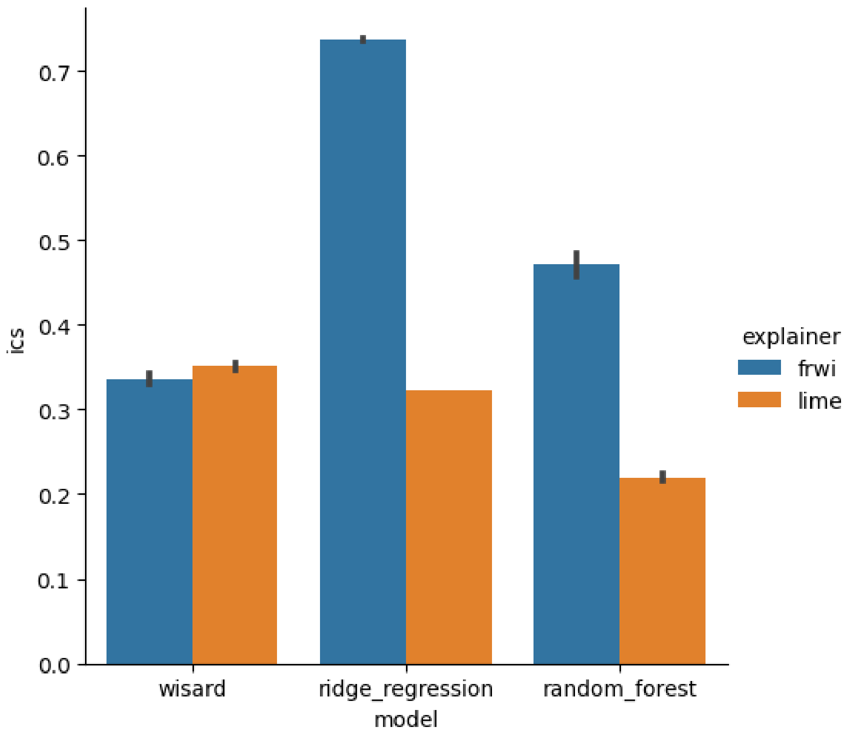


Figure B.22: Fashion MNIST models ICS

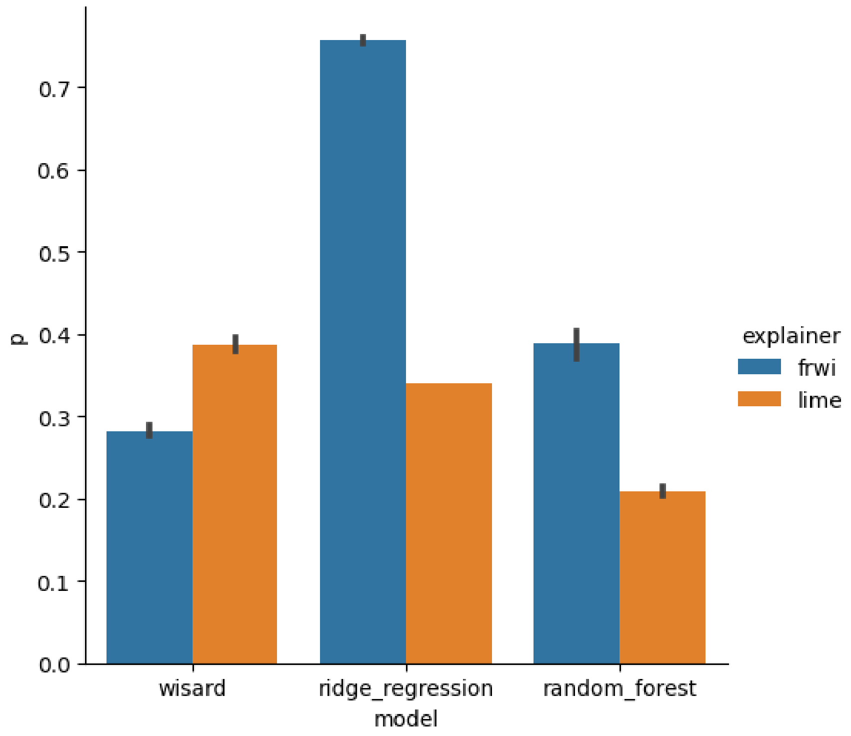


Figure B.23: Fashion MNIST models; mean from p variable

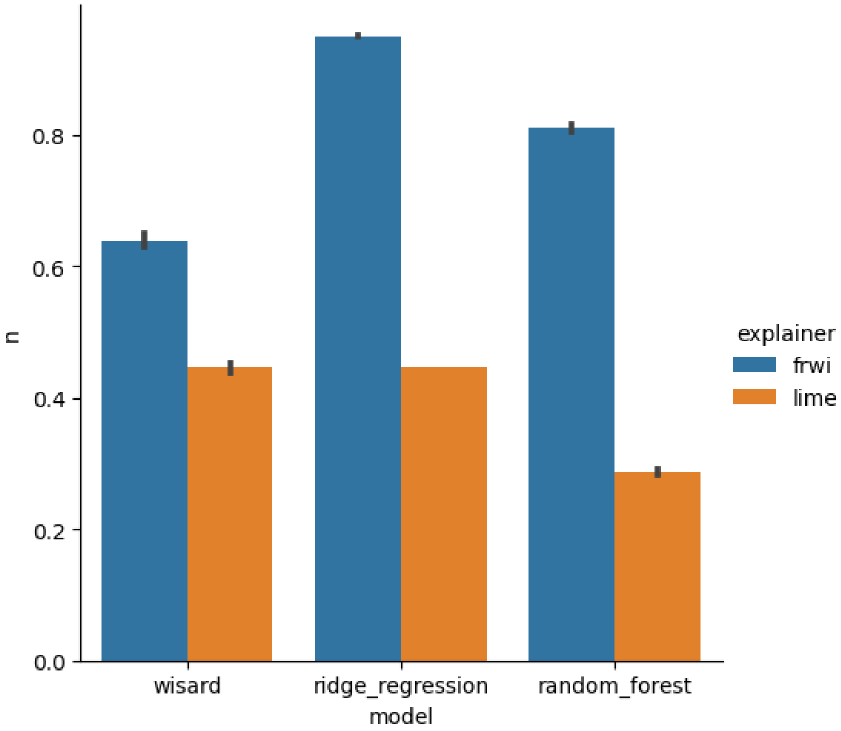


