Sensitivity Analysis for Deep Learning Models Interpretability in Epistasis Detection

Bernardo Ferreira Mendes Cotovio de Bastos

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Prof. Aleksandar Ilic
Prof. Sergio Santander-Jiménez

Examination Committee

Chairperson: Prof. Teresa Maria Sá Ferreira Vazão Vasques
Supervisor: Prof. Aleksandar Ilic
Member of the Committee: Prof. Rui Miguel Carrasqueiro Henriques

September 2021

To my grandparents Aldino Cotovio and Manuela Cotovio

Declaration:

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the *Universidade de Lisboa*.

Declaração:

Declaro que o presente documento é um trabalho original da minha autoria e que cumpre todos os requisitos do Código de Conduta e Boas Práticas da Universidade de Lisboa.

Acknowledgments

First, I would like to thank my supervisors, Professor Aleksandar Ilic and Professor Sergio Santander-Jiménez, for the amazing guidance, availability and patience. Their helpful feedback and insight was crucial to the development of this Thesis. Also, I would like to thank INESC-ID Lisbon for the provided resources which made this Thesis possible.

A special thanks to my mother, father and brother, for the continuous love and support. Thank you for keeping my head up and always being there for me. Without you, none of this would have been possible. Also, I would like to thank my grandparents, for their support, advice and endless love during my studies and every day of my life.

To all my friends who were with me during my studies. Thank you for the laughs, dinners, sunsets, late-night studies and support. Those are moments I will remember for the rest of my life and I am sure many others will come.

Last but not least, a special thanks to Rita for the endless support, patience and for putting a smile on my face.

Resumo

Avanços recentes feitos por estudos de associação do genoma completo têm sido essenciais para a descoberta de associações entre Polimorfismos de Nucleótico Único (SNPs) e a manifestação de certas doenças. As interações entre SNPs são denominadas de Epistasia, e a sua deteção consiste, atualmente, num dos maiores desafios epidemiologia genética. A deteção de epistasia é um problema complexo cujos métodos tradicionais de estatística não conseguem resolver. Recentemente, devido à sua habilidade de extrair informação dos dados sem recorrer a métodos de procura exaustiva, redes de aprendizagem profunda têm sido aplicadas na previsão de doenças. No entanto, o facto de serem modelos complicados de interpretar (caixa preta) ainda consiste numa das maiores desvantagens destes métodos. Nesta dissertação, um novo método que permite a interpretação da informação extraída pela rede é apresentado. Análise da sensibilidade é utilizada para atribuir uma pontuação de relevância a cada SNP. Através da análise dos resultados em dados com e sem a presença de efeitos marginais, é possível estabelecer um limiar de precisão (accuracy) acima do qual é seguro interpretar os resultados da rede. Para MLPs e CNNs o limiar detetado foi 0.5482 e 0.5478, respetivamente. Para finalizar e confirmar os resultados, o método desenvolvido foi aplicado em dados reais de Cancro da Mama e os resultados comparados com um estudo recente que aplicou procura exaustiva nos mesmo dados. Os resultados identificaram os SNPs ”rs2010204”, ”rs1007590”, ”rs660049”, ”rs0504248” e ”rs500760”, presentes em interações de ordem dois três e quatro, entre os 30% SNPs mais relevantes.

Palavras-chave: deteção de epistasia, estudos de associação do genoma completo, aprendizagem profunda, interpretação de modelos, análise sensitiva, interações de ordem superior

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.

Keywords: epistasis detection, genome-wide association study, deep learning, model interpretability, sensitivity analysis, higher-order interactions

Contents

List of Tables

List of Figures

Acronyms

ACO Ant Colony Optimization. 5, 16, 26

AI Artificial Intelligence. 13, 16

CNN Convolutional Neural Network. ix, xi, xvii–xx, 4, 5, 17, 20, 21, 23, 26, 30, 31, 37, 52, 53, 56–67, 69, 73, 74, 87

DL Deep Learning. 2, 3, 5, 18–28, 30–32, 34–36, 38, 40, 52, 73, 75

GE Grammatical Evolution. 15

GWAS Genome-wide Association Studies. 1, 3, 5–7, 11, 22, 25

MAF Minor Allele Frequency. 9, 28, 35, 38, 40, 48, 49, 52, 70, 73

ME Marginal Effects. xix, xx, 7, 28–30, 35, 37, 38, 42, 47–50, 52, 59–63, 65, 69, 74

ML Machine Learning. 13, 16

MLP Multilayer Perceptron. ix, xi, xvii–xix, 4, 5, 17, 19–21, 23, 26, 30, 31, 37, 40, 41, 43–54, 58, 60, 62–67, 73, 74, 85–87

NME Non Marginal Effects. xvii, xix, xx, 8, 28, 35, 37, 38, 40–49, 52, 53, 56, 57, 59, 60, 63–65, 74

NN Neural Network. 17, 18

RF Random Forest. 5, 13, 14, 16, 20, 23, 25, 26

SIS Swarm Intelligence Search. 15, 26

SNP Single Nucleotide Polymorphism. ix, xi, xix, xx, 1–3, 5–9, 11–16, 18–28, 30–38, 41–50, 52, 53, 55, 56, 58–62, 64, 65, 67–69, 73, 74

SVM Support Vector Machines. 5, 14–16, 20, 26

T2D Type 2 Diabetes. 1, 7, 15

Chapter 1

Introduction

In recent years, the association between certain genetic markers and a disease phenotype has become a crucial step in the diagnosis and treatment of certain diseases [1]. Single nucleotide polymorphisms (SNPs) are the most common genetic variant in the human genome [2]. These genetic variants have been studied regarding their association with the presence of certain phenotypes in a population. These studies are denoted as genome-wide association studies (GWAS) and have gained increasing popularity in the past years [3].

GWAS studies tend to individually evaluate the relation of an SNP with the phenotype. Thus, when evaluating Mendelian diseases, which are only caused by the expression of a single SNP, these studies had shown good results. However, complex diseases like type 2 diabetes (T2D) and obesity are often caused by non-linear interactions between multiple genes [4]. Interactions between genes are referred to as epistasis. Thus, when studying complex diseases, epistatic interactions between SNPs must be considered.

In response to the need of discovering epistasis interactions, a wide variety of methods have been suggested. Several approaches including exhaustive search, machine learning and artificial intelligence methods have been proposed, each one providing a different perspective on how the problem should be approached.

1.1 Motivation

The detection of gene interactions has become an important field of research. Though, the detection of epistatic interactions in real genomic datasets is a complex challenge. GWAS datasets are complex high dimensional datasets with a large number of SNPs to be tested and a limited number of samples [5, 6]. Thus, detecting all possible epistatic interactions in these datasets can quickly turn into a combinatorial overload that current computers and traditional statistical methods cannot handle [7].

To overcome some of the epistasis detection challenges, several machine learning and artificial intelligence methods have been proposed. These methods are able to detect epistatic interactions without the need to examine every SNP combination and have revealed to be able to capture gene interactions. However, despite being promising approaches, these methods still face difficulties applied to the big and complex genetic datasets [8, 9].

Deep learning is an emerging field with several applications in genomics [10]. The ability to extract deep features from the data makes these approaches promising tools for detecting SNP interactions. Studies using deep learning architectures have revealed that these methods can capture SNP interactions and make accurate predictions on the phenotype [11]. Still, deep learning methods are hard to interpret, meaning that analysing which SNPs have the most impact when predicting the phenotype is a complex task. Model interpretation algorithms can be used to evaluate neural network decisions and detect the interacting SNPs, however, no evidence was found on the application of these methods in epistasis detection. Thus, there is a gap in the literature regarding model interpretation that must be explored.

The scope of this Thesis consists of developing a framework to allow model interpretation and overcome the black-box nature of DL models. Sensitivity analysis is used to interpret network individual decisions and evaluate the interacting SNPs in a population. A new interpretability performance measure is defined. This way, network performance can be evaluated not only in accuracy but also in interpretability.

Additionally, to test the limitations of DL models in terms of interpretability, several datasets simulating different epistasis scenarios are created. On a real dataset application, the interacting SNPs are unknown, thus, the interpretability performance measure cannot be calculated. A threshold between interpretability and other performance measures that can be calculated on a real dataset must be established. This way, networks having performance values above the defined threshold can be trusted and their results interpreted.

To conclude, the defined methodology is applied to a real Breast Cancer dataset. To validate the results, these are compared with a previous study on the same dataset. Hence, providing new insight into network interpretability and its limitations is one of the main contributions of this dissertation.

1.2 Objectives

From the state of the art review, it is concluded that there is a gap to be filled related to deep learning models interpretability. This aim of this Thesis is provide new insight into deep learning networks interpretability and evaluate its limitations. Hence, the objectives of this dissertation are:

- Provide an overview on the different state-of-the-art methods for epistasis detection.
- Identify the different types of Deep Learning architectures used in the state-of-the-art methods for detecting gene-gene interactions.
- Define a framework that allows Deep Learning models to be interpreted and classify SNPs according to their impact on predicting the phenotype.
- Define a new performance measure that allows networks not only to be evaluated not only on accuracy but also on interpretability.

- Study the networks interpretability performance under different simulated epistatic scenarios.
- Establish a threshold relation between interpretability and other performance measures to allow the application on real epistasis datasets.
- Validate the approach on a real Breast Cancer dataset.

1.3 Contributions

The main objective of this dissertation is to provide an analysis of network interpretability in epistasis detection. To evaluate network interpretability and extract information from network decisions, a new methodology is defined. Sensitivity analysis is used to interpret network decisions and assign each input SNP a relevance score. These relevance scores are calculated throughout the entire dataset and added. The final result is a vector of increasingly ordered SNPs according to their relevance values. Once the interacting SNPs are detected, networks are classified using a new interpretability performance measure. The closest the network was to identify the correct solution, the higher its interpretability value was. This type of approach allows the classification of networks not only in terms of accuracy but also in terms of interpretability.

This type of analysis provides new insight into DL models interpretability in detecting SNP interactions. By testing networks under a different set of epistatic scenarios, it is possible to evaluate DL networks limitations in terms of interpretability and evaluate the conditions at which DL predictions can be trusted. The conclusions are further validated on a real Breast Cancer dataset. Thus, it is believed that this Thesis provides a valuable insight into network interpretability and its limitations in epistasis detection.

1.4 Thesis Outline

This report is structured as follows:

- **Chapter 2 - Background: Epistasis:** This chapter presents an overview of the state of the art methods for detecting gene-gene interactions. An introduction to gene expression and how the human genome is organised is provided in the first place. Then, the idea of SNPs is introduced. The presence of SNPs GWAS and the concept of epistasis are explained. An overview of how epistasis is mapped into computers is also provided. Further, exhaustive search methods are introduced as a brute-force approach to discover epistasis detection. The limitations of these methods and epistasis detection challenges are also discussed. Machine learning and artificial intelligence methods are introduced as promising methods to handle epistasis detection challenges. Moreover, an overview of swarm intelligence approaches used to detect gene interactions is also presented. Finally, a more detailed evaluation of deep learning approaches is provided. Multilayer perceptrons (MLP), Convolutional Neural Networks (CNNs), hybrid methods and model interpretation approaches are introduced. The training, hyperparameter optimization, limitations and further applications of deep learning architectures are also addressed.

- **Chapter 3 - Methodology: Interpretability of Deep Learning Models:** Based on the previous analysis on the state-of-the-art methods, a new methodology based on Deep Learning interpretability is proposed. Sensitivity analysis is used to explain individual network decisions and classify SNPs as interacting or non-interacting. A new interpretability performance measure is introduced to classify networks not only based on accuracy but also on the information learnt from the data. The different types of datasets created to simulate a wide variety of epistasis scenarios are also presented.

- **Chapter 4 - Experimental Results:** The results of Deep Leaning models interpretability on the variety of datasets created is provided in this chapter. CNNs and MLPs interpretability is evaluated on datasets with and without marginal effects. On datasets with marginal effects both pairwise and high order datasets are considered. Since the interpretability performance measure cannot be calculated on real datasets, a threshold relation between other performance measures and interpretability is established. Thus, any network having a performance value over the defined threshold can be trusted. To conclude and validate the approach, the results are applied in a real Breast Cancer dataset.

- **Chapter 5 - Conclusions and Future Work:** A summary of the conclusions from this dissertation is presented. A discussion on further improvements for future work is discussed.

Chapter 2

Background: Epistasis

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 [3]. GWAS are case-control studies which 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 [7].

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 [12].

For a better understanding of the problem, a brief introduction to gene expression is provided. Further, SNPs are introduced, clarifying their relation to GWAS and providing a clear definition of epistasis detection. An overview of how epistasis is mapped into computers is also provided. Afterwards, exhaustive search methods are introduced as the most accurate approaches to solve epistasis and challenges associated with epistasis detection are discussed.

Once presented the challenges related to epistasis detection, other state-of-the-art approaches that do not rely on exhaustive search methods are introduced. A general overview of machine learning and artificial intelligence methods applied to epistasis detection is provided. Random forest (RF), support vector machines (SVM) are introduced. Also, methods based on swarm intelligence are presented, with the main focus on ant colony optimization (ACO) methods.

Following, a detailed overview of deep learning methods (DL) is provided. Several state-of-the-art methods that employ deep learning for epistasis detection are analysed and classified into distinct categories. To that respect, MLPs, (CNNs, hybrid and model interpretation approaches are discussed in this section. Additionally, the strategies involved in hyperparameter optimization and training are also analysed.

In each section, the strengths, limitations and further improvements of these methods are also discussed.

2.1 Gene Expression

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, there are many genes that 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).

2.2 Single Nucleotide Polymorphisms (SNPs)

SNPs are the most common genetic variant in the human genome. In [2], a study providing the description of 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. Thus, the development of SNP maps is an important step in the identification of genes involved in complex diseases [1].

2.2.1 Genome-Wide Association Studies (GWAS)

GWAS are case-control studies which identify SNPs which are strongly associated with a phenotype in a population. These studies are becoming more popular, as they are crucial to the understanding and treatment of genetic diseases [7].

The association between SNPs and the phenotype is not causal in most cases. Complex diseases such as T2D or obesity are caused by multiple SNPs, each which a small per-SNP effect. Thus, associating SNPs with a phenotype is a challenging process as the impact of SNPs must, not only, be evaluated individually, but also, the interactions between them. Notably, many SNPs are located in non-coding DNA regions, which proves the fact that these must be indirectly involved with the phenotype [4]. Thus, gene-gene interactions must be evaluated when evaluating complex diseases.

Several common diseases have been a subject of GWAS such as Alzheimer's disease [13], Crohn's disease [14], age-related macular degeneration [5] and prostate cancer [15]. Still, a large subset of SNPs related to complex diseases has not been identified yet. Besides being a complex problem due to the presence of interactions between SNPs, these disease-related SNPs have a low population frequency, making the GWAS sample size small when compared to the high number of SNPs per sample [7].

2.2.2 SNP-SNP Interactions: Epistasis

An important goal of human genetics consists in identifying a DNA sequence that is responsible for or increases the probability of a disease being expressed. Unlike Mendelian diseases, which are only caused by the expression of a single SNP, complex diseases are often caused by non-linear interactions between multiple genes. Thus, it is essential to have a clear definition of what is epistasis and how to define gene interactions.

Biological epistasis was firstly introduced by Bateson in [16] where a phenotype controlled by two alleles was identified. This work presented a breakthrough on how to understand allele interactions and how the behaviour of an allele can suppress the other. Despite being introduced as epistasis, today Bateson work is presented as the concept of allele dominance at an inter-loci level. Since then, epistasis definition has been changing its meaning. A more recent approach says biological epistasis occurs when the effect of an allele of a particular genetic variant depends on the presence or absence of another genetic variant [7].

Deviations from the biological concept of epistasis emerged once statistical models started to be applied as a means to understand disease behaviour. Statistical epistasis was introduced in 1918 by Fisher [17], and its main goal consists in finding statistically relevant interactions in a way to get closer to their biological meaning. The average deviation of allele combinations between different loci in a population is estimated, in order to associate those combinations to the expression of a certain phenotype [18]. Still, the transition between statistical and biological epistasis is not a simple process meaning that statically relevant interactions always need to be validated through biological experiments. Thus, not all statistical relevant interactions actually occur at a biological level [8].

When considering genetic interactions, other concepts such as marginal effects and heterogeneity need 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). Heterogeneity occurs when distinct SNPs represent different causes of the disease. Thus, in the presence of heterogeneity distinct SNPs interact with the phenotype separately, without interaction effects between them [19].

2.3 Epistasis Mapped Into Computers

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 2.1): one representing the SNP data and the other the labelling data [20]. 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).

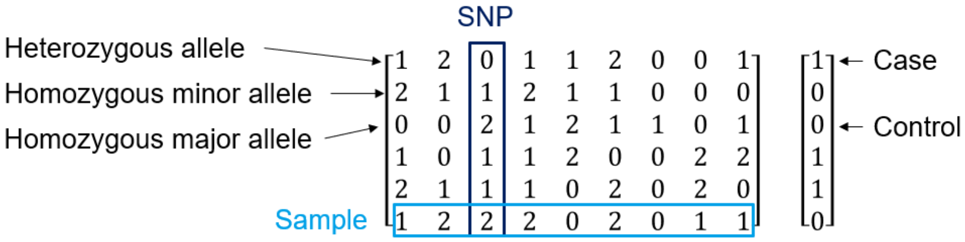
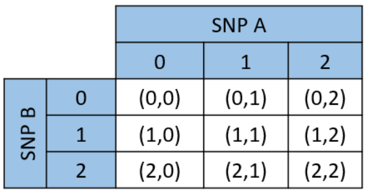


Figure 2.1: Mathematical matrices used to represent genomic data.

Interactions between SNPs are represented as genotype combinations, with the number of SNPs per interaction denoted as order. Tow-order interactions are often classified as pairwise SNP interactions and interactions with order greater than two are designated as higher-order SNP interactions.

In Figure 2.2, pairwise interactions between $SNPA$ and $SNPB$ are represented. Since each SNP can have three possible genotypes (0, 1 or 2), a total of nine possible genotype combinations is presented. If the interaction order is increased to three, the number of genotype combinations will be twenty-seven. Thus, when detecting higher-order SNP interactions, the number of genotype combinations to be tested grows exponentially which turns epistasis detection into a combinatorial optimization problem [21].



		SNP A		
		0	1	2
SNP B	0	(0,0)	(0,1)	(0,2)
	1	(1,0)	(1,1)	(1,2)
	2	(2,0)	(2,1)	(2,2)

Figure 2.2: Genotype combinations between two order SNP interactions.

To evaluate the association between each SNP combination and the phenotype, objective functions are introduced. These functions statistically evaluate each SNP combination and classify them as interacting or non-interacting. A wide variety of objective functions has been used and there is still no consensus about which one is the most appropriate [20, 21]. Still, the *K2 score* test, which is a Bayesian network scoring model, is a commonly used approach [22].

2.3.1 Penetrance Function

When simulating epistatic interactions, the most common way of representing epistatic relationships is by using a penetrance table. A penetrance table contains the probability of having a particular phenotype for each allele combination. When generating datasets, generators like GAMETES [23] and Toxo [24] use these tables to express epistasis relations and generate epistasis datasets.

Table 2.1 represents an example of a penetrance table generated by GAMETES, where each value of the table corresponds to the probability of having the phenotype for each allele combination between $SNPA$ and $SNPB$. Since these values represent a probability, they must be in the interval $[0, 1]$.

Table 2.1: Penetrance Table without marginal effects [23].

			SNP B		Marginal
	Genotypes	BB (0.25)	Bb (0.5)	bb (0.25)	Penetrance
SNP A	AA (0.36)	0.266	0.764	0.664	0.614
	Aa (0.48)	0.928	0.398	0.733	0.614
	aa (0.16)	0.456	0.927	0.147	0.614
Marginal Penetrance		0.614	0.614	0.614	

Additionally, the Minor Allele Frequency (MAF) of each SNP is also represented. This value denotes the frequency at which the second most common allele occurs in a population. Both GAMETES and Toxo use the assumption of Hardy-Weinberg Equilibrium (HWE), meaning that genotype frequencies can be calculated as:

$$\begin{cases} f(AA) = p^2 \\ f(Aa) = pq \\ f(aa) = q^2 \\ p + q = 1 \end{cases} \tag{2.1}$$

where $f()$ represents the frequency, q the MAF of the recessive allele 'a', and p the frequency of the dominant allele 'A'.

The example provided in Table 2.1 represents an epistasis model without the presence of marginal effects, meaning that between the two interacting SNPs, none of them is individually interacting with the phenotype. The marginal penetrance of each genotype can be calculated by the dot product between the frequency vector and the correspondent penetrance value vector. Thus, the marginal penetrance of genotype 'AA' is given by:

$$(0.25, 0.5, 0.25) \cdot (0.266, 0.764, 0.664) = 0.25 \times 0.266 + 0.5 \times 0.764 + 0.25 \times 0.664 = 0.614. \tag{2.2}$$

If the same expression in 2.2 is applied to calculate the marginal penetrance of genotypes 'Aa' and 'aa', the values will be the same. Since all the marginal penetrances are equal for both $SNPA$ and $SNPB$, the model in Table 2.1 is considered to not have any marginal effects.

2.3.2 Parameters

When building penetrance tables and generating epistasis datasets using generators like GAMETES and Toxo, it is necessary to understand the concept of heritability and prevalence.

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 [23]. According to [24], heritability can be obtained by:

$$h^2 = \frac{\sum_i (P(D|g_i) - P(D))^2 P(g_i)}{P(D)(1 - p(D)))} \tag{2.3}$$

$$P(D) = \sum_i P(D|g_i)P(g_i) \tag{2.4}$$

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 2.4).

Moreover, generators like Toxo also require the definition of an additional epistasis model. Epistasis models are mathematical expressions that define the penetrance table values. Marchini et al. [25] proposed several epistasis models. Table 2.2 provides an example of a Threshold model for a pairwise interaction between $SNPA$ and $SNPB$.

Table 2.2: Threshold model penetrance table.

Genotypes	AA	Aa	aa
BB	α	α	α
Bb	α	$\alpha(1+\theta)$	$\alpha(1+\theta)$
bb	α	$\alpha(1+\theta)$	$\alpha(1+\theta)$

Also worth mentioning that since the penetrance values must be in the interval $[0, 1]$, not every combination of heritability, prevalence and model produces a valid penetrance table. Thus, both GAMETES and Toxo have this limitation.

2.4 Exhaustive Search Methods

In response to the challenge of finding gene interactions related to a phenotype, exhaustive search methods started to emerge. Exhaustive search methods analyze all possible combinations of SNPs to determine the most accurate solution, thus avoiding the final result to be a sub-optimal solution.

For a better understanding of these algorithms, the concept of a contingency table needs to be clarified. A contingency table is a matrix displaying the frequency distribution of all possible genotypes between the interacting SNPs. Hence, in the case of two order SNP interactions, the contingency table will be a 3×3 matrix [7]. These tables are used in some exhaustive search methods to rank the importance of each SNP interaction (e.g. by calculating the log-likelihood ratios [26]) when predicting the phenotype. Consequently, for each SNP interaction to be tested, a contingency table has to be build which makes exhaustive search methods computationally demanding, when applied to high dimensional genomic data and to the detection of high order SNP interactions.

To make the use of contingency tables in GWAS datasets possible, Wan et al. [26] proposed a Boolean operation-based testing and screening method (BOOST). In this approach, the input data and the contingency tables are represented in a binary way meaning that computer operations are faster and easily performed. This way the exhaustive search becomes less demanding in terms of time and space. Further, adaptations of the algorithm using graphical processor units (GPU) were proposed in [27, 28], which made the algorithm even faster. Moreover, TEAM [29] proposed a more efficient way of building contingency tables, without the need of examining all individuals. This approach revealed to be more efficient than brute-force approaches. Lastly, BiForce [30] combined the use of contingency tables with multithreading and efficient data structures to improve its performance.

