Sensitivity Analysis for Deep Learning Models Interpretability in Epistasis Detection

Bernardo Ferreira Mendes Cotovio de Bastos
Instituto Superior Técnico
Lisbon, Portugal
bernardo.bastos@tecnico.ulisboa.pt

***Abstract*—Recent discoveries made by genome-wide association studies (GWAS) have been crucial to understanding the association between genes and diseases. Until today, thousands of SNPs have been associated with diseases and contributed to a better understating of disease genetics. Interactions between genes are often referred to as Epistasis, and their detection consists of one of the biggest statistical challenges in genetic epidemiology. Epistasis detection has been revealed to be a complex phenomenon that can not be solved by traditional statistical methods. In recent years, due to their ability to non exhaustively extract information from the data, the emerging field of deep learning has been applied in genomic prediction. However, the black-box nature of deep learning networks remains one of the biggest drawbacks of these approaches. In this dissertation, a new framework to interpret the information extracted from deep learning algorithms is presented and tested under different epistatic scenarios. A relevance score is assigned to each SNP using sensitivity analysis. From the results on datasets with and without marginal effects, an accuracy threshold from which networks can be interpreted is established. For MLPs and CNNs with accuracy over 0.5482 and 0.5478, their results can be trusted and interpreted. To conclude, the findings are tested on a real Breast Cancer dataset and compared with a recent study that performed an exhaustive analysis on the same dataset. The results identify SNPs "rs2010204", "rs1007590", "rs660049", "rs0504248" and "rs500760", which belong to interactions of order two three and four, among the Top 30% most relevant SNPs.**

***Index Terms*—epistasis detection, genome-wide association study, deep learning, model interpretability, sensitivity analysis, higher-order interactions**

I. Introduction

Genes are the basic unit of heredity and act as a guide to synthesize proteins. A gene is a sequence of nucleotide pairs that together code a protein, still, many genes do not code any protein, often referred to as non-coding DNA. The total number of genes in an organism is known as the genome. The genome is coded into several long sequences of DNA named Chromosomes. Locus, plural Loci, is the specific region of a chromosome where a particular gene is encoded.

Genes can suffer small changes (mutations) in their sequence of nucleotides, leading to different versions of the synthesised protein. These different variants of a gene are known as alleles. Alleles influence the expressed phenotypes and are the reason why every individual is unique. When creating a new individual, each parent provides a copy of an allele to their descendent. Therefore, for each locus, every individual has two alleles provided by each parent.

At an inter-loci level, alleles interact with each other meaning that the expression of an allele might be oppressed by the other corresponding allele. This concept is referred to as allele dominance, with the allele being expressed named as dominant and the allele being oppressed named as recessive. Considering a dominant allele A and a recessive allele a, each gene there can have three possible genotypes: homozygous major allele (AA, both parents provide a dominant allele), heterozygous allele (Aa or aA, each parent provides a different type of allele) and homozygous minor allele (aa, both parents provide a recessive allele).

SNPs are the most common genetic variant in the human genome. In [1], a study describing common human genetic variants, identified over 88 million variants, out of which 84.7 million were classified as single nucleotide polymorphisms.

An SNP is a substitution of a single nucleotide in a certain stretch of the DNA. For example, the nucleotide thymine can be replaced by the nucleotide cytosine in a certain location of the genome. Most of these genetic differences do not affect the health or development of an individual, however, some SNPs have proven to be an important genetic marker in predicting the human response to certain drugs or the susceptibility of developing a certain type of disease.

The identification of genetic markers which can be associated with a disease phenotype has become an important field of research in genetic epidemiology. These studies are often referred to as GWAS and have gained a lot of popularity in recent years [2]. GWAS are case-control studies that identify SNPs that influence a particular phenotype. Each SNP is evaluated individually regarding their association with the phenotype. However, complex diseases are often caused by multiple interacting SNPs, each with a small effect per SNP. Thus, when analysing complex diseases, interactions between SNPs must also be considered. These interactions between SNPs are denoted as epistasis and detecting them has become a significant area of research in human genetics [3].

When considering epistatic interactions the concept such as marginal effects needs to be clarified. An SNP is said to have marginal effects if it directly interacts with the phenotype. Hence, SNP interactions displaying marginal effects (ME) are interactions whose SNPs directly interact with the phenotype. However, there are some cases where each individual SNP has no effect on the phenotype but their combination has a strong effect. These combinations are SNP interactions displaying no

marginal effects (NME).

A wide variety of methods, from traditional statistical methods to more complex machine learning and artificial intelligence methods, have been proposed to detect gene-gene interactions. The epistasis detection problem has revealed to be a complex problem with a heavy computational burden associated that current computers cannot handle efficiently [3].

II. Background: Epistasis

To understand how statistical methods can be applied for epistasis detection, it is essential to clarify how the biological concepts of epistasis are mapped into computers.

Mathematically, SNP sequences are represented as two numerical matrices (Figure 1): one representing the SNP data and the other the labelling data [4]. Individuals are represented as samples. For the SNP data, a row represents the genotypes of a sample and each column an SNP. Genotypes are coded as 0, 1, or 2, corresponding to homozygous major allele, heterozygous allele or homozygous minor allele, respectively. The label matrix is a column listing the binary phenotypes of each sample, where samples having the phenotype are classified as 1 (cases), and samples not having that phenotype are classified as 0 (controls).

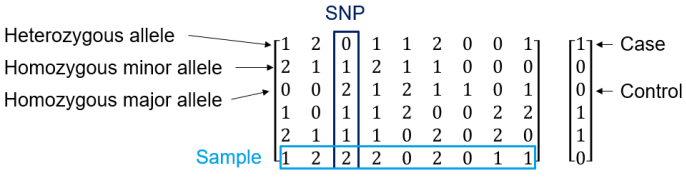

Fig. 1. Mathematical matrices used to represent genomic data.