Figure B.24: Fashion MNIST models; mean from n variable

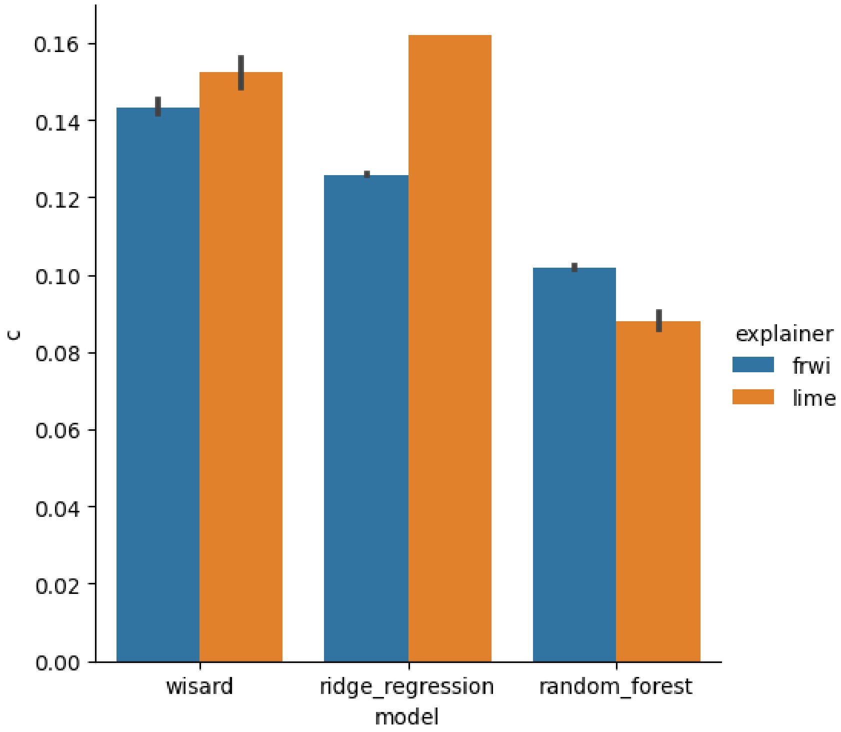


Figure B.25: Fashion MNIST models; mean from c variable

B.3.4 Fashion MNIST 10k

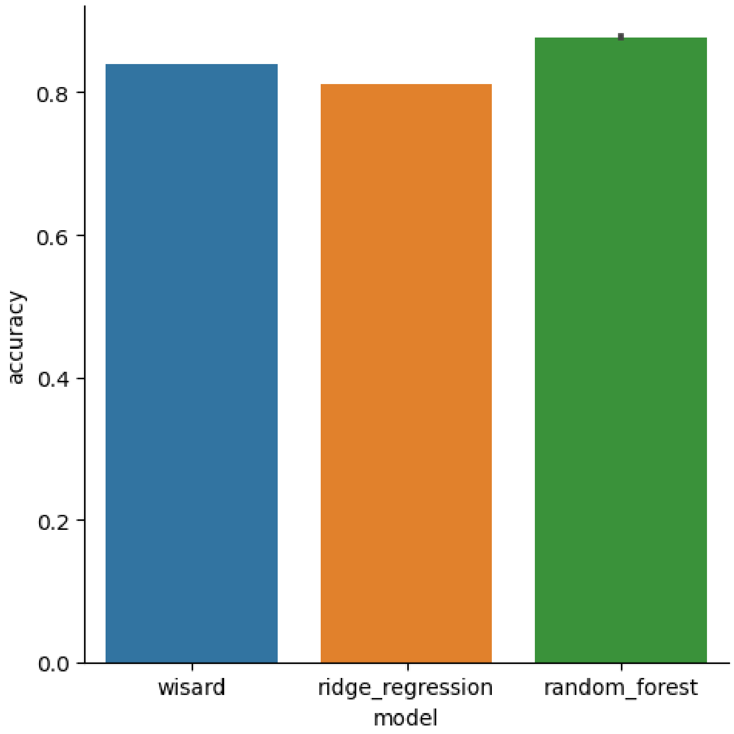