Multifactor dimensionality reduction (MDR) has been greatly used and adapted to detect gene-gene interactions. MDR are supervised learning methods which do not need to estimate any parameters (non-parametric) and do not make any assumptions on the genetic model (model-free) [7]. These methods build tables similar to contingency tables to classify each the each genotype combination as high or low risk. In the case of two-order SNP interactions, there are nine possible genotypes, thus, the table will have nine cells corresponding to each genotype. Based on their classification, cells are merged and the dimensionality of the problem is reduced. In [31] a detailed view of different MDR approaches in the context of gene interactions is provided. Despite having great performance in reducing the dimensionality of the data, these approaches are still exhaustive having poor performance detecting higher-order SNP interactions.

Approaches based on exhaustive search are efficient in handling two, at maximum three-order SNP interactions. Scaling exhaustive search to detect higher-order interactions results in a combinatorial overload that current computers can not handle. Still, the application of these methods in two-stage methods is promising and currently used. Once a set of promising SNPs is selected from the original data, exhaustively searching for higher-order SNP interactions becomes more computationally tractable.

2.5 Challenges in Epistasis Detection

Detection of higher-order epistatic interactions is more challenging when compared to pairwise 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. This increase in complexity is what makes the previously discussed exhaustive search methods not computationally efficient when applied to higher-order epistasis detection. Hence, when considering high-order interactions at a genome-wide scale several challenges need to be taken into consideration. These can be divided into three distinct categories: statistical, computational and interpretation [7].

Statistical challenges arrive when considering traditional statistical methods that evaluate each SNP individually and its association to the phenotype. These approaches only detect SNPs with high marginal effects, thus missing interactions between SNPs which have a small effect per SNP on the phenotype. Hence, these approaches only have great success when identifying single-locus SNPs responsible for some Mendelian diseases [32]. As a result, when considering more complex diseases with several interacting SNPs and underlying epistasis, these methods are not able to detect SNP interactions.

The second challenge is computational. At a genome-wide scale analyzing all potential epistatic interactions usually turns into a combinatorial overload and result in a heavy computational burden. The increase in the order of tested interactions provokes an exponential increase in the number of potential interactions to be tested [12]. This high exponential growth in complexity is what makes higher-order epistasis detection an *NP-Hard* problem. For those reasons, exhaustive search methods are not computationally feasible for higher-order epistasis detection, when considering computational resources available today.

The third challenge can be divided into results interpretation and model interpretation. The results interpretation challenges are due to the fact that the translation between statistical and biological results is not direct. To this respect, experiments need to be performed to evaluate the veracity of the results, since not all statistically relevant interactions are true biologically. Thus, results from statistical methods cannot be interpreted directly. Model interpretation challenges are considered when dealing with disease predictive models (e.g. Deep Neural Networks). In genomic and epistatic contexts, it is most important to understand the disease-causing genes rather than being able to just predict it [33]. Consequently, information needs to be extracted from the model revealing SNP interactions, avoiding black box predictive models [11].

These difficulties make higher-order epistasis detection a challenging problem that cannot be effectively solved by exhaustive search methods. Several machine learning and artificial intelligence methods were proposed to overcome these challenges and better identify SNP interactions. The state-of-the-art methods are further discussed in this chapter.

2.6 State-of-the-Art Approaches: Machine Learning

The computational challenges in epistasis detection introduced by exhaustive search methods made the approach not computationally feasible to be applied in high dimensional datasets. The computational explosion caused by analysing all possible interactions in high dimensional datasets cannot be computed. To overcome this drawback, other non-exhaustive methods started to emerge. In this section, ML and Artificial Intelligence (AI) methods are introduced.

2.6.1 Machine Learning

ML techniques are non-exhaustive methods that learn through experience. These methods are more advanced than traditional statistical models and can mathematically map SNP interactions and relate them to the expression of complex disease phenotypes [34]. Hence, these algorithms can overcome some of the challenges introduced by the high dimensionality in the epistasis detection problem.

These algorithms can be divided into supervised and unsupervised methods. Supervised ML algorithms use labelled data to train the model, meaning that each sample must be assigned to a certain class (case and control in the case of epistasis). On the other hand, unsupervised machine learning methods use unlabelled data and try to learn useful properties of the data.

A simple diagram on how to build ML models is presented in Figure 2.3. The first stage consists of a data preprocessing stage, where the features to be learned by the model are selected. Further, several models are trained and the ones showing the best predictive performance are selected for the last stage. In the validation stage, models are tested using unseen data to evaluate if the training stage was performed correctly. A more detailed explanation of these stages is provided in Chapter 2.7.

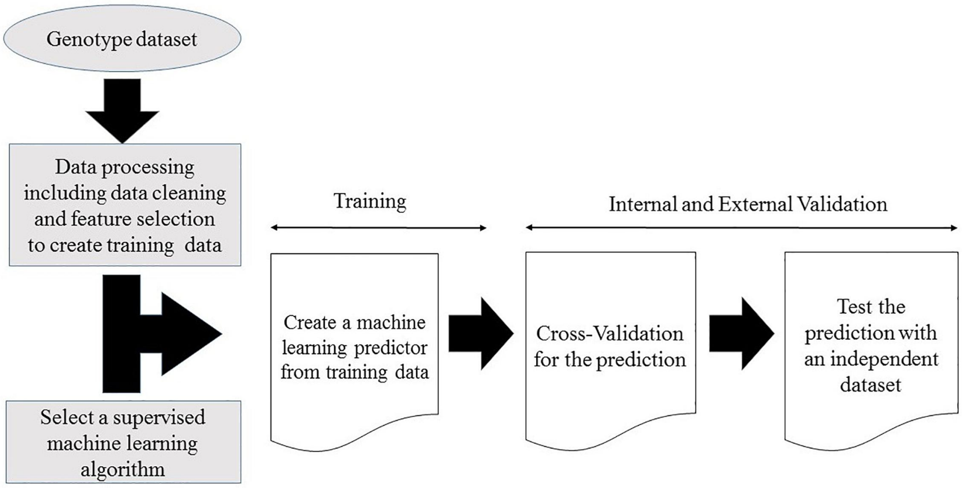

Genotype dataset
Data processing including data cleaning and feature selection to create training data
Select a supervised machine learning algorithm
Training
Internal and External Validation
Create a machine learning predictor from training data
Cross-Validation for the prediction
Test the prediction with an independent dataset

Figure 2.3: Workflow for creating a supervised ML learning model [4].

These methods have been gaining a lot of popularity since the computational burden caused by exhaustive methods could be avoided. Therefore, a wide variety of methods have been published using several ML algorithms, including several review surveys [7–9, 34–36] comparing the different methods and their highlights and drawbacks. For simplicity reasons, in this section, only the Random Forest (RF) and Support Vector Machines (SVM) are considered.

Random Forest (RF)

Random forest (RF) [37] is a collection of classification or regression trees (CART) that are grown from a preselected bootstrap dataset. A bootstrap dataset is created from the original dataset, leaving one-third of the samples as out-of-bag (OOB), which are later used to estimate the prediction error. A tree is a classifier that finds the SNP set that better predicts a certain phenotype. The final prediction is given by the majority of votes provided by all of the trees in the forest, using the OOB samples. One of the main advantages of the RF method is that, for each node (SNP), variable importance can be calculated. Hence, the model decisions can be interpreted and the epistasis interactions discovered.

Jang et al. [38] used random forest with Gini importance to pre-select a set of candidate SNPs and further investigate up to three-order epistasis interactions. Schwarz et al. [39] proposed random jungle (RJ) with permutation importance to allow the random forest algorithm to be applied in large-scale SNP data. This approach aims to make RF implementation less computationally demanding by using multicomputer and multithreading techniques. Since variable importance can be calculated in RF, these methods started to be applied as filters in two-stage approaches to select possible interacting SNPs [40, 41] and perform further exhaustive analysis to detect interactions [40]. To conclude, variants of the RF algorithm combining several SNPs in one node have been implemented [42] which, despite being very computationally demanding, were still able to overcome exhaustive search methods [9].

Random forest methods and their variants have been applied in several areas of genomic data analysis [43]. For epistasis detection these methods have been successfully implemented into real genomic datasets such as Chron's disease [39], asthma [40], rheumatoid arthritis [42], Prostate cancer [41], Bladder cancer [44] and Familial combined hyperlipidemia (FCH) [45].

Support Vector Machines (SVM)

SVM is a machine learning algorithm often used in classification and regression tasks. This algorithm is a probabilistic binary linear classifier, able to find a hyperplane that separates points into two distinct classes. The training data consists of a set of feature vectors labelled as positive or negative. The model is trained to find the hyperplane able to separate data into two distinct classifications. Once trained, the model can classify new samples into two distinct categories [8]. If the data is difficult to separate, a kernel is applied to map the data into a high dimensional space able to separate it. Figure 2.4 shows an example of a kernel application that maps the data into a feature space where it can be linearly separated. According to [35], kernels can be linear, polynomial and radial basis functions.

In the context of epistasis detection, feature vectors are a pair of SNPs. Thus, training data is formed by positive labelled SNP pairs, meaning that interactions exist between those two SNPs and by negative labelled data, meaning that no interactions exist between those SNPs. These features are mapped into a high dimensional space (Kernel), where a hyperplane can divide SNP pairs into interacting and non-interacting. Polynomial and or radial basis kernels are used [35]. However, these models cannot handle

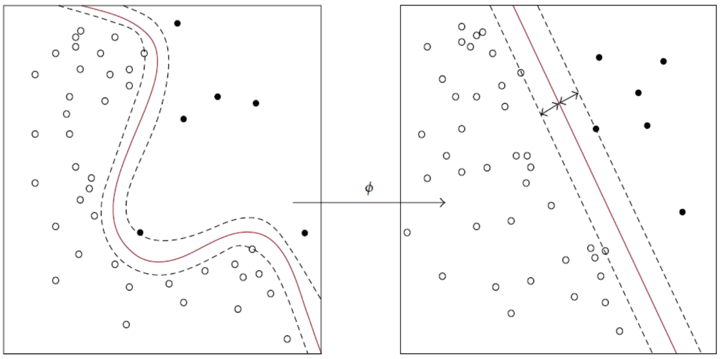


Figure 2.4: Kernel mapping features into a linearly separable space [35].

large SNP datasets [9].

Chen et al. [46] developed four different approaches combining support vector machines with combinatorial optimization techniques (local search and genetic algorithms) and feature ranking criteria. When compared with MDR, the authors concluded that the proposed methods achieved better results when handling unbalanced datasets and provided a more stable and less susceptible to overfitting model. A two-stage design was proposed in [47]. In the first stage, SVM with L_1 penalty acts as a filter by selecting the most promising SNP interactions. In the second stage, a logic regression method is applied to evaluate the previously selected interactions and remove non-significant candidates. Marvel et al. [48] proposed a method combining grammatical evolution (GE) and SVM. GE is a method similar to genetic programming, able to optimize features (SNPs) and potential architectures (e.g. kernel initial hyperparameter/s value). Models are converted into binary strings and compete with each other, surviving the one with better accuracy values. In [49], a data-driven guided random algorithm BMSF (binary matrix shuffling filter) is combined with SVM. This filter is able to select a relatively small number of SNPs and accurately classify them. This method allows SVM to be applied in datasets with a higher number of SNPs.

SVM models and their variations have been used to discover epistatic interactions in real datasets such as: Prostate cancer [46], Parkinson disease [47], Human cancer [49] and T2D [50].

2.6.2 Swarm Intelligence Search Algorithms

SIS algorithms are decentralized, self-organized systems that often mimic the collective behaviour of swarms avoiding exhaustive search in initial datasets. These non-exhaustive and efficient algorithms have been applied in high-order epistasis detection [21].

SIS algorithms are global optimization methods as they can escape local optimum solutions and do not depend on the initial population [21]. These algorithms consist of a population of simple agents that interact with each other and explore the environment. The communication between the different agents allows the algorithm to discover the optimal solution faster. Furthermore, since there is no restriction on the optimization function to be used, any function capable of evaluating epistasis interactions can be employed.

Ant Colony Optimization (ACO)

Among the SIS methods, ant colony optimization (ACO) has been receiving a lot of attention due to their heuristic positive feedback search and high detection power. ACO was inspired by the behaviour of ants when searching for the optimal path for the colony to reach a food source. In [20], a detailed review of twenty-five ACO methods is provided, analysing their strengths and limitations.

Wang et al. [51, 52] proposed a two-stage design that combined ant colony algorithms with an exhaustive search called AntEpiSeeker. In the first stage, a generic ACO is used for finding highly suspected SNPs. Once filtered, the second stage applies an exhaustive search to detect epistatic interactions. Other methods combining filtering approaches with ACO algorithms were proposed in an attempt to reduce the initial search space [53].

Also, to improve ACO methods performance, methods combining multiple objective functions were proposed. MACOED [54] is a multi-objective (logistic regression and Bayesian network methods) two-stage design for detecting genetic interactions. Multi-objective methods capture more information about iterations than single-objective methods, allowing more accurate analysis of SNP interactions.

Moreover, Zhou et al. [55] incorporated heuristic information not only to the path selection strategies and for the pheromone updating rules. The incorporation of heuristic information makes the search less random and more accurate, however, one major drawback is the computational cost.

Due to the great amount of published ACO approaches, only the most relevant ones were cited in this Section. Detailed analysis and comparison between the different ACO methods are provided in [20, 21].

2.6.3 Highlights and Further Improvements

ML and AI approaches were proposed to overcome the limitations established by the exhaustive search methods. These non-exhaustive algorithms have been proposed as powerful approaches for the detection of high order gene interactions.

SVM revealed to be powerful methods able to handle high dimensional data. These methods are not pruned to *overfitting* and were able to handle unbalanced datasets. However, one major drawback of these methods is that they are restricted to pairwise interactions and can not be directly used as a filter (they need to be merged with other methods).

RF are promising algorithms able to capture variable importance and rank SNPs according to their relevance in predicting the phenotype. The development of random jungle made it possible to apply random forest methods at a genome-wide scale. Still, RF has some limitations since SNPs without marginal effects might not be detected by these algorithms.

Additionally, ACO algorithms have been extensively explored in the past years [20]. Two-stage designs and multi-objective optimization strategies were proposed. The simplicity and effectiveness in exploring and exploiting high dimensional search spaces have made these methods very popular in recent years. Consequently, this approach has been over-explored in the past years with new released approaches just representing small improvements over previous ones.

Most of the state-of-the-art works represent ML and AI approaches, hence, there is a wide variety of methods and only some of them are discussed in this Section. Several review papers discussing in more detail all the different methods, as well as, their advantages and disadvantages, have been published [7–9, 20, 21, 34–36]. These are promising approaches since can handle high dimensional datasets and avoid the computational explosion caused by exhaustive search methods.

2.7 Deep Learning

Deep learning models can learn relevant internal features of high dimensional and complex datasets. These models have improved the state-of-the-art methods in areas related to speech recognition, image recognition, and genomics [56]. The high dimensionality of genomics data makes the datasets too complex to be analysed by traditional statistical approaches, hence the use of deep learning is an advantage.

Since the first applications of deep learning in genomics in 2015, where deep convolutional networks were applied to DNA sequences [57, 58], the number of methods involving deep learning in genomics has largely increased [10]. The application of several types of deep learning networks (fully connected, convolutional, recurrent, and graph convolutional) in genomics have been reviewed in [10].

In this section, an overview of deep learning methods applied in epistasis detection is provided. First, the initial methodologies using neural networks are presented. Further, a theoretical introduction on MLP and CNN is presented since these are the two most relevant architectures in the state-of-the-art approaches. Also, variable and hyperparameter optimization strategies are also discussed. Moreover, hybrid and model interpretation approaches using deep learning architectures are presented. To conclude, a final balance of advantages, disadvantages, and further improvements of deep learning and neural network methods is addressed.

2.7.1 Neural Networks (NN)

NNs were inspired by the neurons behaviour and their ability to solve complex problems. These methods are universal function approximates able to capture patterns in the data without making any assumption on the output. Due to all these features, NN methods have been applied to genetic studies for capturing gene-gene interactions and make accurate predictions on the phenotype [59]. A detailed view of how NN architectures are built is provided in Section 2.7.2 (Multilayer perceptrons are NNs with extra hidden layers, however, their methodology is the same).

The first approaches using NN architectures focused mainly on improving the accuracy values of the networks through hyperparameter optimization [35, 59]. Due to the number of parameters and the variety of values that can be assigned to each of them, approaches avoiding exhaustively search for the best architecture started to emerge. The main goal of these approaches was to produce a network with the highest accuracy possible. Thus, accuracy is used as a performance measure to evaluate if the interactions were being captured.

One of the first approaches to solve this parameter optimization problem was a genetic programming neural network (GPNN) [60], which uses genetic programming to find the network configuration that best fits the input data. Ritchie et al. [60] compared the performance of a regular neural network and the new genetic programming neural network. GPNN was used to optimize the variables, connectivity, and weights of the network. The results showed that GPNN outperformed BPNN, having better prediction power. Further applications of GPNN had been used to detect gene-gene interactions in real datasets such as in Parkinson's disease [61], and Alzheimer's disease, colorectal disease, breast's disease, and prostate's disease [62].

In order to handle higher dimensional epidemiological data, grammatical evolution neural networks (GENN) [59] were introduced. Grammatical Evolution (GE) is a variation of a genetic algorithm that uses a set of rules for translating NNs into an array of bits. This set of rules is called grammar. In [59], both GENN and GPNN were applied to a real HIV immunogenomics dataset. The results demonstrated that GENN is able to optimize NNs in fewer generations and capture SNP interactions in the presence of noise. However, this model could only identify strong pair-wise interactions.

Later, Hardison and Motsinger-Reif et al. [63] bounded GENN with quantitative traits analysis together, creating QTGENN. The grammar is modified allowing the NN output to be more than a binary value. Despite having a more diverse grammar, this model requires high computational power.

Tomita et al. [64] used a neural network with a parameter decreasing method to predict SNP associations related to a childhood allergic asthma dataset. By removing input parameters (SNPs) from the network and measuring their impact on the network accuracy, the original 25 SNPs were reduced to 10 SNPs revealing epistatic interactions between them. Despite achieving good prediction values, this method only works for relatively small datasets.

The described approaches using NNs all focussed on improving network accuracy through hyper-parameter optimization. Due to the great number of parameters to be optimized, genetic programing algorithms and other adaptations were developed. These new approaches managed to find accurate networks in both simulated and real datasets. Still, the only performance measure being used to evaluate if networks are detecting interactions is accuracy. From these simple networks, more complex and deeper models started to emerge. In the following Section 2.7.2, deep learning methods are presented and explained.

2.7.2 Main Architectures

Despite the great variety of architectures available, certain concepts are common in all DL architectures. The generic DL architecture is obtained by combining several layers of neurons. Neurons are the basic unit of a DL architecture, which were proposed in the 1950s with Rosenblatt ”perceptron” [65].

The neurons in each layer receive the output of the neurons in the previous layer as input. The importance or strength of the connection is represented by a constant value called weight. The higher the weight the stronger the connection, with zero weight meaning that a neuron does not influence the next layer. The output of each neuron is defined by a non-linear transformation called activation function and a constant value named bias. Hence, the output of a neuron is given by:

$$y = f(W_i x_i + b) \tag{2.5}$$

where x_i represents the input values, $f()$ a non-linear activation function, b the bias and W_i the connection weight of each input value.

The activation function is a non-linear function that transforms the linear input of a neuron into its output. Choosing the right activation function is an important step in optimizing the performance of DL algorithms. In Section 2.7.3, a more detailed explanation on how to choose the activation function is provided. The most common activation functions and the corresponding mathematical expressions are represented in Table 2.3.

Table 2.3: Activation functions in neural networks.

Name	Function $f(x)$
Tanh	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$
Relu	$\max(0, x)$
Elu	$\begin{cases} \alpha(e^x - 1), & \text{if } x \leq 0 \\ x, & \text{if } x > 0 \end{cases}$
Softplus	$\ln(1 + e^x)$
Linear	x
Sigmoid	$\frac{1}{1 + e^{-x}}$

When DL is applied to capturing gene-gene interactions, each input node corresponds to an SNP locus, with the corresponding genotype coded as 0 (homozygous major), 1 (heterozygous) or 2 (homozygous minor). Each sample is a set of genotype values and the observed disease status, with 1 representing case and 0 control.

The model is trained to capture interactions between the input SNPs and make accurate predictions on the phenotype. Therefore, DL algorithms are binary classifiers whose output must be mapped into the interval $[0, 1]$ using the sigmoid function [10]. This way, the activation function of the output layer is always the sigmoid function.

Multilayer perceptrons (MLP)

MLP also called fully connected feed-forward networks, are one of the most popular methods in deep learning. It consists of fully connected layers named input, hidden and output layers. In Figure 2.5 an architecture of an MLP with two hidden layers is presented.

The first layer of an MLP model is the input layer. The output of the input layer given by the weighted nonlinear activation function of each input added to a ”bias” (constant value). Thus, the first output layer is given by:

$$z^{(1)} = b_0 + W^0 f^{(0)}(x) \tag{2.6}$$

where x represents the input genotypes of each individual (sample), b the ”bias”, W^0 the weights of the input layer and $f()$ a nonlinear activation function. In the following hidden layers, the same formula in Equation 2.6 is used, however, instead of x, the outputs from previous layers are used:

$$z^{(k)} = b_{(k-1)} + W^{(k-1)} f^{(k-1)}(z^{(k-1)}) \tag{2.7}$$

where $z^{(k-1)}$ represents the output from the previous layer. The output layer gives a final classification, which in the context of epistasis is case or control.

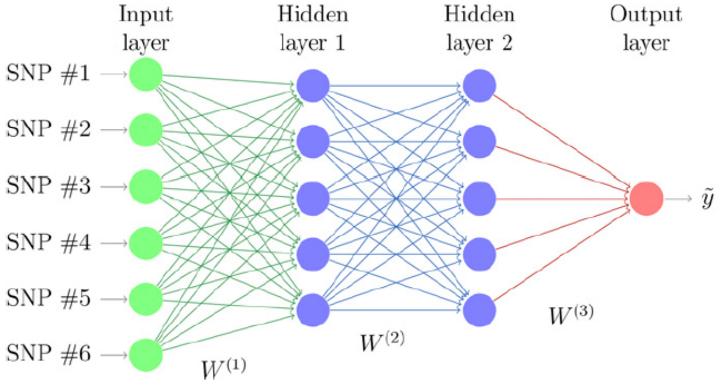


Figure 2.5: Multilayer Perceptron architecture representation. Each hidden layer is fully connected to the previous one. The weight matrices are represented by $W^{(i)}$ [66].

Uppo et al. [67–69] 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 breast cancer dataset [70]. 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. [71] 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. [66] 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 DL networks have great potential when dealing with genomic prediction. Still, further investigation must be performed for DL to overcome current linear models.

Convolutional neural networks (CNNs)

Convolutional neural networks (CNNs) is a variant of MLPs proposed to solve complex problems such as text, image and speech recognition [56]. It consists of dense, fully connected layers and convolutional layers. In Figure 2.6 the architecture and methodologies used in CNNs is represented.