Also, it is necessary to understand the concepts of Minor Allele Frequency (MAF) and heritability (h^2). MAF represents the frequency at which the recessive allele occurs in a population. Heritability is the proportion of observable differences between individuals caused by genetic differences. It quantifies how much the variation of a trait can be assigned to genetic factors [5]. According to [6], heritability can be obtained by:

$$h^2 = \frac{\sum_i (P(D|g_i) - P(D))^2 P(g_i)}{P(D)(1 - p(D)))} \tag{1}$$

$$P(D) = \sum_i P(D|g_i) P(g_i) \tag{2}$$

where $P(D|g_i)$ is the probability of expressing the phenotype having the genotype g_i, $P(g_i)$ is the probability of having genotype g_i and $P(D)$ is the probability of expressing a phenotype in a population, also called disease prevalence (equation 2).

When creating datasets, generators like GAMETES [5] and Toxo [6] use the heritability and MAF parameters to express epistatic relations between genes.

In response to the challenge of finding gene interactions related to a phenotype, exhaustive search methods started to emerge. These methods analyze all possible combinations of SNPs to determine the most accurate solution, thus avoiding the final result being a sub-optimal solution. Despite being able to detect epistasis, these approaches are only efficient in handling two, at maximum three order SNP interactions. Since each SNP can have three genotype configurations (0, 1 or 2), the number of possible genotype combinations I on an interaction of order k is given by $I = 3^k$. Thus, increasing the order of the interaction k causes the number of genotype combinations I to grow exponentially, making exhaustive search algorithms to be not computationally feasible.

To make higher-order epistasis detection computationally possible, other non-exhaustive approaches using Machine Learning (ML) and Artificial Intelligence (AI) methods were developed. These methods are more advanced than traditional statistical methods and can detect SNP interactions that are related to the phenotype. There is a wide variety of methods and only some of them were mentioned in this Thesis. Random Forest (RF), Support Vector Machines (SVM) and Ant Colony Optimization (ACO) were considered.

RF methods are promising algorithms able to capture variable importance and rank SNPs according to their relevance in predicting the phenotype. Additionally, SVM approaches proved to be able to handle high dimensional data and did not overfit. However, the ability to only detect pairwise interactions and in the presence of marginal effects makes the application of these approaches limited. ACO methods are search algorithms good for exploring and exploiting high dimensional search spaces. Their simplicity and efficiency in handling high dimensional data made them popular in recent years. Still, the absence of a good activation function to classify SNPs as interacting or not remains one of the main drawbacks of these approaches.

Several other approaches using and adapting ML and AI approached have been released. Additionally, several review papers discussing in more detail the different approaches, as well as, their advantages and disadvantages, have been published [3], [4], [7]–[12].

A. Deep Learning (DL)

From the non-exhaustive methods, approaches using DL networks for epistasis detection started to appear. These models can learn relevant internal features from high dimensional and complex datasets and have improved the state-of-the-art methods in areas related to speech recognition, image recognition, and genomics [13].

1) Multilayer Perceptron (MLP): MLP also called fully connected feed-forward networks, are one of the most popular methods in deep learning. Uppo et al. [14]–[16], trained a deep feedforward network to identify SNP interactions in high-dimensional data. This method exhaustively analyses higher-order interactions (from one-locus to ten-locus SNP interactions) on a sporadic breast cancer dataset [17]. Besides, a logistic regression filter is initially used to select a set of candidate SNPs. Results revealed the top 20 highly ranked interacting SNPs. Moreover, in this study, the accuracy of deep learning is compared with previously developed machine

learning approaches (RF, SVM, NN) and revealed better accuracy results. Still, the authors conclude that the performance of MLP models needs to be tested in the presence of noisy data.

Montaez et al. [18] used unsupervised learning algorithms to preselect a set of possible interacting SNPs on an obesity dataset. Further, an MLP is trained with the selected SNPs to evaluate if the selection was correctly made.

Additionally, Bellot et al. [19] preselected a set of possible interacting SNPs based on single-marker regression analysis and trained a set of different MLP architectures. The study compared the predictive performance of MLPs with CNNs and Bayesian linear regressors, concluding that Deep Learning networks have great potential when dealing with genomic prediction. Still, further investigation must be performed for DL to overcome current linear models.

2) Convolutional neural networks (CNNs): CNNs are also one of the most commonly used architectures from the state-of-the-art papers. CNN networks are a variant of MLPs proposed to solve complex problems such as text, image and speech recognition [13] and had been recently applied in epistasis detection.

Bellot et al. [19] preselected a set of possible interacting SNPs based on single-marker regression analysis and trained a set of different CNN architectures. The study compared the predictive performance of CNNs with MLPs and Bayesian linear regressors. Besides concluding that DL Algorithms show great potential in genomic prediction, the potential of CNN architectures with small 1D kernels (width of 2 or 3) is highlighted, declaring that these should be investigated in future works.

Uppo et al. [20], performed a similar analysis to [14]–[16] but instead of using an MLP, used a CNN. The results concluded that CNNs show high potential however further investigation must be performed.

Additionally, Salesi et al. [21] applied a selection of feature filtering methods to identify the most important SNPs. Once the most important features have been selected, a CNN is trained and tuned to make accurate predictions on the phenotype. The accuracy of the network is used as a performance measure for the filtering method. The study concludes that applying feature selection methods improves the performance of the DL models..

B. Deep Learning and Epistasis Detection

Deep Learning methods have proven to be valuable approaches due to their ability to extract information from the dataset and make accurate predictions on the phenotype. Despite the great potential of these methods, the lack of interpretability remains one of the major drawbacks of these approaches. A study on the state-of-the-art approaches applying DL models [14]–[16], [18]–[21] revealed that these methods are still treated as black-box predictive models. Accuracy is used to evaluate if epistatic interactions are being detected but the actual information extracted by the model is never interpreted.

To overcome the black-box nature of DL networks, model interpretation algorithms can be applied. Despite showing great potential in image recognition and other DL tasks [22], no evidence was found that these methods had been applied in epistasis detection. Model interpretation algorithms can assign a relevance score to each input based on their impact when making predictions on output, meaning that a relevance score could be assigned to each input SNP based on their impact when making predictions on the phenotype. If the network is correctly trained, the interacting SNPs will have the highest relevance and can be selected for further analysis. Hence, there is a gap in the literature regarding network interpretability in epistasis detection that must be filled since these algorithms consist of promising approaches and have demonstrated their value in other DL areas [23].