Figure B.26: Fashion MNIST models accuracy

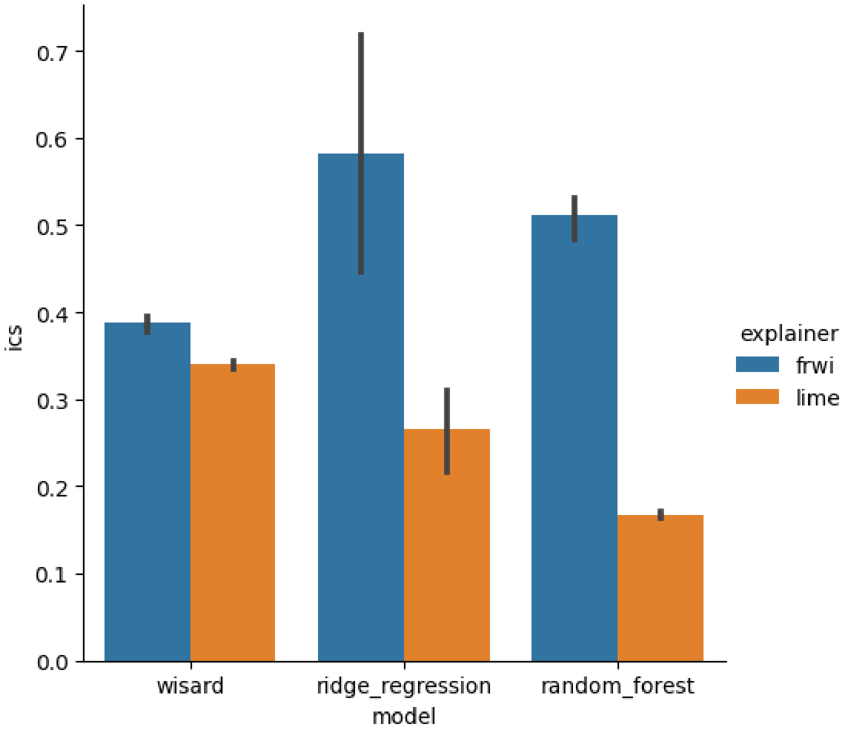


Figure B.27: Fashion MNIST models ICS

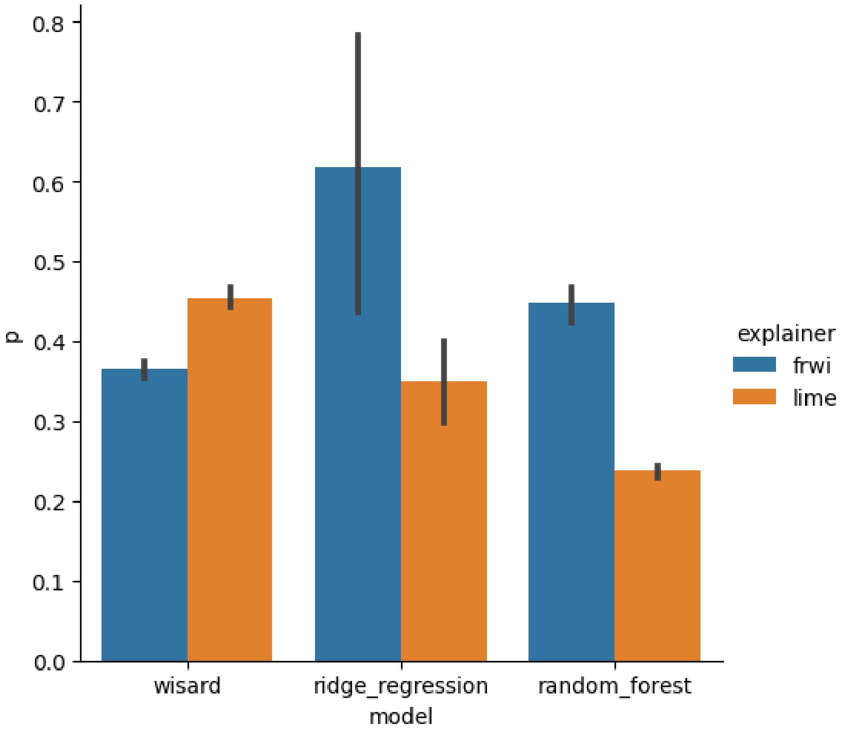


Figure B.28: Fashion MNIST models; mean from p variable