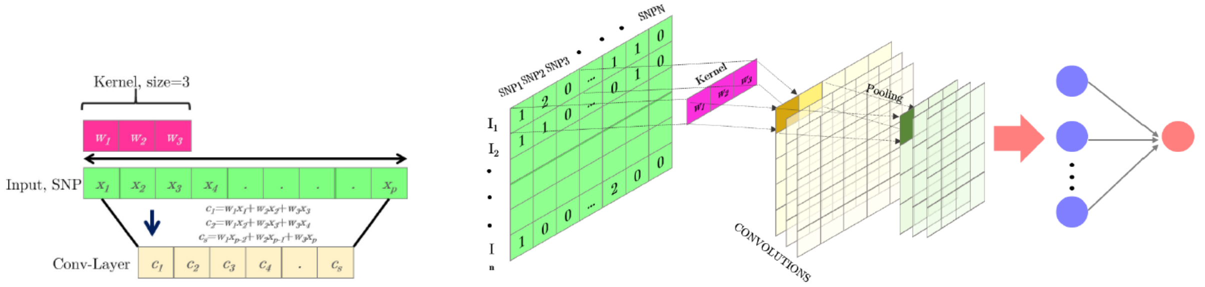
Unlike regular neural networks, convolutional neural networks are formed by convolutional layers, pooling layers, fully connected layers and normalization layers. In each convolutional layer, a ”kernel” or ”filter” is applied to extract features from the data, followed by a non-linear activation function to produce its output. A filter can be defined as a combination of input values where the weights are the same across the same input sample. To define a filter it is necessary to define its dimension, width and stride (Figure 2.6). Further, a pooling layer merges the filter outputs by taking the mean, maximum or minimum of those values. Finally, the last layer is a fully connected layer which gives the final classification based on the previously extracted high-level features.

When applied to epistasis detection, genomic data can be treated as a signal or sequence of genotypes (0, 1 or 2). Thus, a one-dimensional kernel can be used to extract features from the data [66].

Bellot et al. [66] 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. [72], performed a similar analysis to [67–69] 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. [73] 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.


Kernel, size=3
w_1 w_2 w_3
Input, SNP
x_1 x_2 x_3 x_4 . . . . x_p
$c_1 = w_1x_1 + w_2x_2 + w_3x_3$
$c_2 = w_1x_2 + w_2x_3 + w_3x_4$
$c_s = w_1x_{p-2} + w_2x_{p-1} + w_3x_p$
Conv-Layer
c_1 c_2 c_3 c_4 . c_s
SNP1 SNP2 SNP3 . . . SNPN
I_1
I_2
I_n
Kernel
Pooling
CONVOLUTIONS

(a) One-dimension (1D) filter convolution

(b) Full representation of a 1D convolutional neural network for a SNP matrix

Figure 2.6: Convolutional Neural Network architecture representation. Convolution outputs are represented in yellow. After, the pooling layer combines the output from the previous layer into a single neuron representation, represented in green. A fully connected layer gives the final classification [11].

2.7.3 Training and Overfitting Issues

To perform the training of a DL algorithm, N-fold cross-validation is a commonly used algorithm for splitting data into training and testing. In the first iteration, the dataset is split into n subsets, with one split used for testing and the remaining n-1 splits used for training. This process is repeated until all the dataset is covered. Hence, in the case of n-fold cross-validation, the algorithm runs n times. The chosen model is the one that shows the highest cross-validation consistency, meaning the model that on average showed the best performance across the entire dataset.

The training of a neural network consists of adapting the weights and biases of each neuron to minimize a loss function. A loss function quantifies the difference between the observed and the predicted data [10]. Since in the context of epistasis detection the DL models are binary classifiers, binary cross-entropy is used as a loss function. Thus, the expression of the loss function is given by:

$$Loss = -\frac{1}{N}\sum_{N}^{i=1} y_i \log\left(p(y_i)\right) + (1 - y_i)log(1 - p(y_i)) \tag{2.8}$$

where y is the label (1 for case and 0 for control) and $p(y)$ is the predicted probability of a sample being a case across all the N samples.

The first step in training a network consists of randomly initializing all the weights and biases. Further, these are iteratively changed using gradient-based optimization function named ADAM [74]. This method takes a small random subset (batch) of the training data and uses it to make small changes in the network parameters.

Besides ensuring the model is able to make accurate predictions on the training data, it is also necessary to ensure that it does not overfit. Overfitting is a scenario in which the model fits the training data, however, does not generalize its predictions to unseen data. To avoid this scenario, early stopping and dropout are used.

Early stopping is a simple, yet very effective method, which stops training once the network starts overfitting, allowing the network to train on the correct number of epochs. Additionally, dropout consists of setting to zero the output of randomly selected neurons. The percentage of inactivated neurons is referred to as the dropout rate.

The use of these methods consists in good practice when training a deep learning model, ensuring that the model is flexible to unseen data and makes accurate predictions [10, 11, 56, 66].

2.7.4 Variable Selection and Hyperparameter Optimization

GWAS involve complex high dimensional datasets, therefore it is not computationally feasible to feed them directly into a deep neural network due to the enormous number of parameters to be estimated. Variable selection or statistical filtering must be used to reduce the high number of SNPs present in the original data.

All of the methods described in this section [67–69, 71, 75–79] use logistic regression [80] for SNP selection based on their *p-value*. This method enables the number of SNPs with significant marginal effects to be reduced. The filtered SNPs have strong linear associations to the phenotype. Despite, being a simple and very used method, logistic regression only individually captures the effects of each SNP with the phenotype meaning that epistatic interaction between SNPs might be missed.

Another main challenge when developing deep learning optimization is choosing the right hyperparameter configuration. This is a fundamental step when implementing deep learning models as it has a great impact on the model performance.

Grid search is an approach that exhaustively builds models based on hyperparameters previously specified in the grid. The grid begins with large steps between parameter values, making them finer when approaching the best configuration. However, this approach is only computationally feasible with a small number of parameters as the number of values to be tested exponentially grows with the increase of the number of hyperparameters [79]. A more efficient and used approach is random search [81] which finds better models by effectively searching a larger configuration space.

Another less used but worth mentioning approach is the adaptation of genetic algorithms to hyperparameter optimization. In [66], an adaptation of a genetic algorithm called DeepEvolve [82] was used to evolve populations of MLPs and CNNs with the objective to achieve the best hyperparameter configuration. This type of approach has shown great success in improving neural networks architecture and is a worth exploring idea [62].

2.7.5 Epistasis Detection using Deep Learning

As described in previous chapters, DL algorithms have shown great potential when dealing with genomic datasets. During training, these algorithms can capture SNP interactions and, once the training is finished, make predictions on the phenotype. Despite the advantages of these algorithms, these models remain as black-box predictive models, meaning that they are good at making accurate predictions but not in explaining them.

The authors from the previously identified methods using MLPs [66–69] and CNNs [66, 73] declare that, since the network was able to make accurate predictions on the phenotype, it was also able to capture the interacting SNPs. Thus, accuracy is being used as a performance measure to identify if the network is capturing or not epistatic interactions.

To overcome this black-box limitation of DL models, other variants combining the predictive performance of DL with other algorithms started to emerge.

Hybrid Approaches

Hybrid methods use the ability of deep learning to extract non-linearities in the data together with other machine learning or unsupervised learning methods.

Uppu et al. [79] merged a DL network with RF for detecting multi-locus interaction between SNPs. In this approach, the hyperparameters of the DNN are trained to learn the data representation, then fed into a random forest for detecting SNP interactions. RF constructs several random trees, which are formed by decision nodes (SNPs) and leaf nodes (case-control classification). Each decision node has a binary decision function that decides the path (right or left, which represent different genotypes) based on the parameters received from the DNN. Still, this method was only applied in simulated datasets for the detection of two-order epistatic interactions.

These approaches are promising as they merge the predictive power of machine learning methodologies with the ability of deep feature extracting in DL.

Model Interpretation Approaches

Model interpretation algorithms are used to overcome the lack of transparency of DL models. These algorithms can interpret neural network predictions and evaluate the contribution of each input to the prediction. These techniques of interpretation are becoming promising tools in image recognition and biology, as they can extract new information from data and give new insights to complex systems [83].

Waldmann et al. [77] trained an Approximate Bayesian Neural Network (ABNN) to make predictions on a pig dataset. Once trained, SNP importance is detected by weight inspection. Besides concluding that ABNNs are a suitable method in phenotype prediction, the ability to extract information from the model weights is also highlighted. Still, this method of weight inspection only works on rather simple networks.

Additionally, in [10] feature importance scores are referred to as possible approaches for model interpretation in genomic studies. Feature importance scores can be computed using input perturbations or using backpropagation algorithms. Input perturbations consist of adding noise to the input and evaluate the impact on the output, whereas backpropagation algorithms backpropagate through the network and assign relevance scores to the inputs.

The epistasis detection problem requires the evaluation of multiple input SNPs. Consequently, adding noise to every input SNP and evaluate the impact on the output can be very computationally demanding. Thus, backpropagation interpretability algorithms are a promising approach to evaluate SNP importance and detect epistasis. Still, no evidence was found that these methods had been used for epistasis detection.

2.7.6 Highlights and Further Improvements

DL is an emerging field and has a wide variety of algorithms that recently have been applied in epistasis detection. By making accurate predictions on the phenotype, these models proved to be able to capture epistatic interactions. This deep feature extraction ability is what makes these methods promising in detecting gene-gene interactions. A table with all the cited methods in this chapter is provided in Table 2.4.

Despite the great advantages, the lack of interpretability remains one of the major drawbacks of these approaches. As observed in neural networks, deep learning models remain as black-box predictive models, meaning that it is hard to interpret the relevance of each input when predicting the output. Consequently, accuracy has been used as a performance measure on the ability of the network to capture epistatic interactions. Variants combining DL with other machine learning or unsupervised learning techniques have been developed [79] in an attempt to overcome this issue.

Additionally, the use of model interpretation algorithms consists of a promising approach. These algorithms can measure the relevance of each input when making predictions on the output. Despite showing great potential in image recognition and other DL tasks, no evidence was found that these methods have been applied in epistasis detection.

By overcoming DL black-box nature using model interpretability methods, epistatic interactions can

Table 2.4: Deep Learning models from state-of-the-art papers.

Study	DL Architecture	Summary
Uppo et al. [67–69]	MLP	Exhaustively searches for SNP combinations and trains an MLP to detect the interacting SNPs.
Montaez et al. [71]	MLP	Preselects a set of possible interacting SNPs in an obesity dataset and trains an MLP to validate the selection.
Bellot et al. [66]	MLP and CNN	Compares the predictive performance of MLPs, CNNs and Bayesian linear models in genomic datasets.
Salesi et al. [73]	CNN	Compares several filtering methods and trains a CNN using the previously selected SNPs.
Uppo et al. [72]	CNN	Exhaustively searches for SNP combinations and trains a CNN to detect the interacting SNPs.
Uppu et al. [79]	MLP + RF	Merged an MLP architecture with random forest in an attempt to interpret the network results.
Waldmann et al. [77]	ABNN	Trained and Approximate Bayesian Network in a pig dataset and evaluated SNP importance by weight inspection.

be detected on a previously trained network by evaluating the relevance of the input SNPs. Still, no evidence of the use of these algorithms was found in epistasis detection. Thus, as suggested in [10], further investigation of these methods is a promising approach.

To conclude, DL algorithms show great potential despite their black-box nature. All the state-of-the-art methods use accuracy to confirm the presence of interacting SNPs, but not to detect the interactions themselves by interpreting the information extracted by the network. To overcome this issue, further investigation on model interpretation methods applied to epistasis detection must be performed. This way the predictive performance of deep learning models can be used to assign relevance scores to each input SNP and detect interactions.

2.8 Summary

This chapter started with a brief description of gene expression and a general view of how the human genome is organised. Further, SNPs were introduced as the most common variant in the human genome, being important in the prediction of certain complex human diseases. Further, GWAS were also introduced and a definition of SNP-SNP interactions was provided. A section explaining how epistasis detection is mapped into computers was also presented.

Exhaustive search methods were introduced as one of the first approaches to detect epistatic interactions. These methods exhaustively search for all possible combinations to find interactions that can be associated with a phenotype. However, when scaled to higher-order epistasis detection these methods turn into combinatorial overloads which cannot be handled by current computers. Further, challenges in epistasis detection introduced by the limitations of the exhaustive search methods were discussed.

To overcome these challenges, machine learning and artificial intelligence methods were proposed.

These non-exhaustive search methods can be used to discover higher-order epistasis detection since they do not exhaustively search for all SNP combinations. RF and SVMs were presented as two of the most used approaches. The predictive power of these algorithms was revealed to be promising. Still, the ability to only detect pairwise interactions and only in the presence of marginal limits the scenarios in which these methods can be applied. Moreover, SIS algorithms are introduced with a more detailed view of ACO algorithms. These methods revealed great exploration capabilities and are referenced as possible filtering approaches. Still, an objective function able to accurately classify interacting SNPs is yet to be discovered.

Finally, a detailed view of deep learning methods was provided. These methods can extract deep features from the data and capture interacting SNPs. DL models have proven to be capable of learning features from the data and make accurate predictions on the phenotype. A review on MLP and CNN architectures was performed and issues regarding the training and optimization of networks were discussed. It is concluded that the current approaches still treat DL networks as black-box predictive models. The only performance measure used to evaluate if networks have captured interactions is accuracy and the information extracted by the network is never interpreted. Despite having been applied in image recognition tasks, no evidence was found that model interpretability algorithms have been used to detect epistasis interactions. Thus, there is a gap in the literature related to network interpretability that must be explored. This way, networks can be evaluated on their ability to detect relevant SNPs and not on making accurate predictions on the phenotype.

Chapter 3

Methodology: Interpretability of Deep Learning Models

An analysis of the literature on the different methods applied in epistasis detection revealed the great potential of DL networks. Recent studies applying these algorithms proved their ability to extract interactions from the input SNPs and make accurate predictions on the phenotype. However, all the fount approaches using DL networks still treat them as black-box predictive models, only evaluating networks on their ability to make accurate predictions on the phenotype. This way, information present in the networks is never interpreted and SNP interactions are never detected. Therefore, there is a gap regarding network interpretability that must be filled.

In this dissertation, a methodology for interpreting DL models and detecting the interacting SNPs is presented. Sensitivity analysis is introduced as a promising approach to evaluate networks individual classifications and assign a relevance score to each input SNP. This type of analysis is performed across the entire dataset to identify the interacting SNPs in a population. This way, network limitations in terms of interpretability can be evaluated, which is one of the main goals of this dissertation.

A new interpretability performance measure is defined to allow networks to be classified not only on accuracy but also on their ability to detect the interacting SNPs. High interpretability means that a network was able to capture all interacting SNPs. To allow the evaluation of networks under a different variety of epistasis scenarios, several datasets need to be created. The datasets also need to follow different models having several heritabilities and minor allele frequencies were created and the performance of the networks was evaluated.

3.1 Datasets

To understand the use of DL in epistasis detection, it is necessary to understand the used datasets and how these were created. For generating the datasets, the GAMETES [23] generator and the Toxo [24] library were used.

Two separate types of datasets were generated, ones displaying marginal effects, named marginal effects datasets (ME), and others with the absence of marginal effects, named non-marginal effects datasets (NME). In this section, both dataset types are explained, as well as, the parameters needed to generate them.

3.1.1 Parameters

In both GAMETES [23] and Toxo [24], models are generated using penetrance tables. To generate them it is necessary to define MAF and heritability (h^2). As explained in Section 2.3.2, MAF represents the frequency of the less common allele in a population and heritability the portion of phenotype differences due to genetic variants in a population.

To evaluate the performance of DL models in different scenarios, datasets with several values of MAF for the epistatic SNPs are generated. The values used for MAF were [0.05, 0.1, 0.2, 0.3, 0.4], similar to the ones used in [54] but with more intervals. For each value of MAF, datasets with heritability [0.01, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4] were generated.

Since penetrance values must be in the interval of $[0, 1]$, some combinations of MAF and heritability do not produce a valid table. Despite having different methods for generating tables, both GAMETES and Toxo generators show this limitation.

3.1.2 Non-Marginal Effects Datasets (NME)

A dataset is considered not to have any marginal effects if, between the interacting SNPs, none of them interacts individually with the phenotype.

NME datasets are created using GAMETES [23]. This generator can generate pure and strict datasets, with pure referring to datasets without any marginal effects and strict referring to epistasis where n SNPs are associated with the phenotype but no subset of them are. A table representing the generated datasets is presented in Table 3.1, with **YES** and NO meaning if GAMETES could generate or not the dataset respectively.

Table 3.1: Generated Pairwise NME Datasets.

		h^2						
	Values	0.01	0.05	0.1	0.15	0.2	0.3	0.4
MAF	0.05	**YES**	**YES**	**YES**	NO	NO	NO	NO
	0.1	**YES**	**YES**	**YES**	**YES**	NO	NO	NO
	0.2	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**
	0.3	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**
	0.4	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**

As observed in Table 3.1, for lower MAF values (0.05 and 0.1), GAMETES cannot generate datasets for higher heritability values (0.2, 0.3 and 0.4). With the increase in MAF, datasets with higher heritability values start to be generated. Additionally, one major limitation of GAMETES when generating pure and strict datasets, is that only pairwise epistatic interactions can be simulated. Thus, in this Thesis the used NME datasets are all pairwise datasets.

3.1.3 Marginal Effects Datasets (ME)

ME datasets were created using the Toxo [24] generator. Toxo is a MATLAB library to calculate penetrance tables of any order. By specifying a heritability, MAF and epistasis model, Toxo can build a penetrance table according to the specified parameters. The built table is further used by existing software simulators, like GAMETES, to generate epistatic datasets.

Hence, the Toxo library provides a new method for generating penetrance tables that overcome some of the limitations imposed by GAMETES, such as the order of the marginal effect epistatic models. Moreover, the Toxo library allows the creation of tables with a wider variety of MAFs and heritability values. Thus, when generating ME datasets, all the combinations of MAF and heritability values could be generated.

Models

When generating datasets with marginal effects it is necessary to define an epistasis model. Models are mathematical expressions that map epistatic interactions and, consequently, determine the penetrance values.

The models used to generate pairwise ME datasets were the additive (Table 3.2), multiplicative (Table 3.3), xor (Table 3.4) and threshold (Table 3.5). These models are commonly used to test the performance of epistasis detection algorithms [25, 54].

Table 3.2: Additive model penetrance table.

Genotypes	AA	Aa	aa
BB	α	$\alpha(1+\theta)$	$\alpha(1+\theta)^2$
Bb	$\alpha(1+\theta)$	$\alpha(1+\theta)^2$	$\alpha(1+\theta)^3$
bb	$\alpha(1+\theta)^2$	$\alpha(1+\theta)^3$	$\alpha(1+\theta)^4$

Table 3.3: Multiplicative model penetrance table.

Genotypes	AA	Aa	aa
BB	α	α	α
Bb	α	$\alpha(1+\theta)^2$	$\alpha(1+\theta)^3$
bb	α	$\alpha(1+\theta)^3$	$\alpha(1+\theta)^4$

Table 3.4: Xor model penetrance table.

Genotypes	AA	Aa	aa
BB	α	$\alpha(1+\theta)$	α
Bb	$\alpha(1+\theta)$	α	$\alpha(1+\theta)$
bb	α	$\alpha(1+\theta)$	α

Table 3.5: Threshold model penetrance table.

Genotypes	AA	Aa	aa
BB	α	α	α
Bb	α	$\alpha(1+\theta)$	$\alpha(1+\theta)$
bb	α	$\alpha(1+\theta)$	$\alpha(1+\theta)$

High-Order

Since DL models do not make assumptions on the dataset, there is no need to tell the algorithms to search for a specific epistasis interaction order. Therefore, DL networks should be able to detect epistasis in a dataset regarding its order. To evaluate the limits of DL algorithms when detecting high order epistatic interactions, several high order ME datasets were generated.

For generating higher-order penetrance tables, the Toxo library was used with the same models, heritability and MAF values. Penetrance tables of order four were generated using the same additive, multiplicative, threshold and xor models. Similar to the pairwise ME models, all combinations of MAF and heritability could be generated.

The detection of the interacting SNPs in higher-order datasets is a more difficult task when compared with pairwise interactions, since more SNPs need to be detected. It is worth mentioning that DL algorithms do not make any assumption on the datasets, thus networks do not need to look for a specific interaction order.

3.2 Deep Learning Implementation

With the increasing popularity of DL methods, several Python libraries for implementing DL networks have been developed. Keras [84] and Tensorflow [85] are two of the most popular libraries and were used to implement both CNN and MLP models.

In this section, 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. True positive predictions and cross-validation are used to apply sensitivity analysis to the entire dataset and detect the interacting SNPs in the population.

By the end of this section, the methodology used to implement and interpret deep learning methods will have been completely introduced.

3.2.1 Architectures and Hyperparameter Optimization

From the literature review on the state-of-the-art methods, it is concluded that the two most used architectures are CNNs and MLPs. Thus, these are the chosen architectures to perform the study on interpretability analysis.

Hyperparameter selection is an important step when implementing deep learning models and directly affect the model performance [66]. To correctly define a search space for each CNN and MLP architecture, the most cited papers from the state-of-the-art works are considered [66–69, 73].

Grid search is applied to exhaustively test all architectures in the defined search space. This approach is very common in hyperparameter optimization, however, it is only valid on a small number of hyperparameters, otherwise, the number of architectures to be tested would explode. It is necessary to define an efficient search space to avoid the number of tested architectures growing exponentially.

When evaluating MLP hyperparameters the number of inputs and number of layers is considered. Varying the number of inputs affects the amount of noise in the input and makes the detection of interacting SNPs more complicated. More or less hidden layers will affect the number of nonlinearities used to learn from the data. Additionally, the number of neurons are also considered.

Regarding the CNN networks, more hyperparameters are considered. CNN networks can be divided into two different parts, a convolutional part where features are extracted and a classification part where the extracted features from the data are used for classification. Each part of the network has its hyperparameters to be optimized, thus, more hyperparameters must be considered in CNN networks and different strategies applied to avoid a combinatorial overload. A more detailed analysis of the used strategy is provided in Section 4.3.1. The kernel size controls the amount of SNPs used to extract information from the data at each interaction. When applied to epistasis detection, the use of smaller kernel sizes (size two or three) is advised [66]. Also, the number of convolutions is considered. This hyperparameter will define the non-linearities in the feature extraction stage. The classification part of CNN is a normal MLP network, thus, the hyperparameters to be optimized are the number of layers and the number of neurons, which have the same impact as in the MLPs.

3.2.2 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 [83]. Relevance values are further represented as heatmaps, which allows visual identification of the most relevant features.