III. Methodology: Interpretability of Deep Learning Models

A methodology for interpreting DL models and detect the interacting SNPs is presented. Sensitivity analysis is introduced as a method for interpreting predictions and ranking SNPs according to their relevance when making predictions for a single sample. Further, this analysis is extended from individual samples to the entire dataset, to ensure SNP relevance is measured across all samples. If the networks are correctly trained, the interacting SNPs will have the highest relevance among all the input SNPs

A. Explaining Predictions: Sensitivity Analysis

Explaining DNN decisions represents an important step for interpreting DL algorithms. It allows asking the model, for a given sample, why the network classified it as belonging to a certain class.

When explaining predictions, a common approach is to consider a sample as a collection of features and assign a score to each. This score represents how relevant this feature is when the network is making predictions [22]. Relevance values are further represented as heatmaps, which allows visual identification of the most relevant features.

Relevance scores are calculated using sensitivity analysis. Sensitivity analysis is based on the model locally evaluated gradient, which is a measure of variation. Sensitivity can be defined as

$$R_i(x) = \left(\frac{\partial f(x)}{\partial x_i} \right)^2 \tag{3}$$

where x represents a sample, $f(x)$ is a function describing the network, x_i the feature i of sample x and R_i the relevance value of feature i. Thus, if x is the input layer, the relevance of each input feature can be determined by calculating the gradients of the output with respect to each input feature x_i.

In the context of epistasis detection, samples are patients and features the SNP values with the corresponding genotype values. This type of sensitivity analysis allows asking the question "What were the input SNPs which caused this patient genotype to be classified as a case?" or in more deep learning

language, "What were the input features which caused this sample to be classified as positive?". The pseudo-code of the algorithm used to perform sensitivity analysis in individual samples is presented in Algorithm 1. The output consists of a vector with each SNP and the corresponding relevance value.

Algorithm 1: Sensitivity analysis for one sample

input : A trained network model M and an input sample x with N SNPs
Casts x to a tensor of type float32
Initiate tf.GradientTape()
Make a prediction $M(x)$
Use GradientTape() to get the gradients of the output with respect to the input
Calculate sensitivity
output: Vector of size N with relevance values for each SNP

B. Epistasis Detection Algorithm

The main goal of epistasis detection consists in finding the interacting SNPs in a population, thus sensitivity analysis must be performed in the entire population and not in a single sample.

When evaluating an epistasis detection dataset, the same features are considered for each sample, meaning that for each patient the same SNPs are considered. Therefore, the sensitivity analysis between different samples calculates the relevance values for the same input SNPs but different patients. This is an advantage when compared to other deep learning tasks such as image recognition. In image recognition, each sample is a different image with different input features (pixels). The evaluation of the network predictions must be performed individually for each sample [22].

Sensitivity analysis can be performed across different samples and the results added. This method allows calculating SNP relevance across the entire dataset and determines which SNPs are interacting and causing the expression of a phenotype in a population. According to the previous definition and Equation 3, SNP relevance can be defined as:

$$R_i(x) = \sum_{j=0}^{S} \left(\frac{\partial f(x)}{\partial x_{ij}} \right)^2 \tag{4}$$

where S represents the set of samples where SNP relevance is measured.

The next step consists of defining the set of samples to perform sensitivity analysis. To understand how these samples are selected it is important to know the concept of true positives (TPs). TPs are samples having the disease (cases) which the model was able to correctly predict as cases in the testing stage. These samples are selected for sensitivity analysis since are the ones considered to have relevant information worth interpreting.

The cross-validation algorithm is used when training the network. This technique ensures that the test set used to evaluate the network performance covers all the dataset samples. Thus, in each iteration of the cross-validation algorithm, the relevance of the TP samples is calculated. Therefore, it is possible to obtain the interacting SNPs using:

$$R_i(x) = \sum_{k=0}^{C} \sum_{j=0}^{S} \left(\frac{\partial f(x)}{\partial x_{ji}} \right)^2 \tag{5}$$

where C represents all the interactions of the cross-validation algorithm. Once sensitivity analysis is performed for the entire population, SNPs are sorted in increasing order according to their relevance. The SNPs with the highest relevance will have the highest index. The objective of the method consists of having the interacting SNPs with the highest index on the sorted relevance vector. The pseudo-code to detect epistasis in a population is presented in Algorithm 2.

Algorithm 2: Epistasis detection using sensitivity analysis

input : An epistatic dataset with N SNPs
Vectors $Accuracy$, $F1\_score$, $Precison$, $Recall$, AUC initialized
Vector R of size N to store relevance values initialized
for $train\_data$, $test\_data$ ***in*** *Cross Validation* **do**
  Create DL Model
  Train model on $train\_data$
  Evaluate performance measures on $test\_data$
  Append Accuracy, F1_score, Precision, Recall and AUC to the corresponding vectors
  r = Calculate Sensitivity Analysis for $test\_data$
  Sums $R = R + r$
end
Calculate mean of $Accuracy$, $F1\_score$, $Precison$, $Recall$, AUC.
R_final = Order indexes (SNPs) increasingly R.
output: R_final, a vector of size N with the sorted indexes of R.

C. Interpretability Metric

A new performance measure is defined to classify networks in terms of interpretability. When evaluating the interpretability of DL on simulated datasets, the interacting SNPs are known, allowing the definition of interpretability (I) as:

$$I = \frac{\min_{\forall k \in K}(p_k)}{N} \tag{6}$$

where K represents the set of interacting SNPs, p_k the position of the interacting SNPs in the sorted relevance vector and N the number of input SNPs.

Since the main goal of the proposed methodology consists of finding all the interacting SNPs, only the minimum position across all interacting SNPs is considered. If an interaction of order k is considered, maximum interpretability is reached if the top k SNPs with the highest relevance, correspond to the interacting SNPs. Thus, the higher the position of the interacting SNPs in the sorted relevance vector the higher the interpretability. This way, the higher the minimum position across all interacting SNPs, the higher the interpretability value will be. Moreover, finding interacting SNPs in the presence of more input SNPs is harder, since there is a larger amount of noise. Thus, dividing by the number of input SNPs considers this factor and penalizes bigger errors with less amount of input SNPs.

IV. Experimental Results

To validate the previously proposed methodology on network interpretation and SNPs detection, MLP and CNN networks were evaluated under a wide variety of epistasis scenarios.