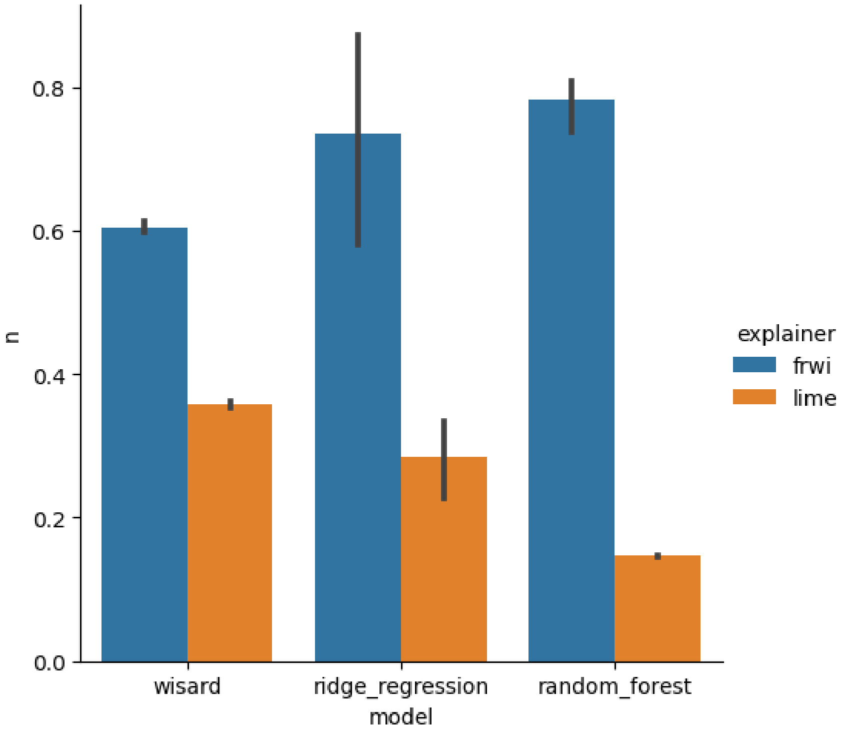


Figure B.29: Fashion MNIST models; mean from n variable

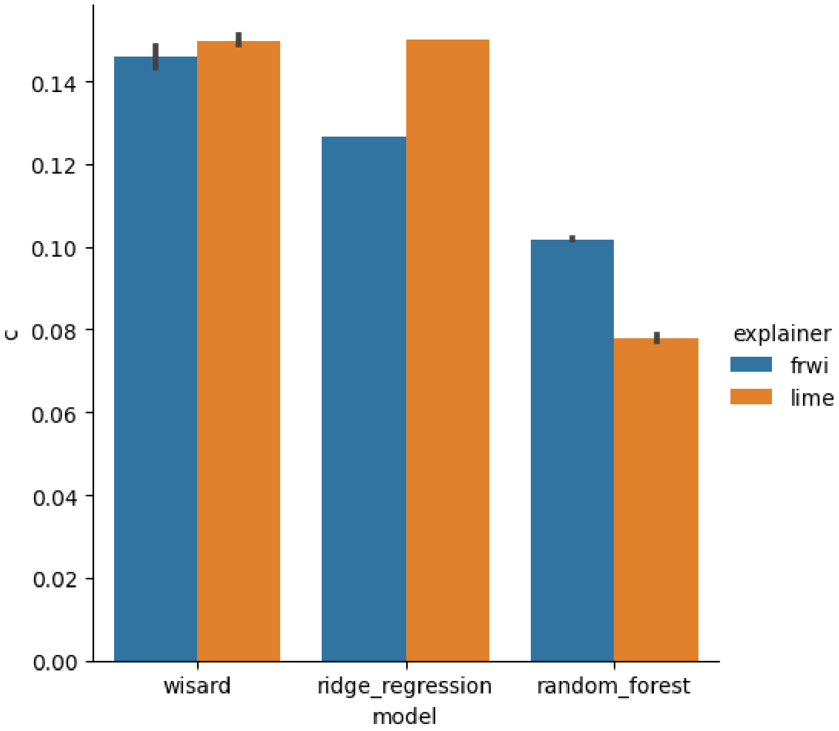


Figure B.30: Fashion MNIST models; mean from c variable