In this Thesis, 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:

$$R_i(x) = \left(\frac{\partial f(x)}{\partial x_i}\right)^2 \tag{3.1}$$

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 workflow for explaining a prediction is presented in Figure 3.1. In this simple example, a patient with eight SNPs and having the disease is used. The first stage consists of using the trained deep learning model to predict the patients’ phenotype. If the prediction is correctly made, and the sample is classified as a case, the sensitivity analysis is performed to identify the SNPs with the highest relevance values. The heatmap presented in Figure 3.1 suggests the presence of a pairwise interaction between $SNP4$ and $SNP7$, since these show the highest 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
```

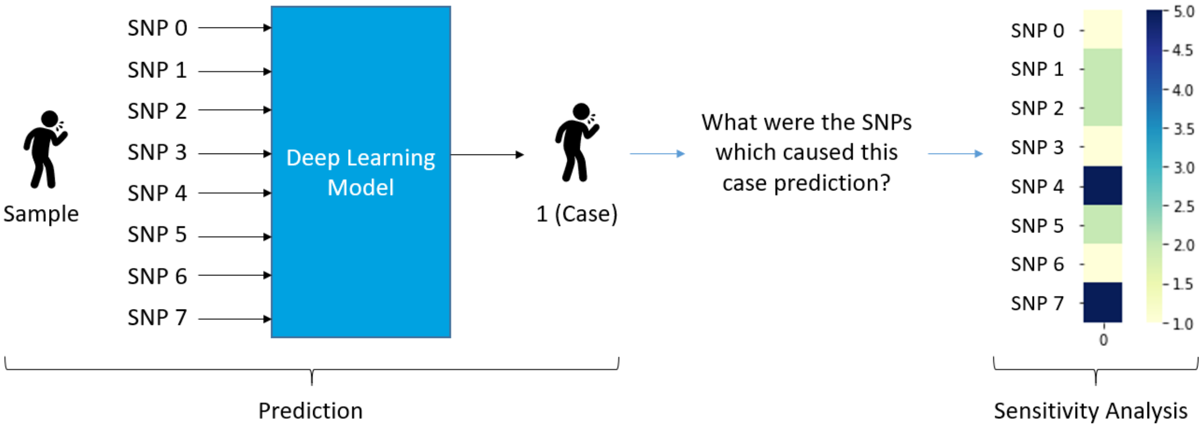


Figure 3.1: Workflow for calculating SNP relevance for one sample, using a sample with eight SNPs and having the disease. In the first stage, a prediction using a pre-trained model is made. If the sample is correctly predicted as a case, the second stage of sensitivity analysis is performed. The presented heatmap suggests the presence of a pairwise interaction between $SNP4$ and $SNP7$.

3.2.3 Epistasis Detection Algorithm

The method described in the previous Section 3.2.2 allows measuring the SNP importance for one sample. The main goal of epistasis detection consists in finding the interacting SNPs in a population, thus sensitivity analysis must be performed in more than one 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 (Figure 3.2). This is an advantage when compared to other DL 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 [83].

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 (Figure 3.2). According to the previous definition and Equation 3.1, SNP relevance can be defined as:

$$R_i(x) = \sum_{j=0}^{S} \left(\frac{\partial f(x)}{\partial x_{ij}} \right)^2 \tag{3.2}$$

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

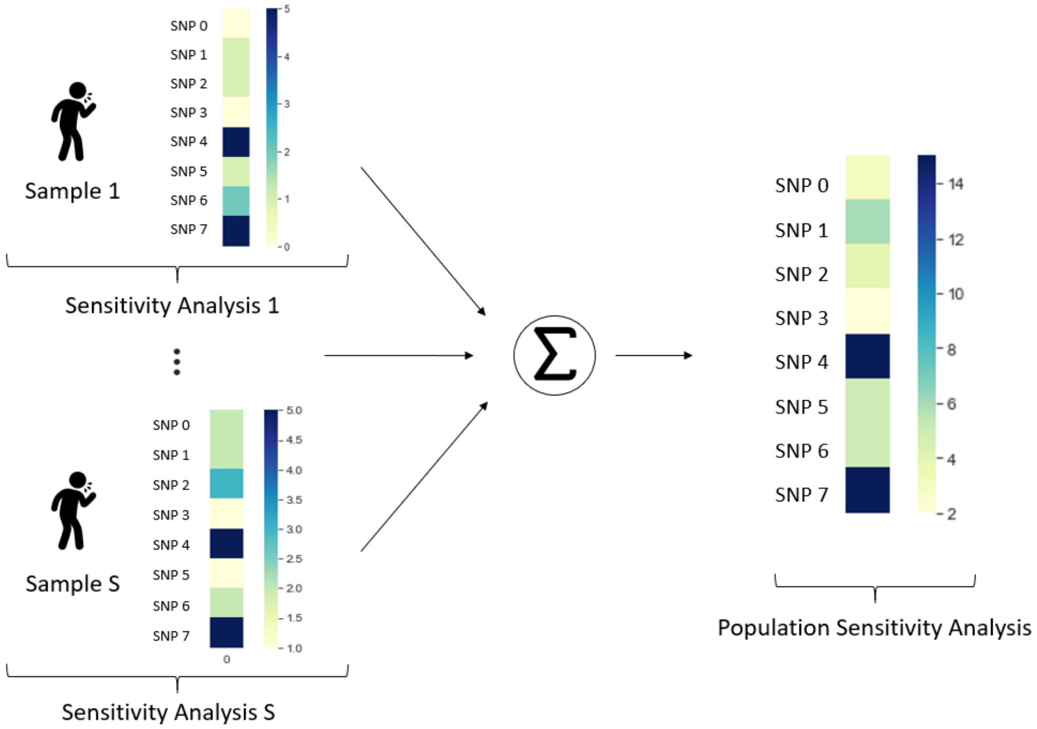


Figure 3.2: Workflow for calculating SNP relevance in a population. This value can be obtained by adding the individual sensitivity analysis of each sample.

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).

True positive samples are one of the values present in the confusion matrix. A confusion matrix is a table used to evaluate the performance of classification models. A more detailed explanation of the confusion matrix and all of its values, is presented in Section 4.1.2 where the used performance measures are discussed.

For now, it is only necessary to understand the concept of true positives. 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.

Additionally, as described in Section 2.7.3, 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 using Equation 3.2. Therefore, it is possible to obtain the interacting SNPs using:

$$R_i(x) = \sum_{k=0}^{C} \sum_{j=0}^{S} \left(\frac{\partial f(x)}{\partial x_{ji}} \right)^2 \tag{3.3}$$

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.