A. Datasets and Initial Setup

For generating the datasets, the GAMETES [5] generator and the Toxo [6] library were used. Two separate types of datasets were generated, ones displaying marginal effects (ME) and others with the absence of marginal effects (NME). For NME datasets, pairwise and higher-order datasets were generated, both using the additive, multiplicative, xor and threshold. The number of samples was set to 4000, with 2000 cases and 2000 controls, and the number of features in the datasets was set to 1000 SNPs. Also, the MAF of the non-interacting SNPs was set to a fixed range of [0.05, 0.5].

The values used for MAF were [0.05, 0.1, 0.2, 0.3, 0.4], similar to the ones used in [14]–[16], [24] but with more detailed intervals. For each MAF value, datasets with heritability of [0.01, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4] were generated. This way, the impact of MAF and heritability on interpretability could be evaluated.

When training neural networks a set of parameters must be defined before the training starts. In all experiments, the number of epochs was set to 1000, with a batch size of 32 samples. For early stopping the patience was set to 50 epochs, meaning that if the validation accuracy does not improve in the following 50 samples, the training stops.

All the experiments were conducted on the same system. The CPU was an i9-10980XE, with 128 GB of RAM and the GPU was an Nvidia TITAN RTX. All the networks were implemented using python 3.7.3 and Tensorflow 2.4.0. The training was performed on the GPU, using CUDA version 11.0.

B. Multilayer Perceptron (MLP)

1) Initial Search Space: Hyperparameter optimization is an important step in DL. The grid search method was used to exhaustively search for all possible combinations in a predefined search space. Since grid search is an exhaustive method, it was necessary to define a good search space with an efficient range of parameters. To ensure that a correct range of parameters was selected, only the most relevant state-of-the-art architectures were considered. Thus, only the architectures from studies [14]–[16], [19] were considered. The initial search space is presented in Table I.

2) Tests on NME Datasets: When dealing with non-exhaustive methods for SNP detection, it is more difficult to detect the interacting genes on an NME than on a ME dataset [5]. Thus, a detailed evaluation of the impact each hyperparameter has on the model interpretability was performed on NME datasets. This type of analysis allowed classifying model architectures in terms of interpretability performance while also reducing the initial search space. The first hyperparameter to be analysed was the number of inputs. Increasing the number of SNPs means that the number of noisy SNPs in the input increased, which made it more difficult for the networks to make accurate predictions.

TABLE I
Search space to evaluate MLP interpretability.

Architeture	Hyperparater	Range
MLP	Inputs	[50,100,500,1000]
	Nº Layers	[1,2,3,5]
	Nº Neurons	[32,64]
	Activation Function	[Elu, Tanh, Softplus]
	Dropout Ratio	0.03
	Learning Rate	1×10^{-3}

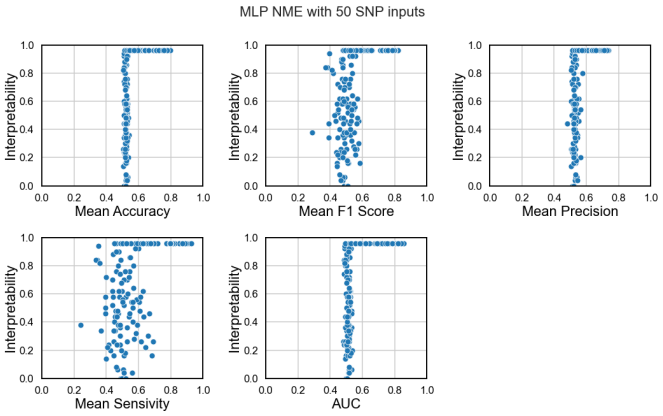

Fig. 2. MLP relation between interpretability and other performance measures on NME datasets with 50 input SNPs.

It was crucial to understand the relationship between other performance measures and interpretability since the latter cannot be calculated if the interacting SNPs are unknown. From the graph in Figure 2 correspondent to 50 input SNPs, a threshold relation between interpretability and some performance measures like accuracy, precision and AUC was observed.The same relation could be observed in models with 100 input SNPs. Networks having performance values above these defined thresholds had always maximum interpretability. The definition of these thresholds was crucial to understand if models could be trusted or not. When the number of inputs was increased to 500 and 1000 input SNPs, no threshold relation was observed between any performance measure and interpretability, meaning that no trustworthy networks were trained.

The threshold value from which maximum interpretability for every network was reached is presented in Table II. From the total networks tested (180), the percentage of networks whose performance values were above the defined threshold were also presented. From Table II it was concluded that accuracy and AUC had the same relation with interpretability since the percentage of networks with performance values above the presented thresholds was, in both accuracy and AUC, 43% for 50 inputs and 29% with 100 inputs. Precision revealed to have a worse relationship with interpretability since

the number of trustworthy networks is smaller (31% for 50 inputs and 23% with 100 inputs). From Table II the drop in performance associated with the increase of input SNPs was also confirmed since the number of trustworthy networks decreased across all performance measures.

TABLE II
MLP THRESHOLD VALUES FOR EACH PERFORMANCE MEASURE TO ACHIEVE MAXIMUM INTERPRETABILITY.

Nº Inputs	Performance Measure	Threshold Value	Max. Interpretability (%)
50	Accuracy	0.5425	43
50	Precision	0.5760	31
50	AUC	0.5386	43
100	Accuracy	0.5355	29
100	Precision	0.5570	23
100	AUC	0.5356	29

Additionally, an analysis of the activation functions and the number of layers hyperparameters was performed. Accuracy and interpretability were used to evaluate the different architectures since these have proven to be related and the results from networks having 500 and 1000 input SNPs were excluded since the models did not fit the training data.

Density curves representing the distribution of networks hyperparameters according to their interpretability value were plotted. It was concluded that the use of the softplus activation function and a reduced number of hidden layers was associated with networks with lower interpretability values. Thus, softplus and single-layer networks were removed from the search space. The pruned search space to evaluate the remaining datasets is represented in Table III.

TABLE III
SEARCH SPACE TO EVALUATE MLP INTERPRETABILITY.