```
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 $R$.
```

3.3 Interpretability Metric

In previous works [66–69, 73], DL performance was only evaluated in terms of accuracy. A model having good accuracy values indicated that the model was capturing interactions and consequently making accurate predictions. However, in this Thesis, DL is evaluated in terms of interpretability, thus, a new performance measure was defined.

The main goal of the DL is to identify all the interacting SNPs as the most relevant across all the input SNPs. When evaluating the interpretability of DL on simulated datasets, the interacting SNPs are known. Knowing the interacting SNPs allows defining interpretability (I) as:

$$I = \frac{\min_{\forall k \in K}(p_k)}{N} \tag{3.4}$$

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.

For a clear understanding of the interpretability measure, an example is herein provided. Consider a pairwise interaction between $SNPA$ and $SNPB$ in a dataset with fifty input SNPs, where two models were trained to identify the interacting SNPs. Model one identifies $SNPA$ in position twenty and $SNPB$ in position thirty of the ordered relevance vector, whereas, model two identifies $SNPA$ in position forty-eight and $SNPB$ in position forty-nine in the relevance vector as well. Using Equation 3.4 interpretability values are calculated as follows:

$$I_1 = \frac{\min(p_{SNP_A}, p_{SNP_B})}{N} = \frac{\min(20, 30)}{50} = \frac{20}{50} = 0.4 \tag{3.5}$$

$$I_2 = \frac{\min(p_{SNP_A}, p_{SNP_B})}{N} = \frac{\min(48, 49)}{50} = \frac{48}{50} = 0.96 \tag{3.6}$$

Comparing the interpretability values between model one (Equation 3.5) and model two (Equation 3.6), it is possible to conclude that model two had better performance when compared to model one. The position of the interacting SNPs has a higher index on the ordered relevance vector, meaning that this model considered $SNPA$ and $SNPB$ as interacting. Consequently, model two has a higher interpretability value when compared to model one.

This type of analysis provides new insight into network interpretability related to epistasis detection. Model performance can be evaluated not only on accuracy but also on the interpretability of networks predictions, overcoming the black-box nature of DL models.

3.4 Summary

One of the main goals of this dissertation consists in evaluating DL algorithms for epistasis detection. To overcome its black-box nature, methods for interpreting neural network decisions were applied and new performance measures were defined.

To test the DL performance under different scenarios, a wide variety of datasets was created. These were divided into NME and ME datasets. For each, different MAF ([0.05, 0.1, 0.2, 0.3, 0.4]) and heritability ([0.01, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4]) values were combined to generate different types of datasets. Due to the current limitations of GAMETES and Toxo generators, not every combination of MAF and heritability could be generated for NME datasets. Also, higher-order ME datasets of order four were generated. All datasets were generated using a fixed number of 1000 SNPs and 4000 samples, with

2000 cases and 2000 controls.

Sensitivity analysis is applied to interpret network decisions and evaluate the impact of each input SNP when making predictions in the phenotype. This type of analysis produces a relevance score for each input SNP. Across all iterations of the cross-validation algorithm, sensitivity analysis on the true positive samples is performed. The obtained relevance scores are added resulting in a vector where each index represents a SNP and the value, the total accumulated relevance for that SNP. This vector is sorted in increasing order producing the final ordered relevance vector, where SNPs with higher relevances have higher indexes. Thus, if successful, interacting SNPs will have higher indexes in the output relevance vector.

To evaluate networks in terms of interpretability, a new performance measure was introduced. This type of measure allows neural networks not only to be classified in terms of accuracy but also in terms of interpretability. This is an important step in DL analysis since it provides new insight into the actual information learnt by the DL models.

Chapter 4

Experimental Results

To validate the previously proposed methodology on network interpretation and SNPs detection, a group of experimental tests is performed. From the literature review on the state-of-the-art methods in Chapter 2, it is concluded that CNNs and MLPs are the most used network architectures. The limitations of these neural network architectures regarding interpretability are discussed. Several types of datasets simulating different epistatic scenarios are tested to understand what are the limitations of networks in terms of interpretability.

First, a section describing the hyperparameters used to generate both ME and NME datasets is presented. Further, an initial search space for both CNNs and MLPs is presented based on the most cited works from the literature review [66–69, 73]. Due to the increased difficulty of NME datasets, the results on these datasets are used to prune initial search space for both architectures. The pruned search space is tested on the remaining ME datasets. Moreover, a detailed analysis of the impact of MAF and heritability (h^2) in network interpretability is performed. With this analysis, networks are evaluated not only on accuracy but also on interpretability. Since the interpretability performance measure cannot be calculated on real datasets, a threshold relation between interpretability and other performance measures is analysed. This way, it is possible to define a value from which networks can be trusted.

To conclude and validate the study on simulated datasets, the pruned search spaces are applied to a real Breast Cancer dataset. The discovered interacting SNPs are compared with a previous study that applied an exhaustive search algorithm on the same Breast Cancer dataset.

4.1 Initial Configurations and Setup

In this section, the initial configurations used in the experimental stage are presented. These configurations were common throughout the entire experimental stage, in order to ensure that the results are correctly compared with the minimum error possible.

First, an overview of the datasets used as well as the parameters used to create them is performed. Moreover, the performance measures, as well as the parameters used in the training of the networks, are also introduced. To conclude, the computer setup used to perform all the experiments is presented.

4.1.1 Datasets

As previously described in Section 3.1, ME and NME datasets were created. Despite using the Toxo library to generate penetrance tables for the ME datasets, later these tables are used by GAMETES to generate the corresponding datasets.

When creating the datasets, the number of samples N 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 [54, 67–69] 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.

The models used to generate, both pairwise and higher order, ME datasets were the additive (Table 3.2), multiplicative (Table 3.3), xor (Table 3.4) and threshold (Table 3.5).

Having this variety of datasets allows a more detailed study of the interpretability of DL algorithms since different epistasis scenarios are tested.

4.1.2 Performance measures

To understand the used performance measures it is necessary to know the concept of confusion matrix. A confusion matrix is a table used to measure the performance of a classification model. The predictions made by the model are compared with the ground truth and registered in the table. The diagonal represents the correctly classified samples, the higher the values in the diagonal, the better the classification performance of the model. The terms defining the values in each cell represented in Table 4.1 are:

- **True Positives (TP):** The correctly predicted positive samples (cases)
- **True Negatives (TN):** The correctly predicted negative samples (controls)
- **False Positives (FP):** The prediction was not correct, the sample is negative (control) but the model predicted positive (case)
- **False Negatives (FN):** The prediction was not correct, the sample is positive (case) but the model predicted negative (control)

From the confusion matrix, performance measures like accuracy, precision, recall (sensitivity) and F1 Score can be defined as:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

$$F1 = \frac{2 * (Recall * Precision)}{(Recall + Precision)} \tag{4.4}$$

Table 4.1: Confusion matrix example.

		Real	Class
		1	0
Predicted	1	TP	FP
Class	0	FN	TN

Accuracy (Equation 4.1) is the most intuitive measure since it represents the ratio between the corrected predicted samples and all the predictions made. Precision (Equation 4.2) is the ratio between the correctly predicted case samples among all the samples identified as cases. Recall or Sensitivity (Equation 4.3) represents the ratio between the corrected predicted cases among all case samples. Lastly, the F1 Score (Equation 4.4) represents the weighted average between precision and recall.

Additionally, the ROC curve is also used as a performance measure for classification models. This curve represents the ratio between the True Positive Ratio (TPR, also known as recall) and False Positive Ratio (FPR). The FPR can be defined as:

$$FPR = \frac{FP}{TN + FP} \tag{4.5}$$

The area under the ROC curve is called AUC and measures the ability of the model to distinguish between positive and negative samples. The AUC is the performance measure used. A model with an AUC closer to 1 can distinguish cases and controls easily.

These performance measures together with the previously defined interpretability measure (Equation 3.4) were the metrics used to evaluate networks.

4.1.3 Training Parameters

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.

4.1.4 Setup

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.

4.2 Multilayer Perceptron (MLP)

When considering recent works using DL architectures for epistasis detection, MLPs are one of the most used architectures. In this section, MLPs are evaluated in terms of interpretability under different scenarios.

Initially, hyperparameter optimization must be performed. A set of relevant works from the state-of-the-art architectures is selected [66–69] and the ranges for each hyperparameter are defined. The defined search space is tested on NME datasets and the impact of each hyperparameter in interpretability is evaluated. Due to the increased difficulty of NME datasets, these are the ones used to prune the search space.

Further, using the pruned search space, an analysis of MLP interpretability over NME is performed. These include pairwise and high order datasets. Additionally, the impact of MAF and heritability (h^2) on network interpretability is discussed.

To conclude, a summary of MLP performance across the different tested scenarios is presented.

4.2.1 Architectures and Hyperparameter Optimization

Hyperparameter optimization is an important step in DL optimization. The grid search method is used to exhaustively search for all possible combinations in a predefined search space. Since grid search is an exhaustive method, it is necessary to define a good search space with an efficient range of parameters. To ensure that a correct range of parameters is selected, only the most relevant state-of-the-art architectures were considered. Thus, only the architectures from studies [66–69] were considered.

The following tables represent the considered architectures. Table 4.2 represents the architectures used in [67–69], whereas, Tables 4.3, 4.4 and 4.5 represent three different types of networks used in [66].

To choose the initial search space, a range for hyperparameters such as activation function, number of neurons and number of layers must be defined. The number of dense layers from Tables 4.3, 4.4 and 4.5 is evaluated and the number of layers hyperparameter set to [1,2,3,5]. Additionally, the number of inputs is also considered as hyperparameter. This allows testing MLP interpretability in terms of how much the amount of noise in the input affects the network performance. Other parameters such as the dropout ratio were fixed. According to [66], dropout ratios must be small ($< 0.05\%$) when applying neural networks to epistatic datasets. Thus, the dropout ratio was fixed to 0.03. Moreover, since the use of DL models in epistasis detection consists of a binary classification problem, the output function used is the Sigmoid activation function, as observed in Tables 4.3, 4.4 and 4.5,

The used search space to evaluate MLP interpretability is therefore presented in Table 4.6.

Table 4.2: MLP architecture used in [67–69].

Study	Layer	Type	Input	Output	Activation	Dropout Ratio
[67–69]	0	Input	50×1	50×1	Tanh	-
	1	Dropout	50×1	50×1	-	0.20
	2	Fully Connected	50×1	50×1	Tanh	-
	3	Dropout	50×1	50×1	-	0.50
	4	Fully Connected	50×1	50×1	Tanh	-
	5	Dropout	50×1	50×1	-	0.50
	6	Fully Connected	50×1	50×1	Tanh	-
	7	Dropout	50×1	50×1	-	0.50
	8	Output	50×1	1×1	Sigmoid	-

Table 4.3: MLP architecture used in [66].

Study	Layer	Type	Input	Output	Activation	Dropout Ratio
[66]	0	Input	946×1	946×1	Softplus	-
	1	Dropout	946×1	946×1	-	0.01
	2	Fully Connected	946×1	32×1	Softplus	-
	3	Dropout	32×1	32×1	-	0.01
	4	Fully Connected	32×1	32×1	Softplus	-
	5	Dropout	32×1	32×1	-	0.01
	6	Fully Connected	32×1	32×1	Softplus	-
	7	Dropout	32×1	32×1	-	0.01
	8	Fully Connected	32×1	32×1	Softplus	-
	9	Dropout	32×1	32×1	-	0.01
	10	Fully Connected	32×1	32×1	Softplus	-
	11	Dropout	32×1	32×1	-	0.01
	12	Output	32×1	1×1	Sigmoid	-

Table 4.4: MLP architecture used in [66].

Study	Layer	Type	Input	Output	Activation	Dropout Ratio
[66]	0	Input	946×1	946×1	Elu	-
	1	Dropout	946×1	946×1	-	0.03
	2	Fully Connected	946×1	64×1	Elu	-
	3	Dropout	64×1	64×1	-	0.03
	4	Fully Connected	64×1	64×1	Elu	-
	5	Dropout	64×1	64×1	-	0.03
	6	Output	64×1	1×1	Sigmoid	-

4.2.2 MLP on NME datasets

In this section, the results obtained for the MLP architectures on pairwise NME Datasets are presented. Once the grid search (using the search space defined in Table 4.6) is concluded, the performance metrics for each architecture are calculated. The performance measures presented are the mean values across all cross-validation iterations.

When dealing with non-exhaustive methods for SNP detection, it is more difficult to detect the inter-

Table 4.5: MLP architecture used in [66].

Study	Layer	Type	Output	Activation	Dropout Ratio
[66]	0	Input	946×1	Elu	-
	1	Dropout	946×1	-	0.03
	2	Fully Connected	32×1	Elu	-
	3	Dropout	32×1	-	0.03
	4	Output	1×1	Sigmoid	-

Table 4.6: 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}

acting genes on an NME than on a ME dataset [23]. Thus, a detailed evaluation of the impact each hyperparameter has on the model interpretability is performed on NME datasets. This type of analysis allows classifying model architectures in terms of interpretability performance while also reducing the initial search space.

Number of Input SNPs

The first hyperparameter to be analysed is the number of inputs. Since the used datasets include a pairwise epistasis interaction, only two SNPs are relevant across all the input SNPs. Increasing the number of SNPs means that the number of noisy SNPs in the input increases, making it harder for the network to make accurate predictions.

The results were grouped according to the number of inputs and evaluated. To examine the relation between the different performance measures and interpretability, scatter plots were made. The scatter plots for 50 (Figure 4.1), 100 (Figure 4.2), 500 (Figure 4.3) and 1000 (Figure 4.4) input SNPs are presented.

It is crucial to understand the relationship between other performance measures and interpretability since the latter cannot be calculated if the interacting SNPs are unknown. Thus, in the case of a real dataset where the interacting SNPs are unknown, there needs to be a performance measure able to evaluate if the network can or cannot be trusted.

By analysing the graphs in Figure 4.1 and Figure 4.2, correspondent to 50 and 100 input SNPs respectively, a relation between interpretability and some performance measures like accuracy, precision and AUC is observed. With the increasing value of these performance measures, interpretability values start to increase as well. Thus, the definition of these thresholds is crucial to understand if networks can be trusted or not.

The threshold value from which maximum interpretability for every evaluated network is reached

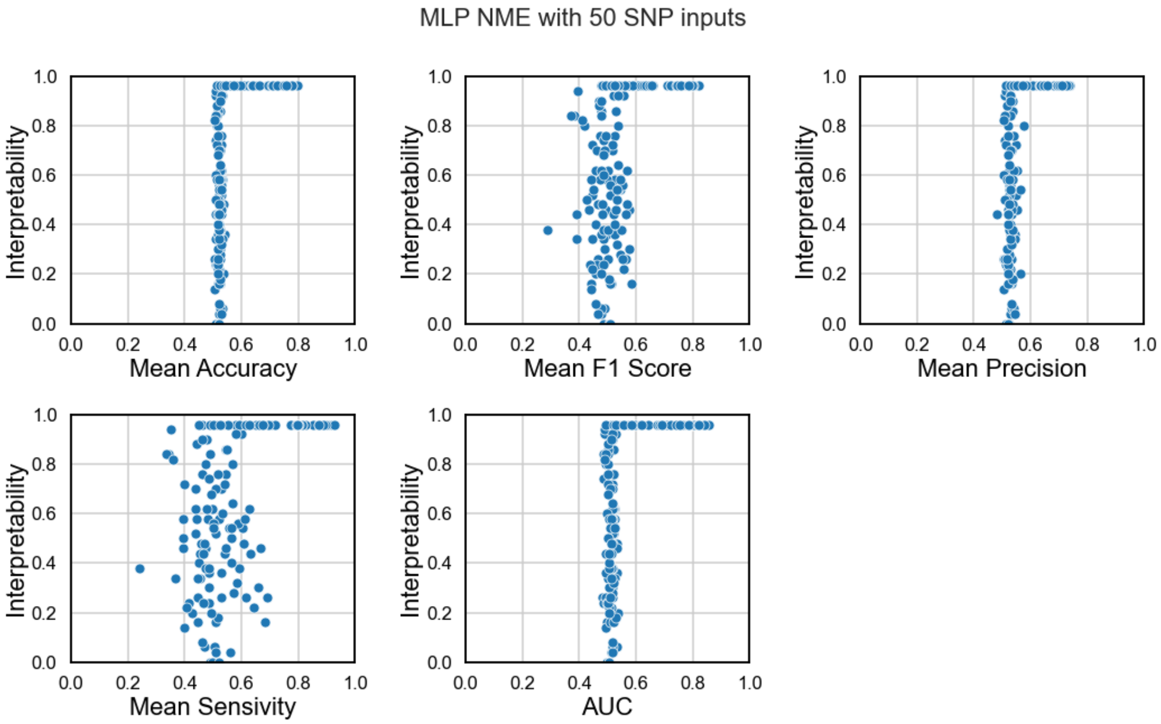


Figure 4.1: MLP relation between interpretability and other performance measures on NME datasets with 50 input SNPs.

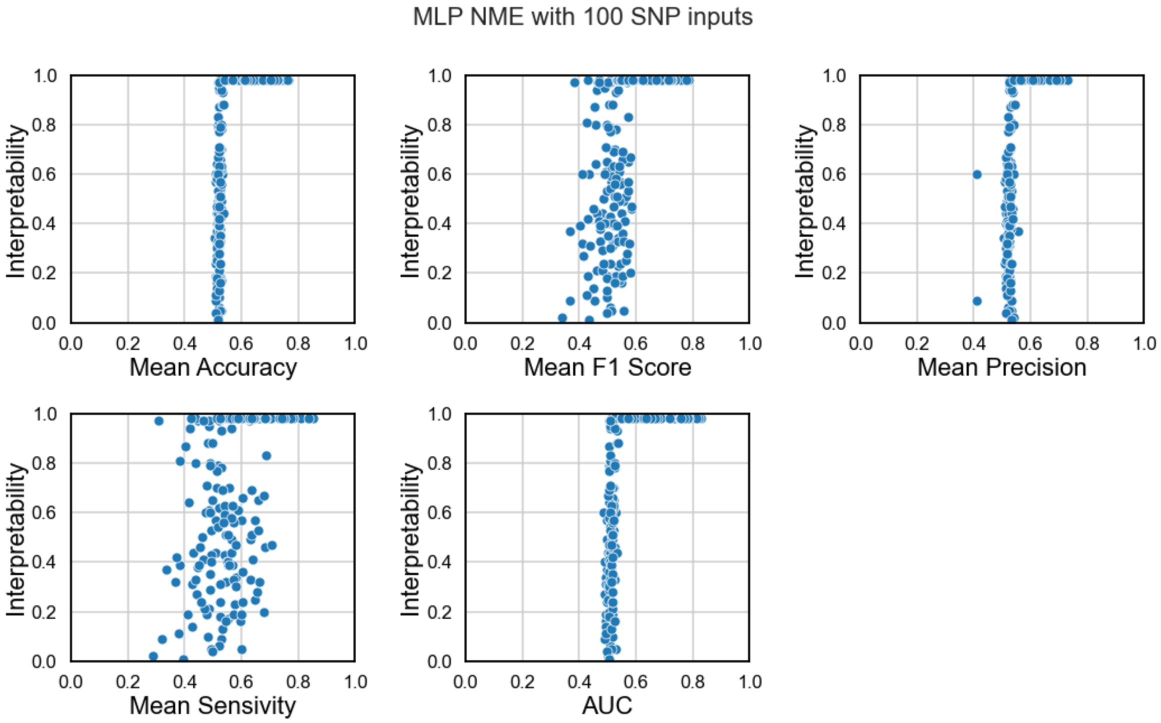


Figure 4.2: MLP relation between interpretability and other performance measures on NME datasets with 100 input SNPs.

is presented in Table 4.7. From the total networks tested (180), the percentage of networks whose performance values are above the defined threshold are also presented. From Table 4.7 it is concluded that accuracy and AUC have the same relation with interpretability since the percentage of networks with performance values above the presented thresholds is, in both accuracy and AUC, 43% for 50

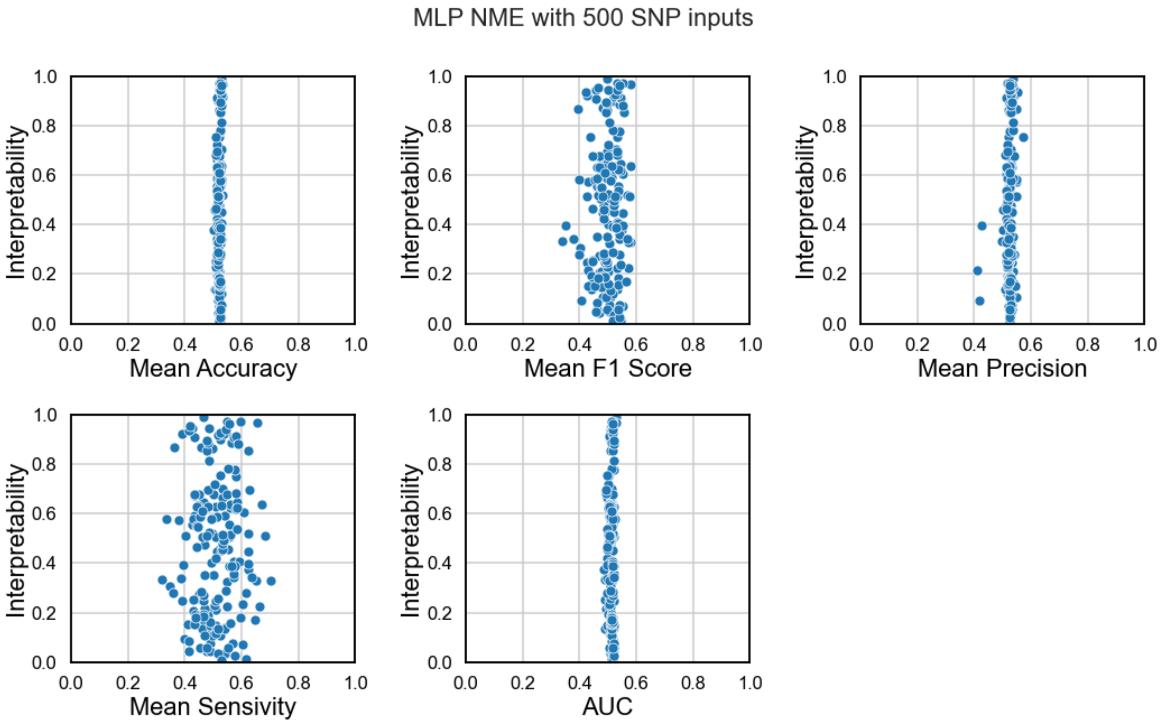


Figure 4.3: MLP relation between interpretability and other performance measures on NME datasets with 500 input SNPs.

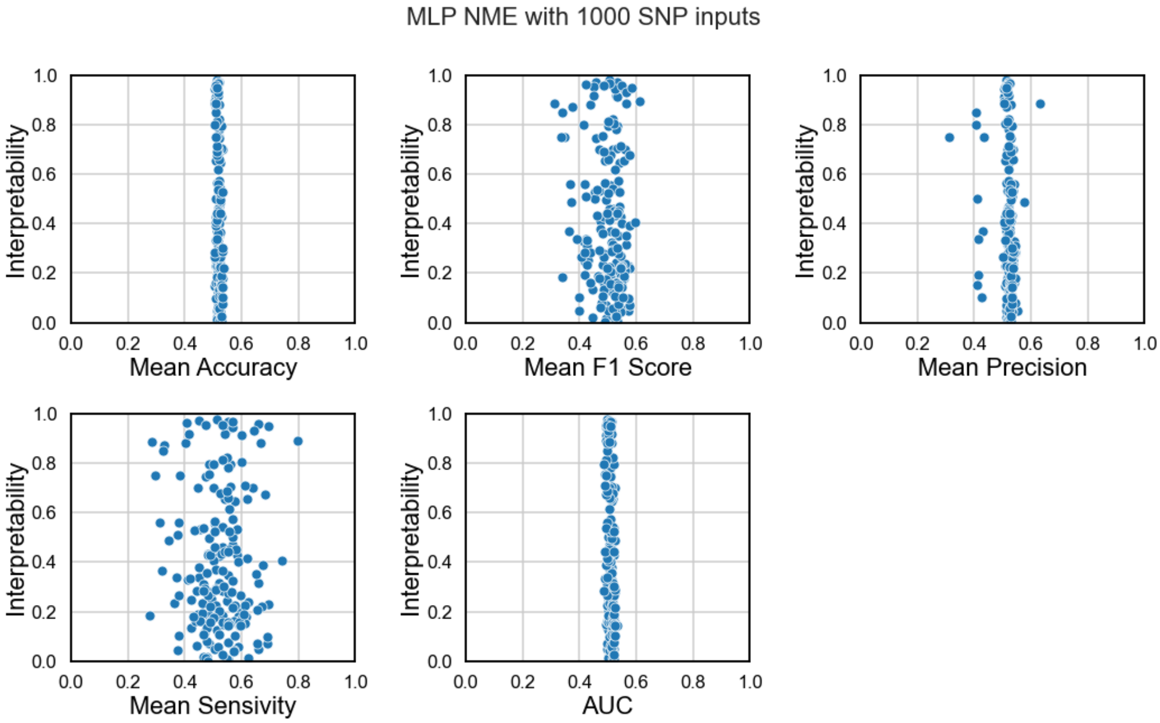


Figure 4.4: MLP relation between interpretability and other performance measures on NME datasets with 1000 input SNPs.

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 4.7 the drop in performance associated with the increase of input SNPs is also confirmed since the number of trustworthy networks decreases across all performance measures.

When increasing of the number of inputs to 500 (Figure 4.3) and 1000 (Figure 4.4), results across all performance measures start to get worse. Networks have difficulty fitting these types of datasets since there is a great amount of noise associated with them. Some networks were able to reach high interpretability values, however, networks having the same accuracy show different interpretability values, making them not reliable.

Based on the previous analysis, interpretability and accuracy are the two performance measures used to evaluate the remaining hyperparameters, since these have proven to be related.

Table 4.7: MLP threshold values for each performance measure to achieve maximum interpretability.

Nº Inputs	Performance Measure	Threshold Value	Maximum Interpretability Networks (%)
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

Activation Function and Number of Layers

To understand which architectures had the best performance, an analysis of the activation functions and the number of layers is performed.

In the previous analysis on the number of inputs, it was observed that the networks having 500 and 1000 Input SNPs did not fit the training data. Thus, when evaluating activation functions and the number of layers, only results from 50 and 100 input SNPs are considered.

In Figure 4.5 and Figure 4.6, a scatter plot, histogram and density curve for 50 and 100 input SNPs is presented. These plots display the distribution of activation functions across the different interpretability values. In both, from the density curve, it is observed that networks with higher interpretability values, have more frequently Tanh and Elu as activation functions.

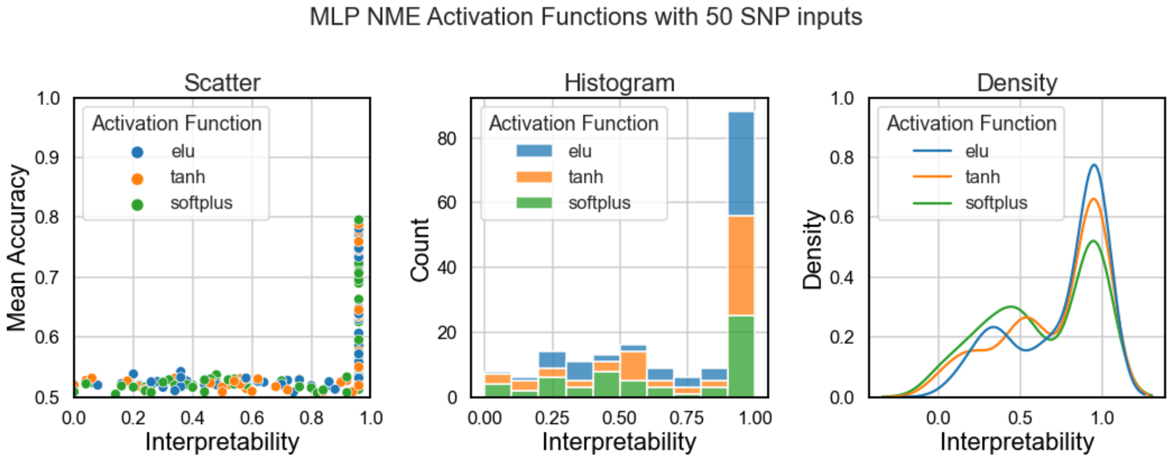


Figure 4.5: MLP activation function analysis on NME datasets with 50 inputs.

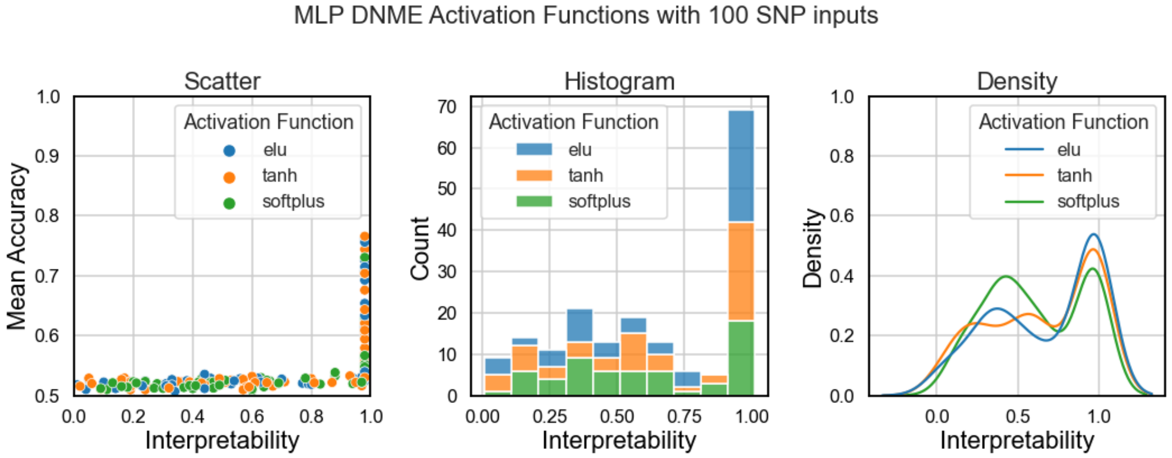


Figure 4.6: MLP activation function analysis on NME datasets with 100 inputs.

Further, the same analysis is repeated for the number of layers. In Figure 4.7 and Figure 4.8, the plots for the 50 and 100 input SNPs are presented. The distribution of the number of layers across different interpretability values is presented.

In the case of 50 Input SNPs (Figure 4.7), there is no clear distinction on which architecture has the best performance since the density curves similar. However, close to the interpretability of 0.5, there is a peak of architectures with 1 hidden layer, which indicates that single-layer architectures are associated with lower interpretability values.

With 100 input SNPs (Figure 4.8), from the density curve, it is observed that architectures with 1 and 2 hidden layers have interpretability values lower than 0.5 more often. Once interpretability values start increasing, the density curves of the 5 and 2 hidden layers have higher values when compared with the single-layer architectures. Thus, single layers networks have lower interpretability values when compared to 5 and 2 hidden layers and can be removed from the search space.

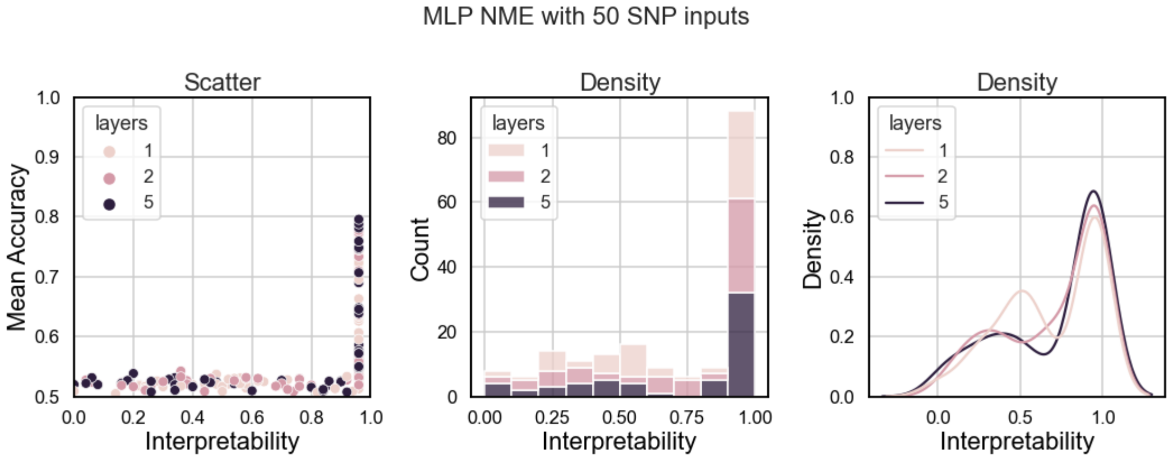


Figure 4.7: MLP number of layers analysis on NME datasets with 50 inputs.

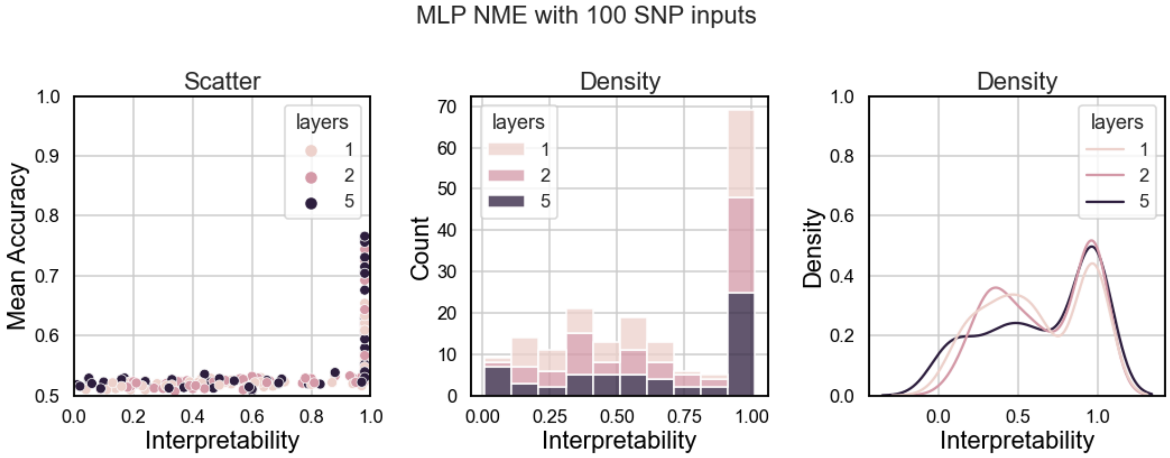


Figure 4.8: MLP number of layers analysis on NME datasets with 100 inputs.

Pruned Search Space

When pruning the search space, only results from 50 and 100 input SNPs were considered, since these networks were the ones able to fit several datasets.

Based on the previous analysis on the number of layers and activation functions, it is concluded that the use of the softplus activation function and a reduced number of hidden layers are associated with networks with lower interpretability values. Thus, softplus and single-layer networks are removed from the search space.

The pruned search space is presented in Table 4.8 and will be used to evaluate the MLP performance on the remaining datasets. The list of tested architectures for a specific input size N, is represented in Table A.2.

Table 4.8: Pruned search space to evaluate MLP interpretability.

Architecture	Hyperparameter	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}

4.2.3 MLP on ME datasets

Once the analysis on NME datasets is concluded, ME datasets are considered. These datasets include pairwise and higher-order datasets, whose penetrance tables were generated using the Toxo library. The additive, multiplicative, threshold and xor models are herein considered.

When compared to the NME datasets, ME datasets tend to be easier for non-exhaustive algorithms to detect interactions [23]. In these datasets, individual SNPs may have information about the phenotypes, thus actual interactions do not need to be detected. Since the difficulty of detecting interacting

SNPs is lower on NME datasets, it is not necessary to perform a detailed analysis on hyperparameter performance as in ME datasets. Network architectures working on NME datasets will work on ME datasets, due to their reduced complexity.

Using the conclusions from the NME datasets, the pruned search space from Table 4.8 is used to evaluate MLP networks. The use of a reduced search space makes this analysis more efficient. Also, accuracy and interpretability are used as the two performance measures to classify architectures.

Pairwise interactions

First, pairwise interactions are considered. Several epistasis models were covered to ensure that a wide variety of possible scenarios is tested. Tests were performed on four different models across several MAF and heritability values. Due to the great number of tested networks, a summary of the MLP performance across four models and the different number of input values are presented in Figure 4.9.

By examining the plots in Figure 4.9 it is confirmed that MLP networks have less difficulty fitting to these datasets when compared to the NME datasets. For the additive and multiplicative models, the networks were able to detect the interacting SNPs regarding the number of input SNPs. The noise in the input does not affect the model interpretability for these models. Additionally, networks having 500 and 1000 input SNPs were able to capture the interacting SNPs, whereas, in NME datasets, none of them could. Independently of the dataset model, the performance of the networks was better when compared to NME models.

MLP networks show great potential when detecting the interacting SNPs in models displaying marginal effects. These networks show high interpretability values even in the presence of noise across different epistasis models. Thus, the number of input SNPs does not have an impact on model interpretability in ME datasets.

High-order interactions

Additionally, high-order interactions are considered. The difference between high-order and pairwise interactions is in the number of interacting genes. In this section, epistatic interactions of order four are considered. Due to the great number of datasets analysed, a summary of all the results is presented in Figure 4.10.

From the results, it is observed a drop in interpretability across all the models. This type of result was expected since, for higher-order datasets, more SNPs need to be captured by the network, thus increasing the difficulty of the epistasis detection task.

Also, as observed in the case of pairwise interactions the increase in the number of inputs does not have an impact on network interpretability, with networks being able to detect the four interacting SNPs in datasets with 500 and 1000 input SNPs. Therefore, these results confirm that in the presence of marginal effects MLP networks can filter noisy SNPs and select the interacting ones.

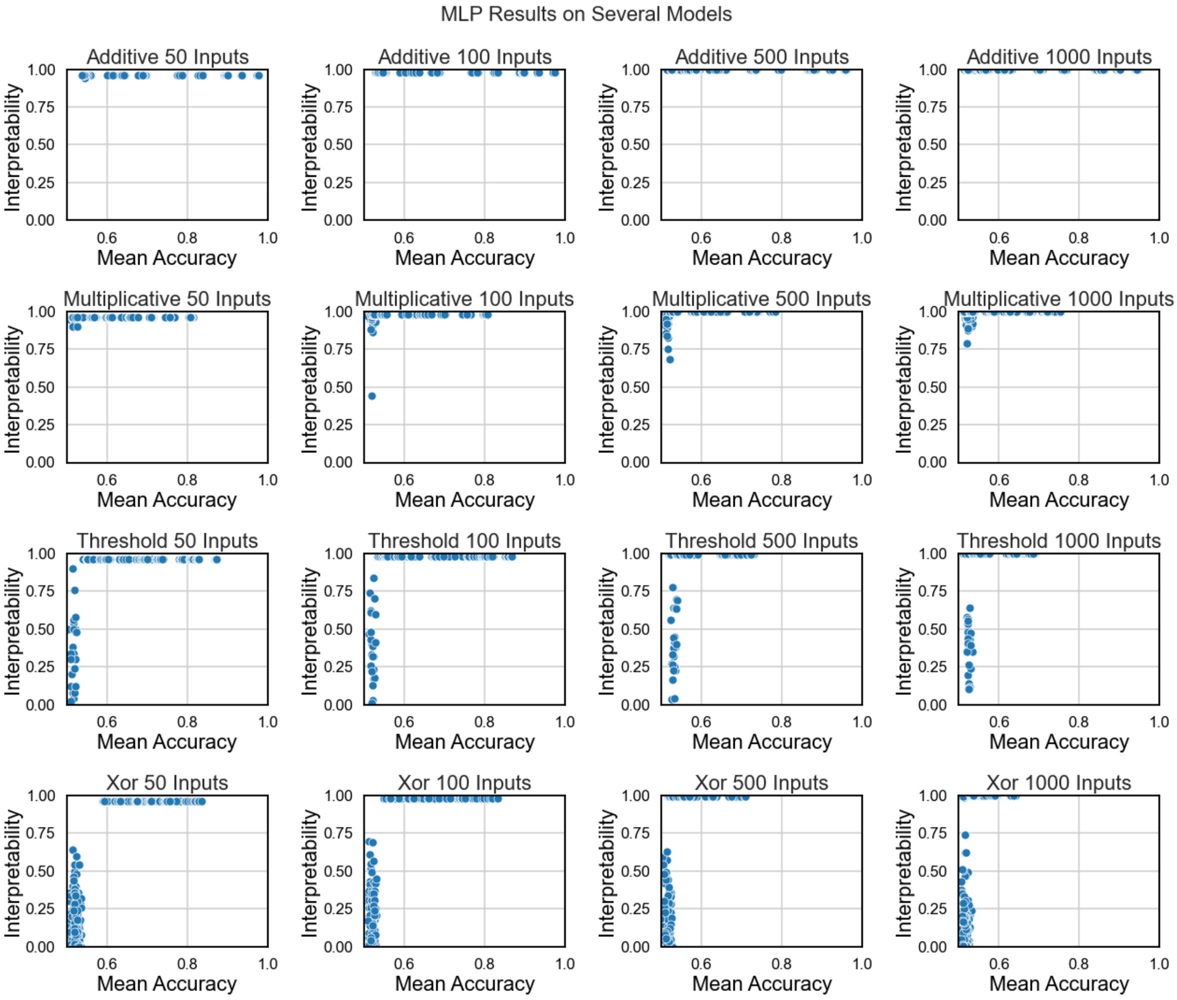


Figure 4.9: MLP interpretability analysis on ME datasets.

4.2.4 Heritability and MAF

To understand the limitations of MLP networks in terms of interpretability, several datasets having different MAFs and heritabilities (h^2) were tested. This allows networks to be evaluated under different epistatic scenarios. Due to the increased difficulty of NME datasets, only those are considered in this section, since including the ME datasets in the analysis would not add any relevant information.

To ensure a correct analysis, several MAFs and heritability (h^2) values were tested. The pruned search space from Table 4.8 is used to perform this analysis. The values used for MAF were [0.05, 0.1, 0.2, 0.3, 0.4] and for heritability were [0.01, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4]. Also, to avoid error in the results, the results from architectures having 500 and 1000 input SNPs were removed from this analysis, since these networks failed to produce reliable networks. In Figure 4.11 the plots displaying the results for 50 and 100 Input SNPs are presented.

First, an analysis of the impact of heritability is performed. Heritability controls the proportion of observable differences between individuals caused by genetic differences. The higher the heritability value, the more information the dataset will have. Thus, the higher the heritability, the easier it is for the network to learn from the dataset.

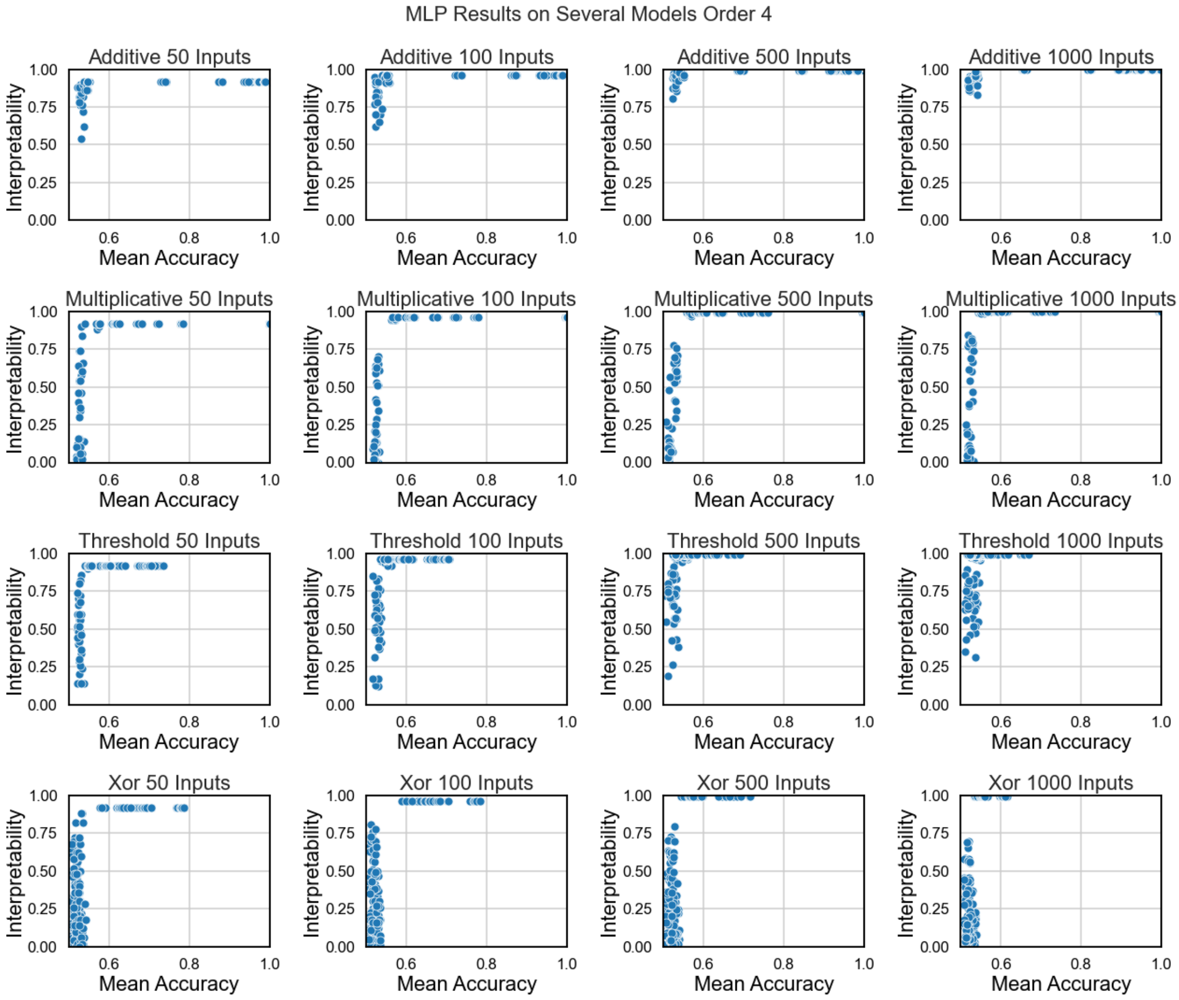


Figure 4.10: MLP interpretability analysis on ME datasets of order 4.

For datasets having lower heritability values ($h^2 = 0.01$ and $h^2 = 0.05$), the networks do not fit the dataset at any MAF. No threshold relation between accuracy and interpretability is observed, meaning that networks having the same accuracy value can have very distinct interpretability values. Thus for heritability values of $h^2 = 0.01$ and $h^2 = 0.05$, no reliable networks were found. Once the heritability value starts increasing ($h^2 \geq 0.1$), networks start having higher interpretability and accuracy values. The accuracy threshold from which networks have maximum interpretability starts to be visible. The box plots from Figure 4.12 confirm the previous conclusions.

Additionally, for $h^2 = 0.4$ and $h^2 = 0.3$ all architectures managed to capture the interacting SNPs (maximum interpretability is achieved). Also, for $h^2 > 0.2$, if the outliers are excluded, it is observed that most architectures managed to reach maximum interpretability value. Thus, it is concluded that, with the increase of heritability, networks start having better performance and being more trustworthy.

Further, an analysis of the impact of MAF in MLP interpretability is performed. MAF controls the frequency at which the second most common allele occurs in a population. From Figure 4.12 it is hard to derive any conclusion since no clear pattern is found. For $MAF \geq 0.2$ network performance starts to improve, however, since there is a significant drop in interpretability performance from $MAF = 0.1$ to $MAF = 0.2$, no conclusion can be derived regarding the increase of MAF.

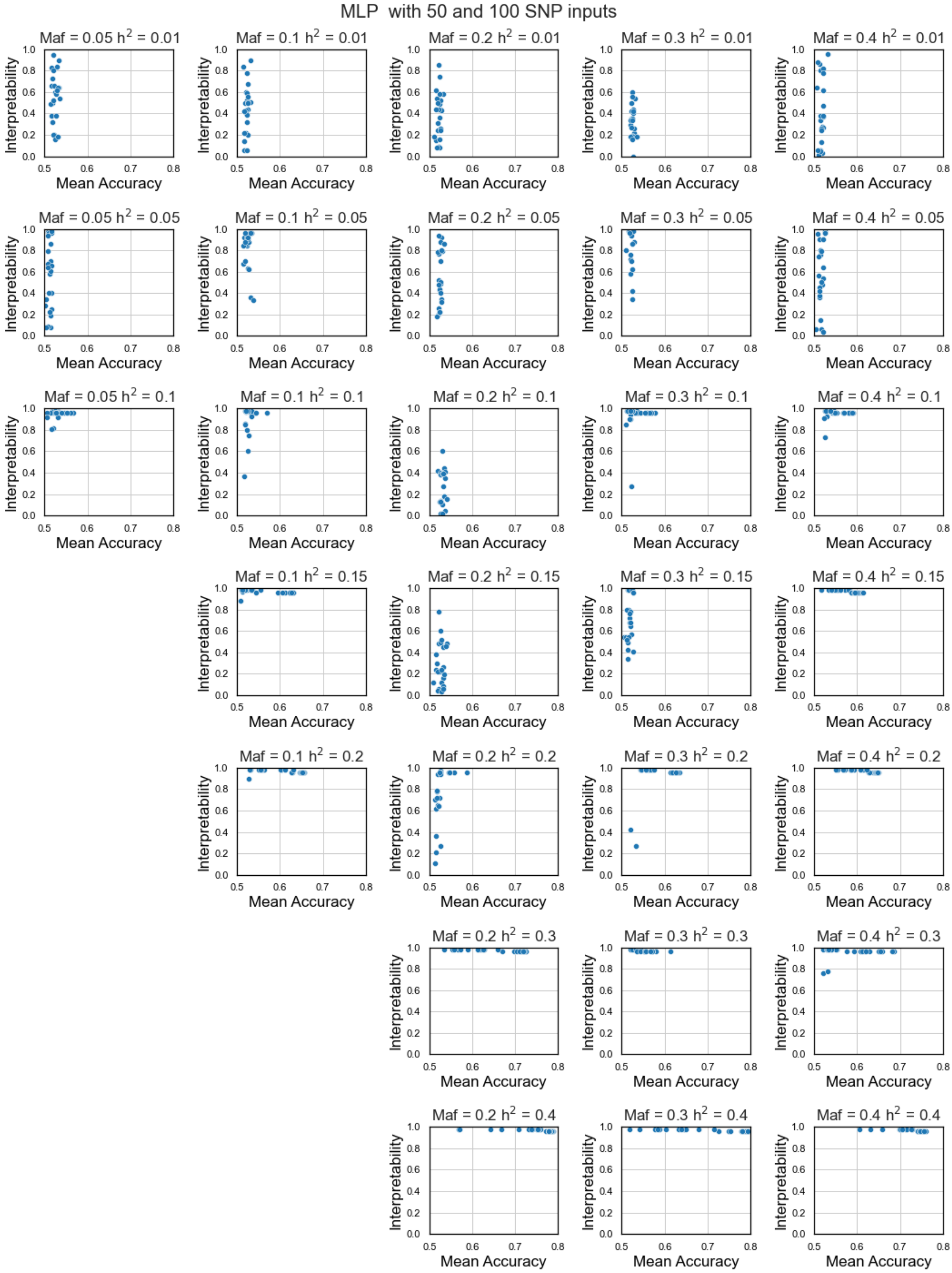


Figure 4.11: MLP interpretability analysis on several MAF and heritability values.

4.2.5 Summary

In this section, the limitations of MLP interpretability were studied on different epistasis scenarios.

First, the most cited works from state-of-the-art architectures were analysed [66–69] and a search

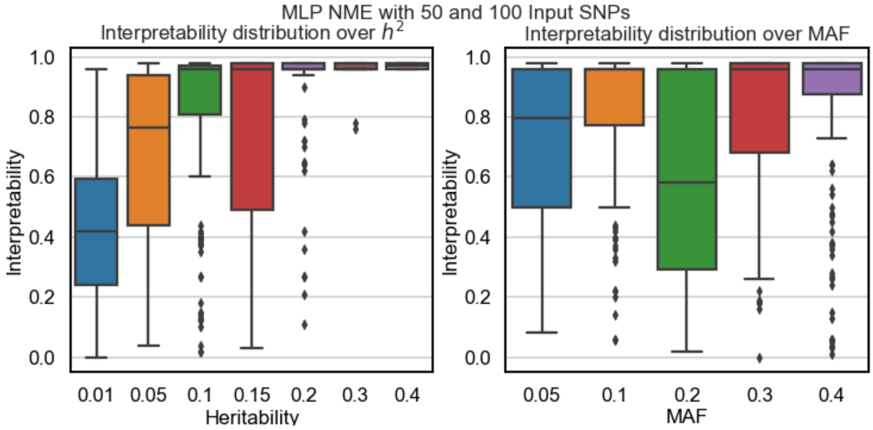


Figure 4.12: MLP boxplot analysis on the impact of MAF and heritability on interpretablity.

space for hyperparameter pruning was defined. Next, an analysis on the relation between interpretability and other performance measures revealed that there is a threshold relation between accuracy and interpretability, meaning that networks having an accuracy over the defined threshold will have maximum interpretability. The threshold values found were represented in Table 4.7.