Architeture	**Hyperparater**	**Range**
MLP	Inputs	[50,100,500,1000]
	Nº Layers	[2,3,5]
	Nº Neurons	[32,64]
	Activation Function	[Elu, Tanh]
	Dropout Ratio	0.03
	Learning Rate	1×10^{-3}

3) Tests on ME datasets: When compared to the NME datasets, ME datasets tend to be easier for non-exhaustive algorithms to detect interactions [5]. In these datasets, individual SNPs had information about the phenotypes, thus actual interactions did not need to be detected. Since the difficulty of detecting the interacting SNPs was lower on NME datasets, it was not necessary to perform a detailed analysis on hyperparameter performance as in ME datasets. Network architectures working on NME datasets would also work on ME datasets, due to their reduced complexity.

Tests on ME datasets were performed using the previously pruned search space from Table III. As expected, MLPs revealed less difficulty in detecting the interacting SNPs in the presence of marginal effects. The number of inputs did not have an impact on network performance and these were able to detect the interacting SNPs on four different epistasis models. Despite the drop in performance on higher-order datasets, the networks were still able to detect the interacting SNPs on different models and input values.

4) MAF and Heritability: A detailed analysis of MLP interpretability under different MAFs and heritability (h^2) values was performed.

From the box plot summarizing MLP performance in Figure 3, it was observed that an increase in heritability caused an improvement in network performance. All networks achieved maximum interpretability with $h^2 = 0.3$ and $h^2 = 0.4$, which was expected since by increasing heritability the amount of genetic information in the dataset increased and networks had less difficulty extracting it. On the other hand, for datasets having lower heritability values ($h^2 = 0.01$ and $h^2 = 0.05$), the networks did not fit the dataset at any MAF and most networks had low interpretability values. No conclusions could be derived from MAF.

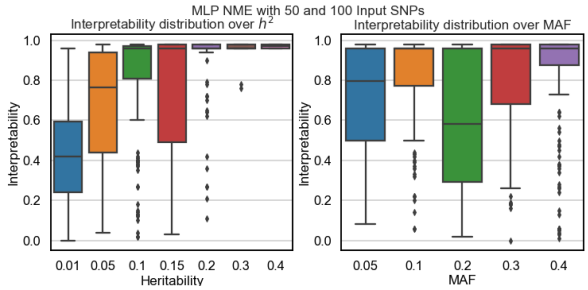

Fig. 3. MLP boxplot analysis on the impact of MAF and heritability on interpretablity.

C. Convolutional Neural Networks (CNN)

1) Initial Search Space: From the state-of-the-art works, the most significant ones were selected and the used network architectures considered. Grid search is applied to exhaustively search for the best network architecture. The considered works to define the initial search space were [19], [21].

When defining a search space for CNN networks, the number of hyperparameters to be considered was higher when compared to MLP architectures. If all hyperparameters were considered, a total number of 1152 architectures should have been tested for each dataset. This would make grid search very computationally demanding, thus, a different strategy was defined. Initially, the number of input SNPs was fixed to 50 input SNPs, reducing the number of tested architectures per dataset to 288. The impact of each hyperparameter was analysed and the search space was reduced.The initial search space is presented in Table IV.

2) Tets on NME datasets: The strategy used to evaluate MLP interpretability was also applied in CNN networks, meaning that NME datasets were used to analyse hyperparameter performance in terms of interpretability and to reduce the initial search space.

To understand the relation between interpretability and the other performance measures, scatter plots in Figure 4 are

TABLE IV
SEARCH SPACE TO EVALUATE CNN INTERPRETABILITY.

Architecture	Hyperparameter	Range
CNN	Inputs	[50]
	Nº Convolutional Layers	[1,2]
	Nº Filters	[16,32,64]
	Kernel Size	[1,2,3]
	Nº Classifier Layers	[32,64]
	Nº Classifier Neurons	[2,3]
	Activation Function	[Linear, Elu, Softplus, Relu]
	Dropout Ratio	0.01
	Learning Rate	1×10^{-3}

presented. As observed with the MLP architectures, Accuracy and AUC had a clear relation with interpretability, meaning that from a certain accuracy and AUC threshold, all networks show high interpretability values. For accuracy and AUC, the observed threshold values were 0.5425 and 0.5372 respectively with 27% of trustworthy networks for both performance measures. Precision, sensitivity and F1 score performance measures did not show any relation with interpretability. There was no precision, sensitivity or F1 score threshold from above which networks displayed maximum interpretability values.

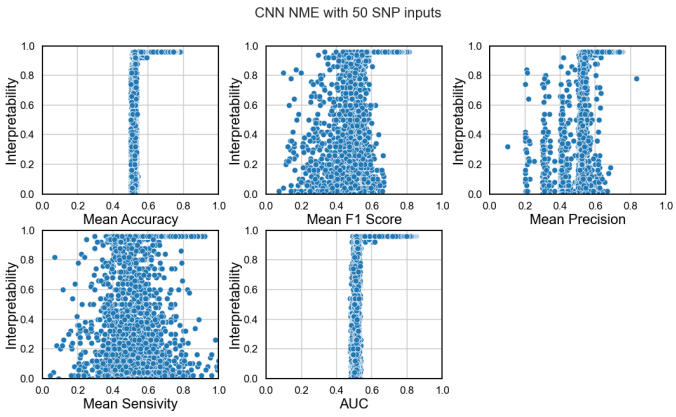

Fig. 4. CNN relation between interpretability and other performance measures on NME datasets with 50 input SNPs.

Further, an analysis of the activation function, number of classifier layers, number of convolutions and kernel size was performed. Accuracy and interpretability were used to evaluate the different architectures since these have proven to be related. Density curves were plotted to evaluate the architectures that were most frequently associated with lower interpretability values.

CNNs having Softplus and Linear activation functions and higher kernel sizes of three revealed to be more frequent in networks having low interpretability values. The number of classification layers proved not to have an impact on model interpretability, thus it was fixed to size three. Based on the previous conclusions, the Softplus and Linear activation functions were removed. Additionally, the kernel size of three was also removed. The new search space is presented in Table V. Regarding the better performance of smaller kernel sizes, in [19] it was concluded that smaller kernel sizes have better performance in epistasis detection.

TABLE V
SEARCH SPACE TO EVALUATE CNN INTERPRETABILITY.