Due to the increased difficulty of NME datasets, these were used to evaluate hyperparameter performance and prune the initial search space. The results revealed that the Softplus activation function and single-layer networks were more frequently associated with networks having lower interpretability, thus these were removed from the initial search space. Also, the number of inputs revealed to have a great impact on MLP performance where only networks having 50 and 100 input SNPs were able to fit the data and have trustworthy results.

Next, tests on ME datasets were performed using the previously pruned search space. 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.

To conclude, a detailed analysis of MLP interpretability under different MAFs and heritabilities was performed. The increase in heritability was revealed to improve network performance with all networks achieving maximum interpretability with $h^2 = 0.3$ and $h^2 = 0.4$. By increasing heritability, the amount of genetic information in the dataset increases and networks have less difficulty extracting it. On the other hand, no conclusions could be derived from MAF.

This analysis on MLP networks revealed their great potential in detecting epistasis under different epistasis scenarios, especially in the presence of marginal effects and for high heritability values ($h^{=}2$). Also, the ability to detect higher-order interactions makes MLPs promising approaches.

4.3 Convolutional Neural Networks (CNN)

From the state of the art review, it is noticed that among all DL architectures, CNNs are one of the most common networks being used in epistasis detection. These networks have been highlighted as one promising approach to solve the epistasis detection problem [11, 66]. In this section, CNNs are tested under different epistasis scenarios.

First, an analysis of the impact of each hyperparameter on model interpretability is performed. The most relevant works from the state-of-the-art architectures are selected and ranges for each hyperparameter selected. The defined search space is tested on NME datasets and pruned.Non-Marginal Effects datasets, including pairwise and high order, are analysed using the pruned search space and the impact of MAF and heritability in model interpretability is discussed.

To conclude, a summary of all the performed tests is presented.

4.3.1 Architectures and Hyperparameter Optimization

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 [66, 73].

The architectures used in [73] are presented in Table 4.9, whereas, the architectures used in [66] are presented in Tables 4.10, 4.11 and 4.12. Taking these architectures into consideration, the initial search space was defined.

When defining a search space for CNN networks, the number of hyperparameters to be considered is higher when compared to MLP architectures. The considered hyperparameters were the number of inputs, kernel size, number of convolutional layers, number of filters, number of classification layers and the correspondent number of neurons. The dropout ratio was fixed to 0.01 as suggested in [66]. Also, since the use of DL models in epistasis detection consists of a binary classification problem, the output function used is the Sigmoid activation function, as observed in Tables 4.9, 4.10, 4.11 and 4.12,

If the search space was defined as it was defined in section 4.2.1, and all hyperparameters were considered, a total number of 1152 architectures should be tested for each dataset. This would make grid search very computationally demanding, thus, a different strategy was defined. Initially, the number of input SNPs is fixed to 50 input SNPs, reducing the number of tested architectures per dataset to 288. The impact of each hyperparameter is analysed and the search space is reduced. The best-performing ones are selected for testing on 100, 500 and 1000 inputs. The initial search space is defined in Table 4.13.

4.3.2 CNN on NME datasets

In this section, the performance of CNN networks is evaluated on datasets without marginal effects (NME). The previously defined search space from Table 4.13 is used to perform grid search and evaluate CNN interpretability.

When compared with datasets having marginal effects, NME datasets tend to be more difficult for non-exhaustive methods when it comes to detecting the interacting SNPs [23]. Hence, these datasets are used to prune the search space which will be further used to evaluate CNN performance on the remaining datasets.

Table 4.9: CNN architecture used in [73].

Study	Layer	Type	Output	Kernel Size	Stride	Activation	Dropout Ratio
[73]	0	Input	1×200×1	-	-	-	-
	1	Convolutional	1×200×64	1	1	Relu	-
	2	Dropout	1×200×64	-	-	-	0.2
	3	Convolutional	1×200×32	1	1	Relu	-
	4	Dropout	1×200×32	-	-	-	0.2
	5	Flatten	1×6400	-	-	-	-
	6	Dense	1×128	-	-	Relu	-
	7	Dropout	1×128	-	-	-	0.5
	8	Dense	1×32	-	-	Relu	-
	9	Dropout	1×32	-	-	-	0.5
	10	Dense	1×1	-	-	Sigmoid	-

Table 4.10: CNN architecture used in [66]

Study	Layer	Type	Output	Kernel Size	Stride	Activation	Dropout Ratio
[66]	0	Input	1×945×1	-	-	-	-
	1	Convolutional	1×944×16	3	1	Linear	-
	2	Dropout	1×944×16	-	-	-	0.01
	3	Flatten	1×15104	-	-	-	-
	4	Dense	1×32	-	-	Linear	-
	5	Dropout	1×32	-	-	-	0.01
	6	Dense	1×1	-	-	Sigmoid	-

Table 4.11: CNN architecture used in [66]

Study	Layer	Type	Output	Kernel Size	Stride	Activation	Dropout Ratio
[66]	0	Input	1×945×1	-	-	-	-
	1	Convolutional	1×945×32	2	1	Elu	-
	2	Dropout	1×945×32	-	-	-	0.01
	3	Flatten	1×30240	-	-	-	-
	4	Dense	1×32	-	-	Elu	-
	5	Dropout	1×32	-	-	-	0.01
	6	Dense	1×32	-	-	Elu	-
	7	Dropout	1×32	-	-	-	0.01
	8	Dense	1×32	-	-	Elu	-
	9	Dropout	1×32	-	-	-	0.01
	10	Dense	1×1	-	-	Sigmoid	-

Initial Search Space

To understand the relation between interpretability and the other performance measures, scatter plots in Figure 4.13 are presented. As observed with the MLP architectures, Accuracy and AUC have 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. When compared

Table 4.12: CNN architecture used in [66]

Study	Layer	Type	Output	Kernel Size	Stride	Activation	Dropout Ratio
[66]	0	Input	1×945×1	-	-	-	-
	1	Convolutional	1×945×16	2	1	Softplus	-
	2	Dropout	1×945×16	-	-	-	0.01
	3	Flatten	1×15120	-	-	-	-
	4	Dense	1×64	-	-	Softplus	-
	5	Dropout	1×64	-	-	-	0.01
	6	Dense	1×64	-	-	Softplus	-
	7	Dropout	1×64	-	-	-	0.01
	8	Dense	1×64	-	-	Softplus	-
	9	Dropout	1×64	-	-	-	0.01
	10	Dense	1×1	-	-	Sigmoid	-

Table 4.13: 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 Neurons	[32,64]
	Nº Classifier Layers	[2,3]
	Activation Function	[Linear, Elu, Softplus, Relu]
	Dropout Ratio	0.01
	Learning Rate	1×10^{-3}

with MLP performance (Table 4.7), it is observed that the percentage of networks with maximum interpretability and above the threshold value is lower in CNNs for 50 input SNPs. This is caused by the increased number of tested CNN networks (2904) when compared to MLP networks (180), thus these results cannot be compared since they were tested under different conditions.

Precision, sensitivity and F1 score performance measures did not show any relation with interpretability. There is no precision, sensitivity or F1 score threshold from above which networks display maximum interpretability values. Thus, to analyse the remaining hyperparameters, accuracy and interpretability are used since these have proven to be related.

Activation Function and Classifier Layers

Next analysis on the activation function hyperparameter is performed. The scatter, histogram and density curves are presented in Figure 4.14. It is noticed a considerable better performance of the Elu and Relu over the Softplus and Linear activation functions. Both histogram and density curves show that networks having higher interpretability values include the Relu and Elu activation functions more frequently. Thus, the Linear and Softplus functions can be removed from the search space.

In Figure 4.15 an analysis of the number of classifier layers is presented. No difference between the

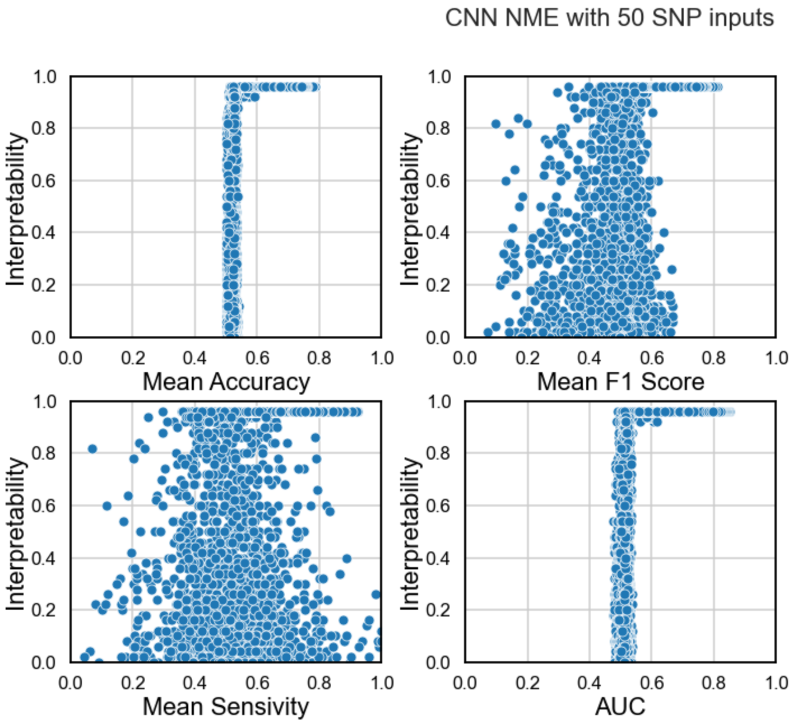


Figure 4.13: CNN relation between interpretability and other performance measures on NME datasets with 50 input SNPs.

two density curves is observed. Hence, the distribution of networks having two or three classification layers is the same across different interpretability values, meaning that this hyperparameter does not influence model interpretability. Thus, the number of classifier layers is fixed to three.

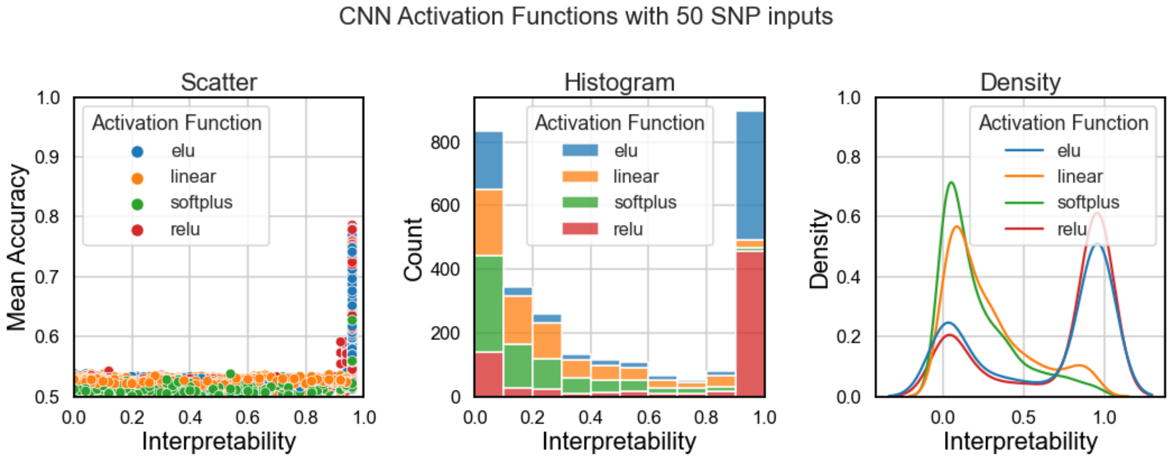


Figure 4.14: CNN activation function analysis on NME datasets with 50 inputs.

Kernel Size and Convolutional Layers

When evaluating the convolutional part of a CNN, the kernel size and number of convolutions hyperparameters must be considered. An analysis of both parameters is presented in Figure 4.16.

The first hyperparameter to be considered is the kernel size. From the density curve analysis, it is observed that networks having lower interpretability values are normally associated with kernels three

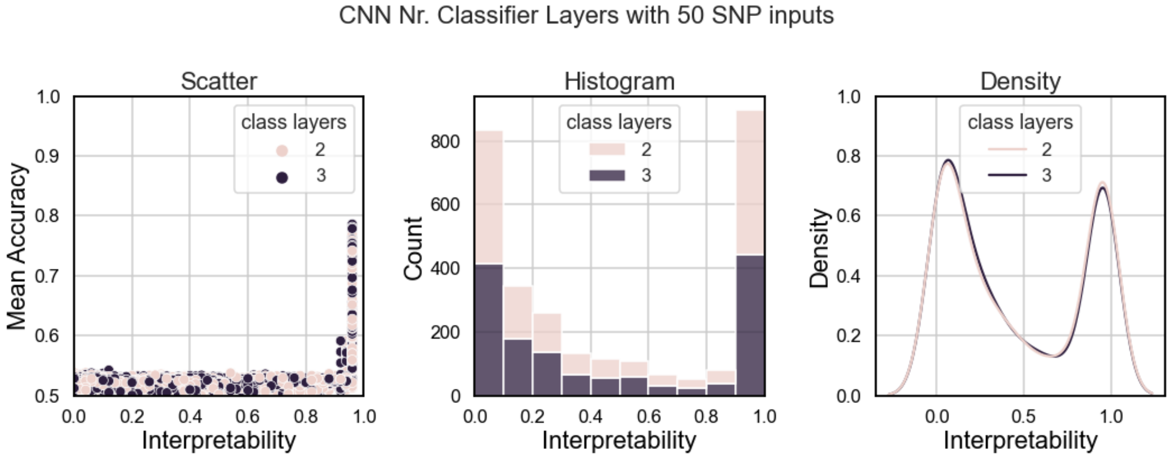


Figure 4.15: CNN number of classification layers analysis on NME datasets with 50 inputs.

and two. On the other hand, network architectures having higher interpretability have smaller kernels of sizes of one. This is observed through the peaks in the density curve. This result confirms what is concluded in [66], which states that when applying CNNs in epistasis datasets, smaller kernel sizes have better performance.

The number of convolutions is also considered in Figure 4.16, however, the results are inconclusive. No clear distinction between having one or two convolutional layers is presented. Hence both of these are kept in the pruned search space.

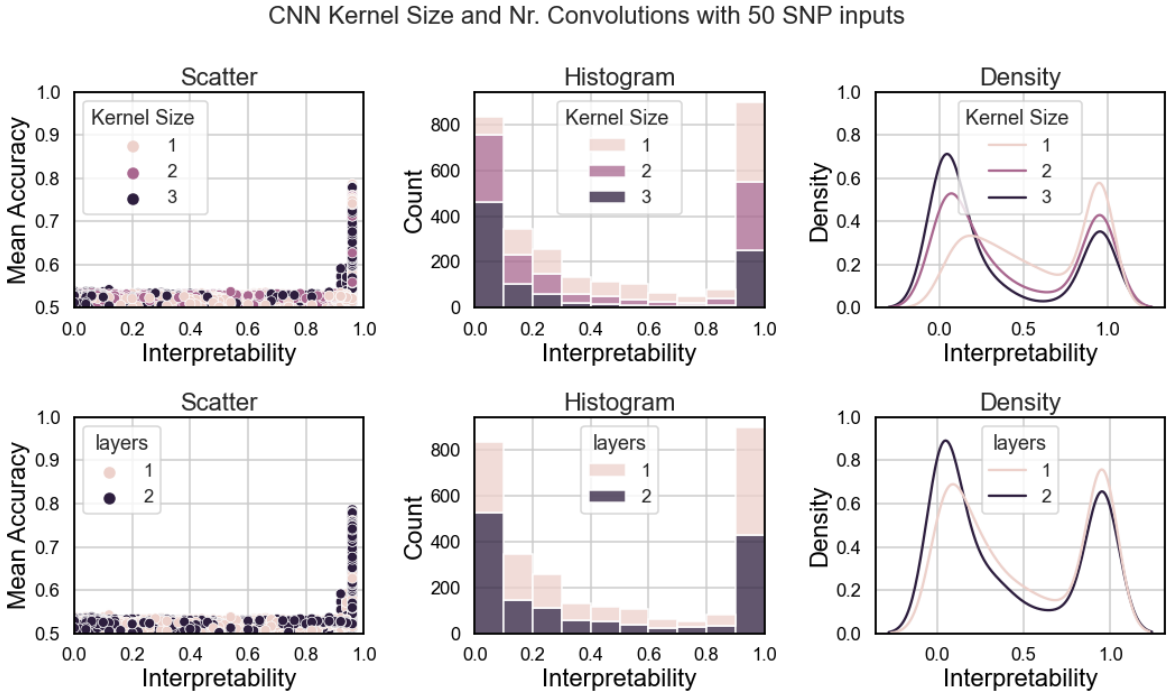


Figure 4.16: CNN kernel size and number of convolution layers analysis on NME datasets with 50 inputs.

Pruned Search Space and Number of Input SNPs

Based on the analysis of the activation function, kernel size, number of convolution layers and number of classifier layers hyperparameters, a smaller search space is defined. In Table 4.14 the new search space is presented. With this reduced search space, only 144 architectures are tested per dataset. This way, the search space used to evaluate architectures with 100, 500 and 1000 input SNPs is smaller and more efficient. In Table A.1 all the tested architectures for a specific input size N, are presented.

Table 4.14: Pruned 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 is reduced, the remaining input values of 100, 500 and 1000 input SNPs are tested. Increasing the number of inputs increases the difficulty of the tested dataset since the number of noisy SNPs increases. The results are grouped according to the number of inputs and evaluated. Scatter plots for 100 (Figure 4.17), 500 (Figure 4.18) and 1000 (Figure 4.19) input SNPs are presented.

The threshold value from which maximum interpretability for every network is reached is presented in Table 4.15. Also, from the total networks tested (480), the percentage of networks whose performance values are above the defined threshold (trustworthy) are also presented. Similar to MLP architectures, increasing the number of SNPs causes architectures to have a significant drop in performance. The number of trustworthy networks reduces from 47% to 11% and further to 6% with the increase of the number of inputs from 100 to 500 and 1000 SNPs. Still, in CNN networks, the threshold relation between accuracy and interpretability is visible across 500 (Figure 4.18) and 1000 (Figure 4.19) 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 4.15: CNN threshold values for each performance measure to achieve maximum interpretability.

Nº Inputs	Performance Measure	Threshold Value	Maximum Interpretability Networks (%)
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

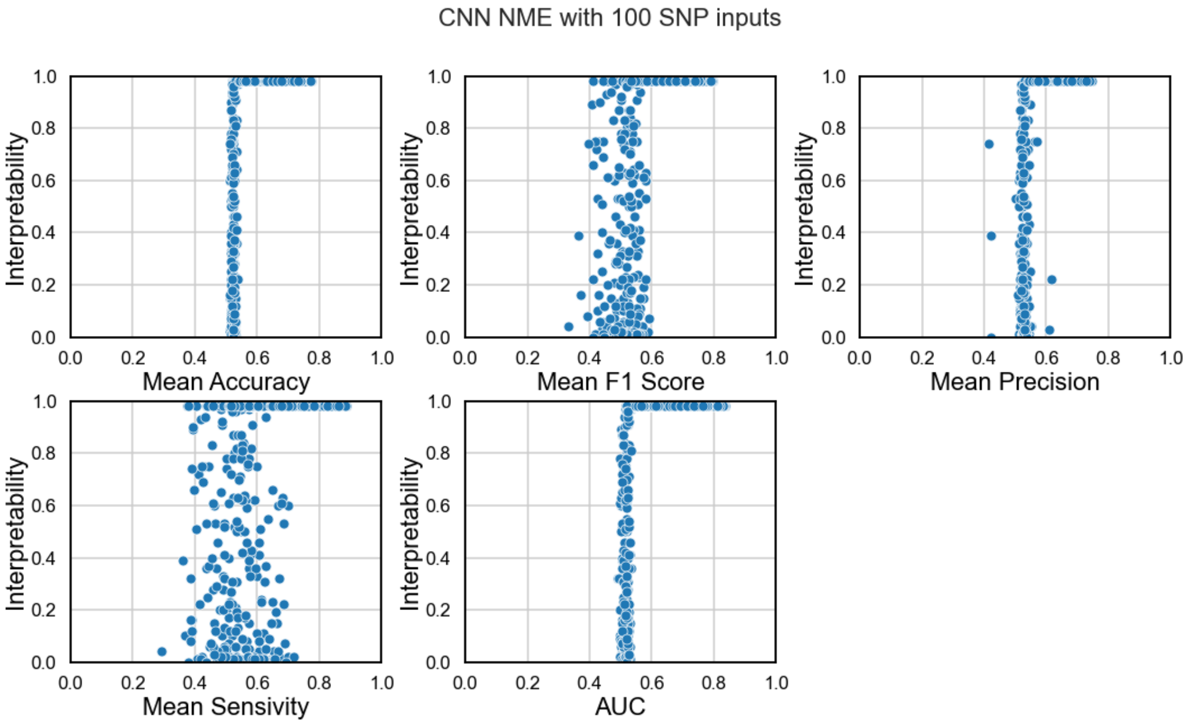


Figure 4.17: CNN relation between interpretability and other performance measures on NME datasets with 100 input SNPs.

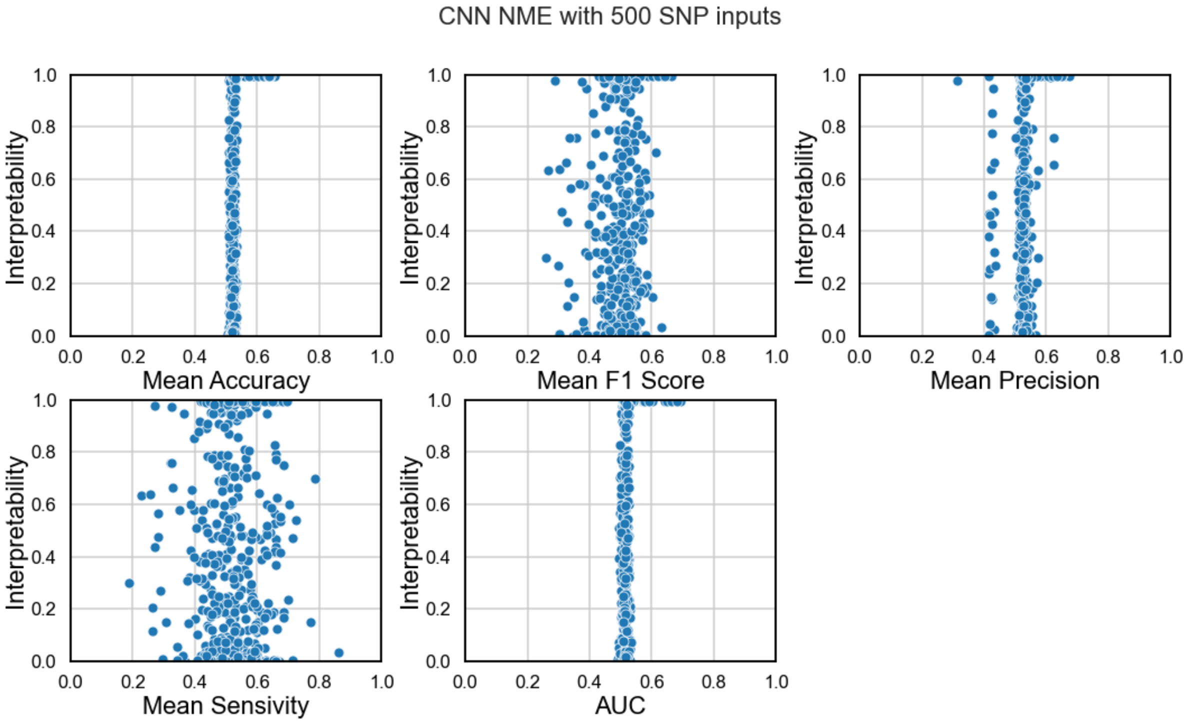


Figure 4.18: CNN relation between interpretability and other performance measures on NME datasets with 500 input SNPs.

4.3.3 CNN on ME datasets

Once the analysis on NME datasets is concluded, CNN interpretability is evaluated on ME datasets. The pruned search space from Table 4.14 is evaluated on additive, multiplicative, threshold and xor epistasis

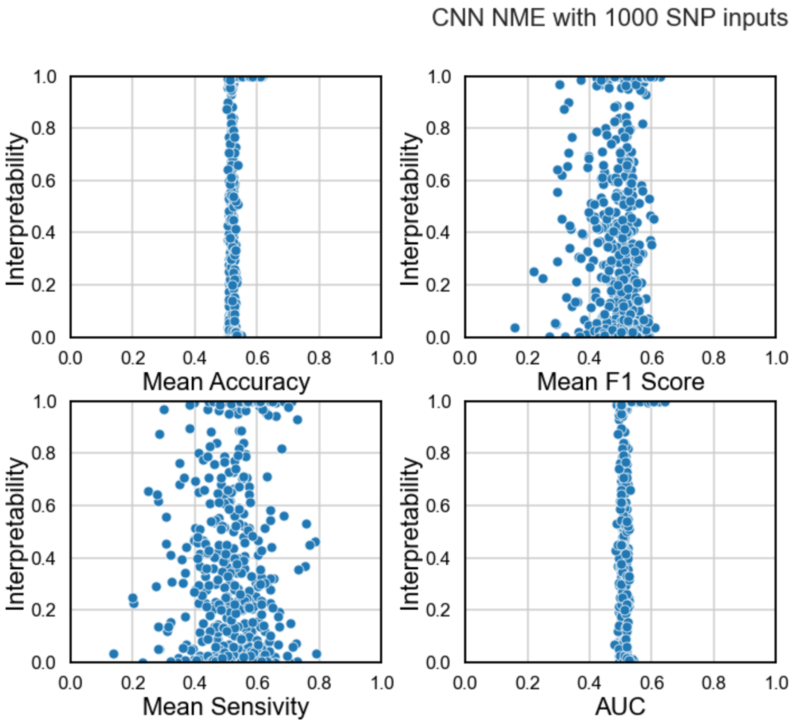


Figure 4.19: CNN relation between interpretability and other performance measures on NME datasets with 1000 input SNPs.

models.

Similar to the MLP analysis in Section 4.2.3, a less detailed analysis on hyperparameter performance is performed on ME datasets. These datasets are easier for non-exhaustive methods to detect the interacting SNPs [23]. Thus, networks having good performance in NME datasets will also have a good performance in ME datasets.

Furthermore, interpretability and accuracy are used as the two performance measures to classify networks since these have proven to be related.

Paiwise interactions

First, pairwise interactions are tested. The performance of CNNs is evaluated across several models, MAF and heritability values to ensure a wide variety of scenarios is covered. To facilitate the analysis of all results a summary of networks performance across the four models is presented in Figure 4.20.

From Figure 4.20, it is confirmed that CNN architectures have less difficulty detecting the interacting SNPs in models displaying marginal effects. The increase in the number of inputs does not seem to have an impact on model interpretability since networks still achieve the maximum value. As observed in the MLP networks, the best performing models are the additive and multiplicative models, as the great majority of networks reached the maximum interpretability value. Also, for the threshold and xor models, networks having 500 and 1000 input SNPs were able to reach the maximum interpretability value. Thus, in the presence of marginal effects, CNNs can ignore the noisy SNPs and capture interacting ones.

When comparing with the NME datasets, the CNN had better performance detecting the interacting SNPs in ME datasets. Even in the presence of noise, these networks were able to have good interpretability and accuracy values across different models.

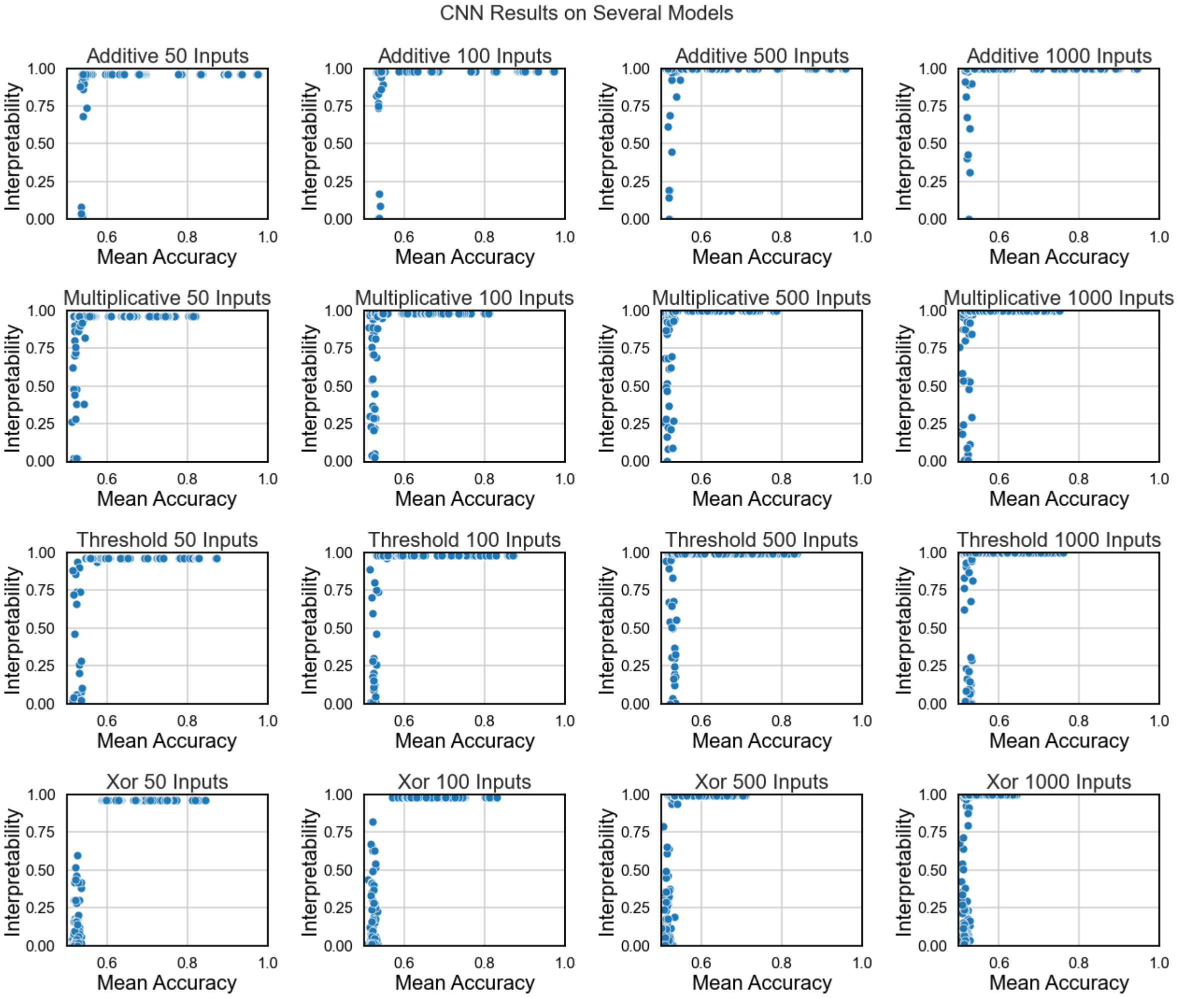


Figure 4.20: CNN interpretability analysis on ME datasets.

High-order interactions