Architecture	Hyperparameter	Range
CNN	Inputs	[50,100,500,1000]
	Nº Convolutional Layers	[1,2]
	Nº Filters	[16,32,64]
	Kernel Size	[1,2]
	Nº Classifier Neurons	[32,64]
	Nº Classifier Layers	[3]
	Activation Function	[Elu, Relu]
	Dropout Ratio	0.01
	Learning Rate	1×10^{-3}

Once the search space was reduced, the remaining input values of 100, 500 and 1000 input SNPs were tested. Increasing the number of inputs raises the difficulty of the tested dataset since the more noisy SNPs are added. The threshold values from which maximum interpretability for every network was reached are presented in Table VI. Also, from the total networks tested (480), the percentage of networks whose performance values were above the defined threshold (trustworthy) are also presented. Similar to MLP architectures, increasing the number of SNPs caused architectures to have a significant drop in performance. The number of trustworthy networks reduced from 47% to 11% and further to 6% with the increase of the number of inputs from 100 to 500 and 1000 SNPs.

Additionally, in CNN networks, the threshold relation between accuracy and interpretability was visible across 500 and 1000 input SNPs. Despite the reduced number of networks able to have maximum interpretability, this might suggest that CNNs are better at handling noisy datasets when compared to MLPs.

TABLE VI
CNN THRESHOLD VALUES FOR EACH PERFORMANCE MEASURE TO ACHIEVE MAXIMUM INTERPRETABILITY.

Nº Inputs	Performance Measure	Threshold Value	Max. Interpretability (%)
100	Accuracy	0.536	47
100	AUC	0.5337	47
500	Accuracy	0.5357	12
500	AUC	0.5317	13
1000	Accuracy	0.5425	6
1000	AUC	0.5366	6

3) Tests on ME Datasets: Tests on ME datasets were performed using the previously pruned search space from Table VI. As expected and observed in MLP architectures, CNNs had less difficulty detecting interacting SNPs in the presence of marginal effects. Networks were able to achieve high interpretability values, on both pairwise and high-order interactions, under different models and number of input SNPs. Despite the drop in performance on higher-order datasets, the

networks were still able to detect the interacting SNPs on different models and input values.

4) MAF and Heritability: First, an analysis of the impact heritability (h^2) was performed. Datasets having higher heritabilities have more information, thus it is easier for networks to fit the datasets.

Similar to the MLP networks, for datasets having low heritability values ($h^2 = 0.01$ and $h^2 = 0.05$), networks could not be trusted. No threshold relation between accuracy and interpretability was observed, meaning that no accuracy value from above which all networks had maximum heritability was found. With the increase of interpretability, network performance started getting better. For $h^2 \geq 0.1$, in almost all datasets the accuracy threshold was observed (for $MAF = 0.2$ and $h^2 = 0.1$ is not), meaning that there was a minimum accuracy from which networks could be trusted. Also, the plot from Figure 5, confirmed the better performance of networks for $h^2 \geq 0.1$.

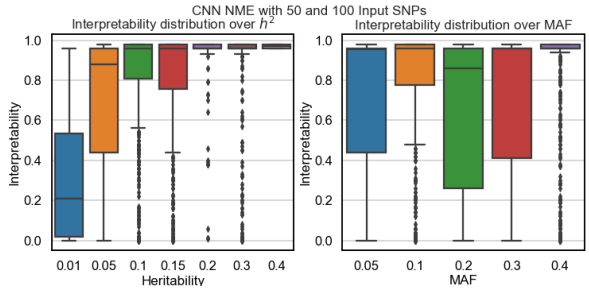

Fig. 5. MLP boxplot analysis on the impact of MAF and heritability on interpretablity.

D. Accuracy Threshold

In a real case scenario dataset, it is not possible to calculate interpretability since the interacting SNPs are unknown. Accuracy, F1 score, PrecisIon, Sensitivity and AUC are performance measures that can be calculated on datasets whose interacting SNPs are unknown. Hence, it was necessary to establish a relation between these measures and interpretability. This way, networks could be classified as trustworthy, or not, and sensitivity analysis could be performed to detect the interacting SNPs.

In the experimental analysis on MLP and CNN architectures, it was observed that interpretability was related to accuracy. In most cases, a threshold relation could be established between the two, meaning that there was an accuracy value from above which networks had always maximum interpretability. In that case, sensitivity analysis could be performed and the SNPs identified as most relevant considered as epistatic.

To detect the accuracy threshold value, for each MLP and CNN networks, the pruned search spaces in Table III and Table V were considered. The results on ME and NME datasets were grouped and analysed.

In Figure 6 the results for the MLP architectures are presented. It was observed that values with accuracies above 0.5478 had achieved maximum interpretability across all tested scenarios. The results for the CNN architectures are presented in Figure 7. The accuracy threshold for CNN networks was 0.54325. This means that any network with accuracy above 0.5482 achieved maximum or close to maximum interpretability values.

This was an important step in epistasis detection since it allowed the classification of networks as being trustworthy or not based on a performance measure that can be calculated on networks applied to a real genomic dataset. This way, using the defined search space for MLP (Table III) and CNN (V), if a network achieved an accuracy value over the defined thresholds (for MLP, an accuracy of 0.5478 and for CNN, an accuracy of 0.5482), networks could be interpreted and their decisions trusted.

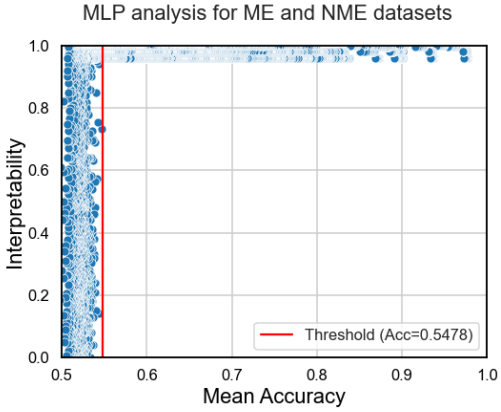

Fig. 6. MLP accuracy threshold from which interpretability values increase.

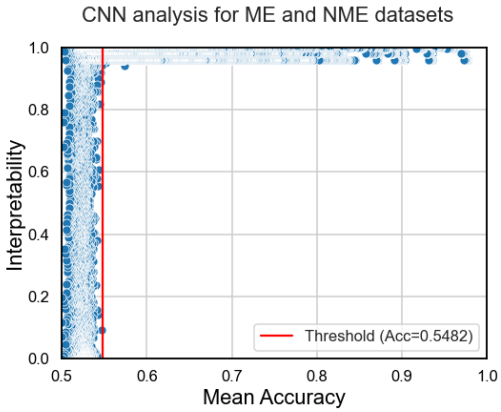

Fig. 7. CNN accuracy threshold from which interpretability values increase.

V. Application On A Real Dataset

In this section, MLPs and CNNs were applied in a real Breast Cancer Dataset [25] to evaluate if trustworthy networks were trained and if sensitivity analysis captured the interacting SNPs.

The considered Breast Cancer dataset has a total of 10000 samples, with 5000 cases and 5000 controls, with each sample

having 23 SNPs. The defined search spaces in Tables III and Tables V for MLP and CNN architectures were trained to fit the provided dataset. The networks whose accuracy was able to pass the threshold defined in the previous Section IV-D were interpreted and the most relevant SNPs analysed.

To validate the results, a previous study on the same Breast Cancer dataset is considered [25]. In [26] an exhaustive search algorithm was used to test all possible SNP combinations and detect the interacting SNPs. Interactions of order two (”rs2010204” ”rs1007590”), three (”rs2010204” ”rs1007590” ”rs660049”) and four (”rs2010204” ”rs0504248” ”rs660049” ”rs500760”) were found by the exhaustive search procedure. Thus, these interactions were considered and compared with the results obtained by MLP and CNN architectures.

First, the results were analysed on the MLP networks. No networks were able to overcome the defined threshold of 0.5478 defined in the previous Section IV-D, meaning that the trained networks could not be trusted.

Next CNN networks were considered. One network was able to reach the threshold value of 0.5482, meaning that this network could be trusted. In Figure 8 the plot displaying each SNP and the correspondent relevance is presented. The SNPs were sorted according to their relevance value. It was observed that interactions of order two had been successfully detected since SNPs ”rs2010204” and ”rs1007590” were considered the most relevant. Additionally, when considering interactions of order three which also include SNPs ”rs2010204” and ”rs1007590”, SNP ”rs660049” was considered the seventh most relevant among all the input SNPs. Also, for interactions of order four which include SNPs ”rs2010204” and ”rs660049”, SNP ”rs500760” was identified as the third most relevant and SNP ”rs0504248” as the fifth most relevant.

The results are very encouraging since the trained models were able to identify SNPs of different interaction orders as being among the most relevant. The CNN networks were able to correctly detect pairwise interactions as the two most relevant SNPs. Also, the interacting SNPs from orders three and four are all included in the top 7 most relevant SNPs. These seven most relevant SNPs represent the Top 30% across all the input SNPs. Thus, the method of sensitivity successfully captured interacting SNPs as relevant SNPs which highlighting the possibility of using this framework as a filtering method.

From the results, it was observed that if trained models achieve accuracy values over the defined thresholds, these could be interpreted. The test on a real Breast Cancer dataset confirmed the high potential of network interpretation methods to detected relevant SNPs or select them for further analysis.

VI. Conclusions

The main objective of this dissertation was to provide an analysis of network interpretability in epistasis detection. From the literature review in epistasis detection approaches, it was observed that deep learning methods were still treated as black-box predictive models, with accuracy being the only performance measure to evaluate the presence of interactions among the input SNPs. This way, the information extracted

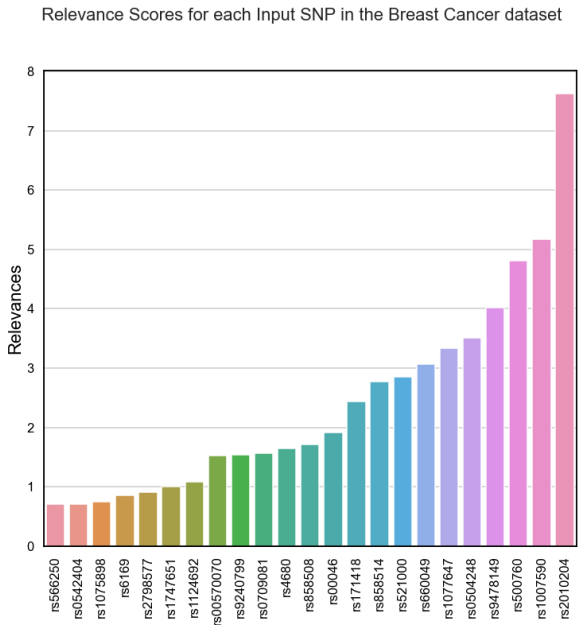

Fig. 8. Relevance scores for each input SNP in Breast Cancer dataset. The SNPs are ordered increasingly according to the relevance score obtained by sensitivity analysis.

by the network during training was not interpreted and the SNP interactions were not detected. Therefore, the field of network interpretability was explored to better understand deep learning models and their limitations regarding different epistasis scenarios. To evaluate network interpretability and extract information from network decisions, a new methodology was defined. Sensitivity analysis was used to interpret network decisions and assign each input SNP a relevance score. Once the interacting SNPs were detected, networks were classified using a new interpretability performance measure. The closest the network was to identify the correct solution, the higher its interpretability value was.

In real epistasis datasets, the interacting SNPs are unknown and the interpretability performance measure cannot be calculated. To allow the application of network interpretability on real datasets an accuracy threshold from which CNNs and MLPs could be trusted was detected. The results from ME and NME datasets were analysed together to detect an accuracy value from which networks achieve maximum interpretability. The detected accuracy thresholds for both CNNs and MLPs were 0.5478 and 0.5482 respectively. As a final step, MLP and CNN architectures were trained to detect interacting SNPs on a real Breast Cancer dataset. The method was able to identify pairwise interactions since the two SNPs identified as most relevant belonged to a pairwise interaction. For interactions of orders three and four, the interacting SNPs were all on the top seven of selected SNPs (Top 30%).

In this dissertation, the first steps on network interpretability applied in epistasis detection were taken. Interpretability of deep learning networks was tested under different scenarios to evaluate its limitations. However, there is still some aspects

of the developed framework that can be improved. Due to the great number of networks to be tested (it is necessary to define a search space with several networks), and the number of datasets tested (several values for heritability and MAF were evaluated), not all the desired tests were made. A wider variety of datasets including more epistatic scenarios could be included. Other epistasis models besides the Additive, Multiplicative, Threshold and Xor can be considered. For example, the use of the Color of Swine model, which is also very used in the literature, could be considered [24]. In this dissertation, the number of samples and the case-control ratio is fixed. These datasets hyperparameters can have a great impact on model performance and should be evaluated.

Additionally, an analysis of other model interpretation algorithms can be performed and compared. In [22], [23], other methods besides sensitivity analysis applied in other DL areas are suggested. Evaluating the performance of these algorithms could be the solution to extract the full potential of model interpretation methods in epistasis detection.

REFERENCES

[1] G. P. Consortium *et al.*, "A global reference for human genetic variation," *Nature*, vol. 526, no. 7571, pp. 68–74, 2015.

[2] P. M. Visscher, N. R. Wray, Q. Zhang, P. Sklar, M. I. McCarthy, M. A. Brown, and J. Yang, "10 years of gwas discovery: biology, function, and translation," *The American Journal of Human Genetics*, vol. 101, no. 1, pp. 5–22, 2017.

[3] C. Niel, C. Sinoquet, C. Dina, and G. Rocheleau, "A survey about methods dedicated to epistasis detection," *Frontiers in genetics*, vol. 6, p. 285, 2015.

[4] J. Shang, X. Wang, X. Wu, Y. Sun, Q. Ding, J.-X. Liu, and H. Zhang, "A review of ant colony optimization based methods for detecting epistatic interactions," *IEEE Access*, vol. 7, pp. 13497–13509, 2019.

[5] R. J. Urbanowicz, J. Kiralis, N. A. Sinnott-Armstrong, T. Heberling, J. M. Fisher, and J. H. Moore, "Gametes: a fast, direct algorithm for generating pure, strict, epistatic models with random architectures," *BioData mining*, vol. 5, no. 1, pp. 1–14, 2012.

[6] C. Ponte-Fernández, J. González-Domínguez, A. Carvajal-Rodríguez, and M. J. Martín, "Toxo: a library for calculating penetrance tables of high-order epistasis models," *BMC bioinformatics*, vol. 21, no. 1, pp. 1–9, 2020.

[7] S. Uppu, A. Krishna, and R. P. Gopalan, "A review on methods for detecting snp interactions in high-dimensional genomic data," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 15, no. 2, pp. 599–612, 2016.

[8] R. Upstill-Goddard, D. Eccles, J. Fliege, and A. Collins, "Machine learning approaches for the discovery of gene–gene interactions in disease data," *Briefings in bioinformatics*, vol. 14, no. 2, pp. 251–260, 2013.

[9] C. L. Koo, M. J. Liew, M. S. Mohamad, M. Salleh, and A. Hakim, "A review for detecting gene-gene interactions using machine learning methods in genetic epidemiology," *BioMed research international*, vol. 2013, 2013.

[10] K. Van Steen, "Travelling the world of gene–gene interactions," *Briefings in bioinformatics*, vol. 13, no. 1, pp. 1–19, 2012.

[11] B. A. McKinney, D. M. Reif, M. D. Ritchie, and J. H. Moore, "Machine learning for detecting gene-gene interactions," *Applied bioinformatics*, vol. 5, no. 2, pp. 77–88, 2006.

[12] S. Tuo, H. Chen, and H. Liu, "A survey on swarm intelligence search methods dedicated to detection of high-order snp interactions," *IEEE Access*, vol. 7, pp. 162229–162244, 2019.

[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[14] S. Uppu, A. Krishna, and R. P. Gopalan, "A deep learning approach to detect snp interactions.," *JSW*, vol. 11, no. 10, pp. 965–975, 2016.

[15] S. Uppu, A. Krishna, and R. P. Gopalan, "Towards deep learning in genome-wide association interaction studies.," in *PACIS*, p. 20, 2016.

[16] S. Uppu and A. Krishna, "Improving strategy for discovering interacting genetic variants in association studies," in *International Conference on Neural Information Processing*, pp. 461–469, Springer, 2016.

[17] M. D. Ritchie, L. W. Hahn, N. Roodi, L. R. Bailey, W. D. Dupont, F. F. Parl, and J. H. Moore, "Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer," *The American Journal of Human Genetics*, vol. 69, no. 1, pp. 138–147, 2001.

[18] C. A. C. Montaez, P. Fergus, A. C. Montaez, A. Hussain, D. Al-Jumeily, and C. Chalmers, "Deep learning classification of polygenic obesity using genome wide association study snps," in *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2018.

[19] P. Bellot, G. de los Campos, and M. Pérez-Enciso, "Can deep learning improve genomic prediction of complex human traits?," *Genetics*, vol. 210, no. 3, pp. 809–819, 2018.

[20] S. Uppu and A. Krishna, "Convolutional model for predicting snp interactions," in *International Conference on Neural Information Processing*, pp. 127–137, Springer, 2018.

[21] S. Salesi, A. A. Alani, and G. Cosma, "A hybrid model for classification of biomedical data using feature filtering and a convolutional neural network," in *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pp. 226–232, IEEE, 2018.

[22] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1–15, 2018.

[23] G. Eraslan, Ž. Avsec, J. Gagneur, and F. J. Theis, "Deep learning: new computational modelling techniques for genomics," *Nature Reviews Genetics*, vol. 20, no. 7, pp. 389–403, 2019.

[24] P.-J. Jing and H.-B. Shen, "Macoed: a multi-objective ant colony optimization algorithm for snp epistasis detection in genome-wide association studies," *Bioinformatics*, vol. 31, no. 5, pp. 634–641, 2015.

[25] C.-H. Yang, Y.-D. Lin, L.-Y. Chuang, and H.-W. Chang, "Evaluation of breast cancer susceptibility using improved genetic algorithms to generate genotype snp barcodes," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 10, no. 2, pp. 361–371, 2013.

[26] R. Campos, D. Marques, S. Santander-Jiménez, L. Sousa, and A. Ilic, "Heterogeneous cpu+igpu processing for efficient epistasis detection," in *European Conference on Parallel Processing*, pp. 613–628, Springer, 2020.