Further, analysis on high order datasets is performed. Interactions of order four are considered and evaluated on the same epistasis models as pairwise interactions. Since a great variety of datasets was tested, a summary of all the results is presented in Figure 4.21.

By analysing the results a drop in performance is observed. CNN networks show lower accuracy and interpretability values across all the tested models. This performance drop was expected since a higher number of SNPs needs to be detected for interactions to be successfully discovered.

Additionally, as it was observed in pairwise interactions, the number of input SNPs does not have an impact on model interpretability. Networks having 500 and 1000 input SNPs were all to capture all four interacting SNPs. This highlights the ability of CNN to filter the noisy SNPs and select the interacting ones, both on pairwise and high order interactions.

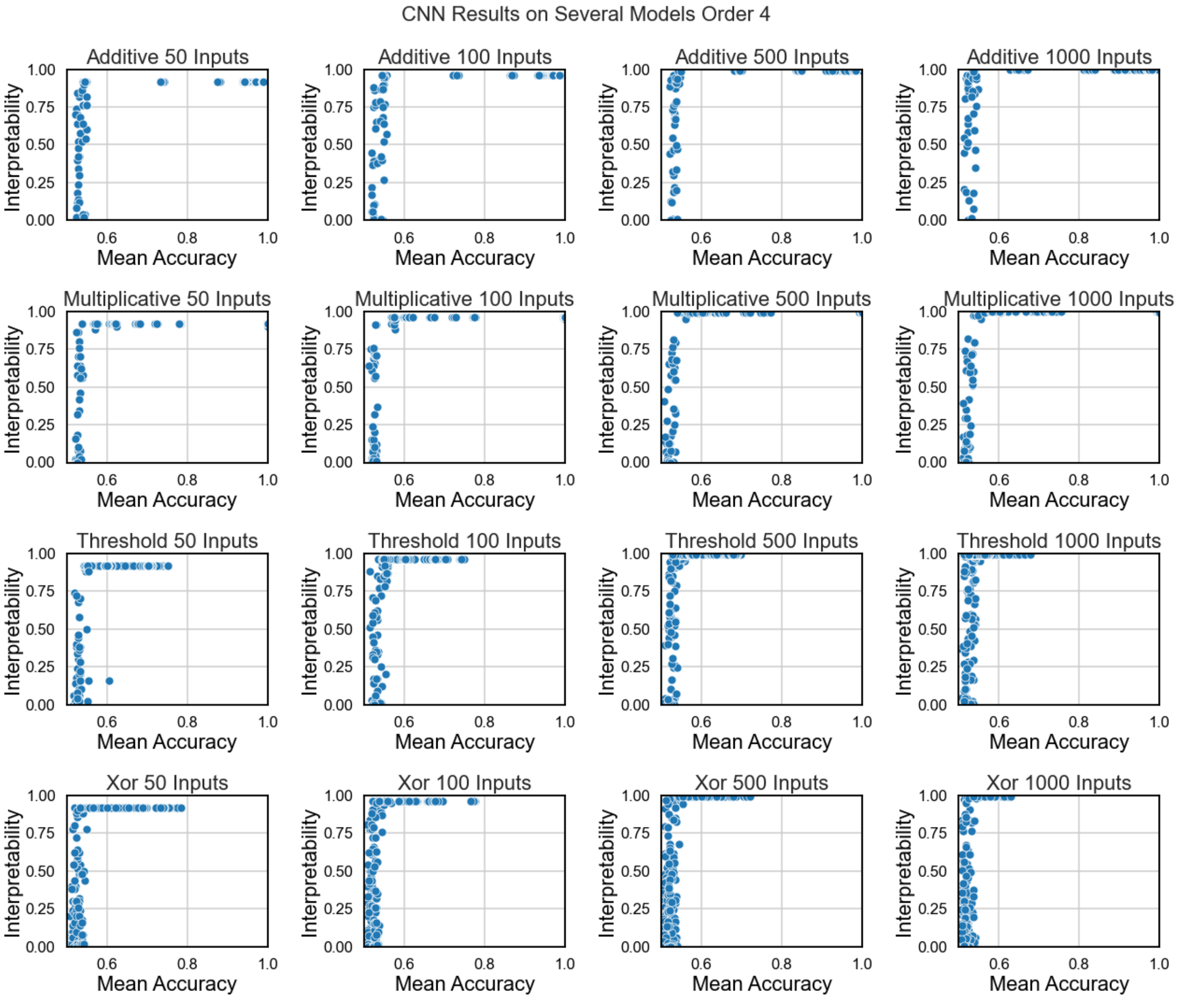


Figure 4.21: CNN interpretability analysis on ME datasets of order 4.

4.3.4 Heritability and MAF

In this section, CNN limitations in terms of heritability (h^2) and minor allele frequency (MAF) are evaluated.

The pruned search space from Table 4.14 is used to test datasets having MAF values of [0.05, 0.1, 0.2, 0.3, 0.4] and heritability values of [0.01, 0.05, 0.1, 0.15, 0.2, 0.4]. This way, it is ensured that different scenarios are covered and a detailed analysis performed. Also, to avoid errors in the results, values from 500 and 1000 SNP inputs are not considered, since the great majority of these networks failed to fit the data. The results are presented in Figure 4.22.

First, an analysis of the impact heritability (h^2) is 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 cannot be trusted. No threshold relation between accuracy and interpretability is observed, meaning that no accuracy value from above which all networks have maximum heritability is observed. With the increase of interpretability, network performance starts getting better. For $h^2 \geq 0.1$, in almost all datasets the accuracy threshold is observed (for $MAF = 0.2$ and $h^2 = 0.1$ is not), meaning that there is a minimum accuracy from which networks can be trusted. Also, the plot from Figure 4.23, confirms the better performance of networks for $h^2 \geq 0.1$. Additionally, if the outliers in Figure 4.23 are removed, for $h^2 \geq 0.2$ all networks managed to reach maximum interpretability.

Further an analysis of the impact of MAF in interpretability is performed. As it was observed in MLPs (Figure 4.12), it is difficult to derive conclusions from the box plot in Figure 4.23. There is a significant drop in performance from $MAF = 0.1$ to $MAF = 0.2$. A performance improvement is not observed with the increase of MAF. Still, if the outliers are removed for $MAF = 0.4$ all networks managed to achieve maximum interpretability.

4.3.5 Summary

In this section, the limitations of CNNs interpretability were studied on different epistasis scenarios.

First, the most cited works from state-of-the-art architectures were analysed and a search space for hyperparameter pruning was defined. Due to the increased amount of hyperparameters, the number of inputs was fixed to 50 SNPs to perform the pruning of the initial search space. As it was observed in MLP architectures, a threshold relation between interpretability and accuracy was detected, meaning that networks having an accuracy over the defined threshold will have maximum interpretability. The threshold values found were represented in Table 4.15.

NME datasets were used to perform the initial hyperparameter pruning since these datasets have increased difficulty. From the experimental results, CNNs having Softplus and Linear activation functions and higher kernel sizes of three revealed to be more frequent in networks having lower interpretability values. Also in [66], it was concluded that smaller kernel sizes have better performance in epistasis detection. The number of classification layers proved not to have an impact on model interpretability. Based on the previous conclusions, the Softplus and Linear activation functions were removed. Also, the kernel size of three was also removed. The remaining input values of 100, 500 and 1000 input SNPs were tested on the pruned search space and increasing the number of input SNPs caused the number of trustworthy networks to reduce, thus increasing the amount of noise in the input caused a drop in performance on CNN networks.

Next, tests on ME datasets were performed using the previously pruned search space. 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.

To conclude, the impact of MAF and heritability was evaluated. CNNs revealed to have better interpretability values in datasets having high heritability values ($h^2 \geq 0.2$). These results were expected since heritability controls the amount of information present in the dataset. Also, as observed in MLPs, for $h^2 = 0.3$ and $h^2 = 0.4$ all networks were able to achieve maximum interpretability. No conclusion could be derived on MAF and its influence on CNN interpretability.

The results on CNN networks revealed their ability to detect interacting SNPs in different epistasis scenarios. With the increase of heritability ($h^2 \geq 0.2$) and in the presence of marginal effects, these

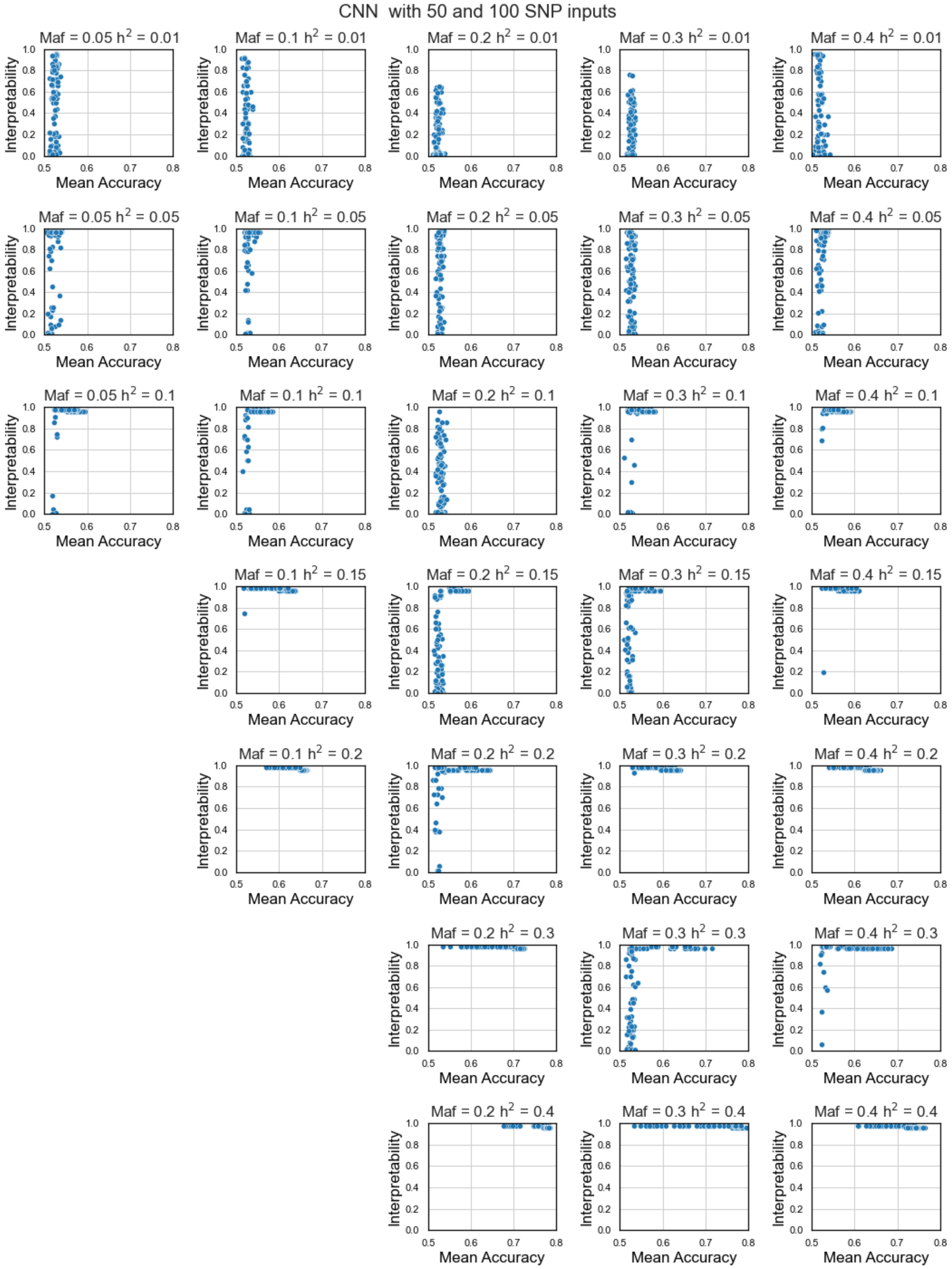


Figure 4.22: CNN interpretability analysis on several MAF and heritability values.

networks revealed to have maximum interpretability values. Additionally, in NME datasets with 500 and 1000 inputs, the threshold relation between interpretability and accuracy could be observed which suggests that these networks can filter noisy SNPs better than MLPs. Moreover, the ability to detect

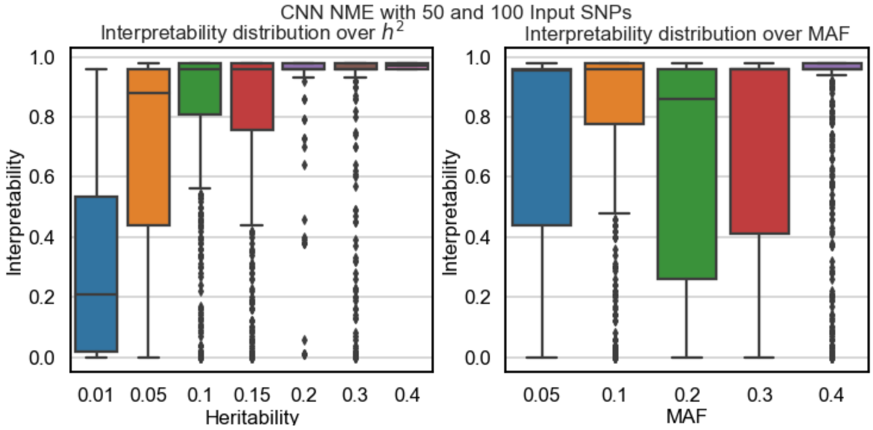


Figure 4.23: CNN boxplot analysis on the impact of MAF and heritability on interpretablity.

high-order interactions makes these architectures promising approaches.

4.4 Accuracy Threshold

The use of the interpretability performance measure from Equation 3.4 allows networks to be evaluated on how well the dataset is being learnt. This performance measure compares the most relevant features (input SNPs) identified by the model with the interacting SNPs in the datasets. Thus, to calculate interpretability it is necessary to know the interacting SNPs in the dataset.

In a real case scenario dataset, it is not possible to calculate interpretability since the interacting SNPs are unknown. It is necessary to establish a relation between interpretability and other performance measures which can be calculated in real datasets. Accuracy, F1 score, Precision, Sensitivity and AUC are performance measures that can be calculated on datasets whose interacting SNPs are unknown. It is necessary to establish a relation between these measures and interpretability so networks can be classified as trustworthy and sensitivity analysis performed to detect the relevance of each input SNP.

As noticed in the analysis on MLP and CNN architectures in Section 4.2.2 and Section 4.3.2 respectively, it is observed that interpretability is related to accuracy. In most cases, a threshold relation can be established between them, meaning that there is an accuracy value from above which networks are trustworthy. In that case, sensitivity analysis can be performed and the SNPs identified as most relevant are considered epistatic.

To detect the accuracy threshold value, for each MLP and CNN networks, the pruned search spaces in Table 4.8 and Table 4.14 are considered. The results on ME and NME datasets are grouped and analysed.

In Figure 4.24 the results for the MLP architectures are presented. It is observed that values with accuracies above 0.5478 have achieved maximum interpretability across all tested scenarios. The results for the CNN architectures are presented in Figure 4.25. The accuracy threshold for CNN networks is 0.54325. This means that any network with accuracy above 0.5482 achieves maximum or close to maximum interpretability values.

From the comparison of SNP interpretability with other performance measures, accuracy was revealed to be a good performance measure. This is an important step in epistasis detection since it allows 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 4.8) and CNN (4.14), if a network achieves an accuracy value over the defined thresholds (for MLP, an accuracy of 0.5478 and for CNN, an accuracy of 0.5482), networks can be interpreted and their decisions trusted.

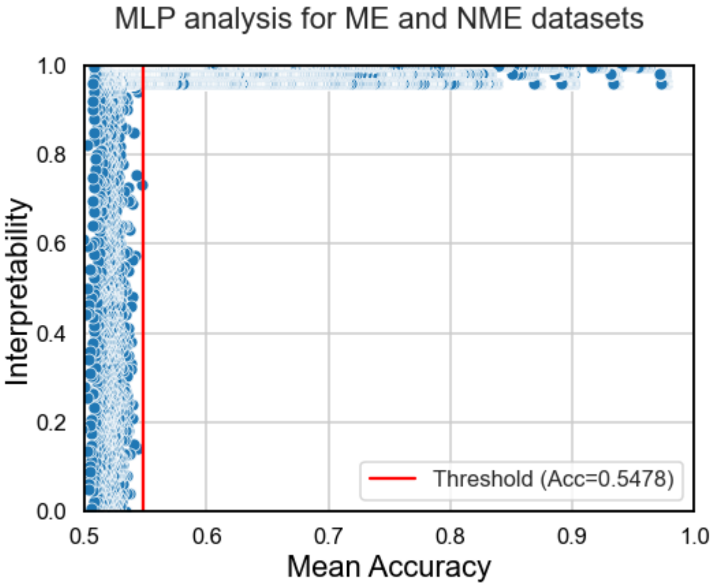


Figure 4.24: MLP accuracy threshold from which interpretability values increase.

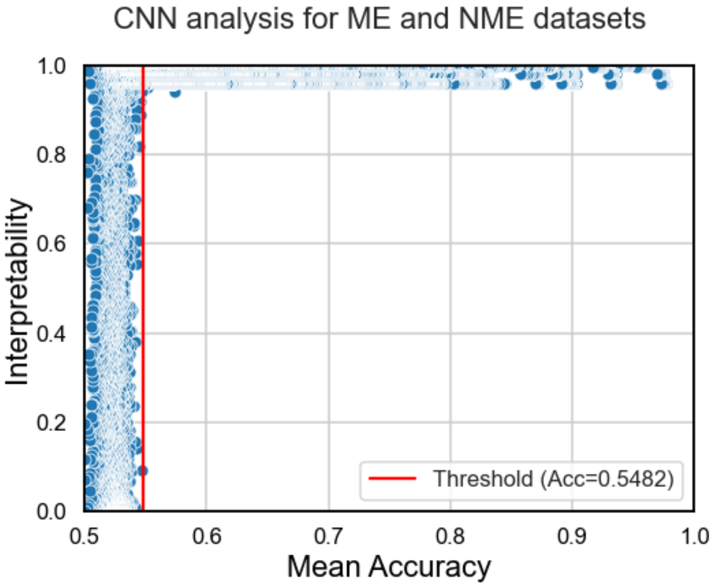


Figure 4.25: CNN accuracy threshold from which interpretability values increase.

4.5 Application On A Real Dataset

In this section, MLPs and CNNs are applied in a real Breast Cancer Dataset [86] to evaluate if trustworthy networks are trained and if sensitivity analysis captures 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 4.8 and Tables 4.14 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 4.4 were interpreted and the most relevant SNPs analysed.

To validate the results, a previous study on the same Breast Cancer dataset is considered [86]. In [87] an exhaustive search algorithm is 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 are considered and compared with the results obtained by MLP and CNN architectures.

First, the results were analysed on the MLP networks. In Table A.4 the results for MLP architectures are presented. As it is observed no networks were able to overcome the defined threshold of 0.5478 defined in the previous Section 4.4. Thus the trained networks cannot be trusted.

Next CNN networks are considered. In Table A.3 the results for the top CNN architectures are presented and it is observed that one network was able to reach the threshold value of 0.5482. Thus, this network can be trusted. In Figure 4.26 the plot displaying each SNP and the correspondent relevance is presented. The SNPs are sorted according to their relevance value. It is observed that interactions of order two have 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.

Additionally, the performance of all the tested CNN networks is analysed. In Figure 4.27 the index of the interacting SNPs of orders two, three and four in the ordered relevance vector is presented. It is confirmed that the increase of the mean accuracy causes networks to have better performance, with the best-performing ones having their accuracy values closer to the previously defined 0.5482 accuracy threshold. Also, SNP ”rs2010204” was considered to be the most relevant SNP in the majority of CNN (Figure 4.27) and MLP (Figure 4.28) networks. This type of behaviour might suggest the presence of marginal effects in SNP ”rs2010204” since networks correctly identify this SNP as interacting independently of the mean accuracy of the model. Furthermore, the presence of weaker marginal effects in SNP ”rs0504248” is also suggested since this SNP is always considered one of the most relevant in both MLP and CNN architectures. More detailed analysis on the results from MLP and CNN architectures is provided in Tables A.5, A.6 and A.7.

Moreover, in [87] a multi-objective analysis combining the K2 and Gini objective functions is applied

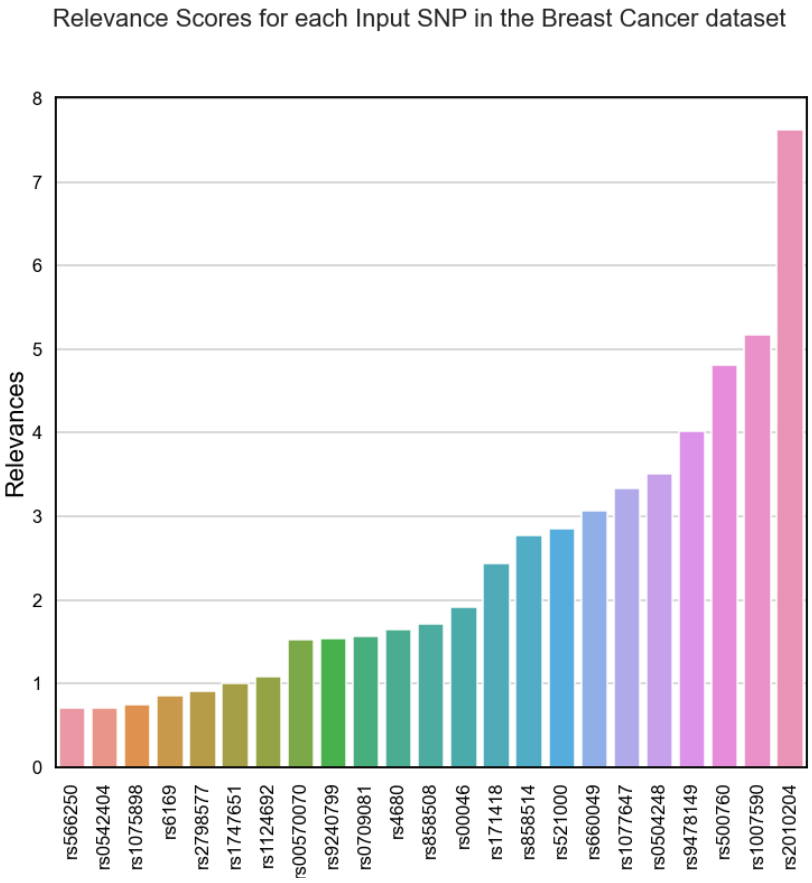


Figure 4.26: Relevance scores for each input SNP in Breast Cancer dataset. The SNPs are ordered increasingly according to the relevance score obtained by sensitivity analysis.

on the same Breast Cancer dataset, from which multiple solutions for different interaction orders were discovered. The interactions detected by the multi-objective method are compared with the most relevant SNPs detected by the CNN network, and the results are presented in Table 4.16. To simplify the analysis a numeric notation (Table A.5) is used to represent each SNP. The CNN network revealed to have a good performance on orders three and four, with the interacting SNPs being on the top 7 most relevant SNPs maximum. There is a significant performance drop on orders five and six caused by SNPs 11 and 17 which the model was unable to detect as most relevant, with SNP 11 (”rs0709081”) as being the thirteenth most relevant and SNP 17 (”rs00570070”) as being the fifteenth most relevant. The remaining SNPs are successfully detected as being relevant.

To conclude, the execution time of both multi-objective exhaustive search (Table 4.17) and the DL interpretability methods (Table 4.18) are discussed. Exhaustive search methods are more computationally demanding and tend to have higher execution time, whereas, neural networks are less computational demanding (non-exhaustive methods) and do not need to look for a specific interaction order, however, it is necessary to perform a grid search and train several networks in order to find a trustworthy network. If only the execution time of one correctly trained network is considered, DL interpretability methods highly overcome exhaustive search methods. The used Breast Cancer dataset represents a relatively small dataset with only 23 SNPs, in the presence of a dataset with a higher number of SNPs, exhaustive search methods would lead to a combinatorial explosion while DL interpretability methods would still be computationally feasible. Thus, despite having a higher grid search execution time (Table 4.18), DL interpretability methods are still promising approaches.

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, it is observed that if trained models achieve accuracy values over the defined thresholds, these can be trusted. This test on a real Breast Cancer dataset confirms the high potential of network interpretation methods to detect relevant SNPs or select them for further analysis.

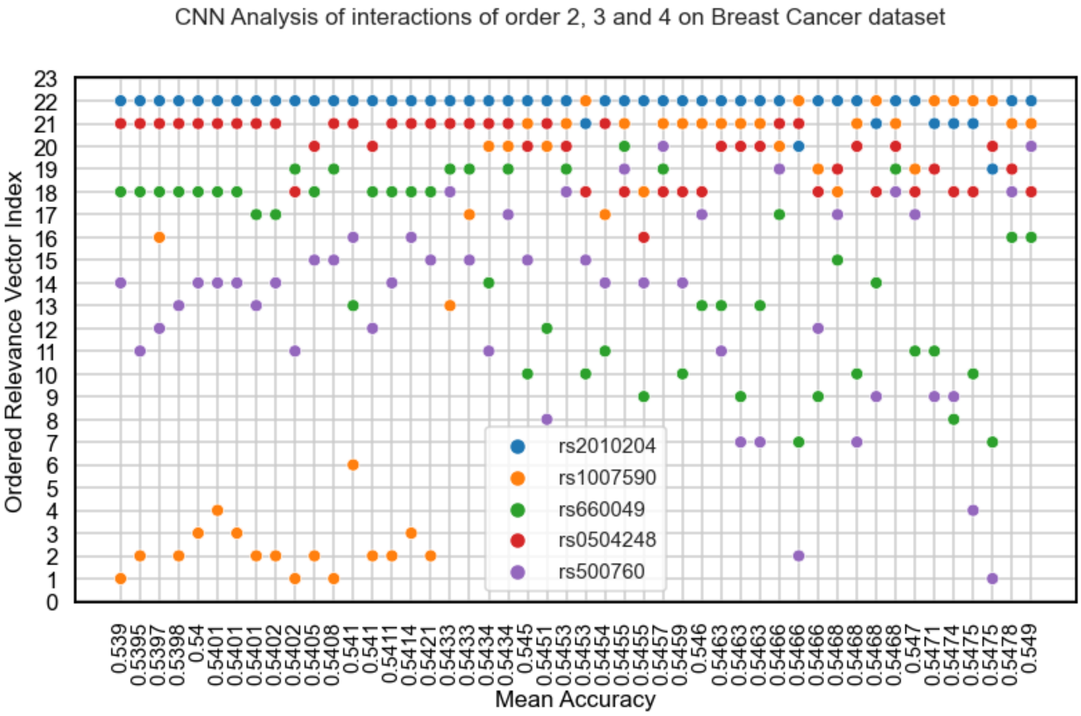


Figure 4.27: CNN index of interacting SNPs of orders 2, 3 and 4 in the increasingly ordered relevance vector for every CNN network. The higher indexes represent the SNPs with higher relevances, thus, networks having the interacting SNPs in higher indexes have better performance.

4.6 Summary

In this section, the experimental results of MLP and CNN architectures across different epistasis scenarios were presented and discussed.

The strategy used to evaluate both CNN and MLP architectures was similar. First, an analysis of the state-of-the-art most cited works was performed to define an initial search space for both CNN and MLP architectures. Due to the increased difficulty of NME datasets, these were used to prune the initial search. The reduced search space was evaluated on the remaining datasets ME datasets. It was confirmed that networks have less difficulty in detecting interacting SNPs in the presence of marginal

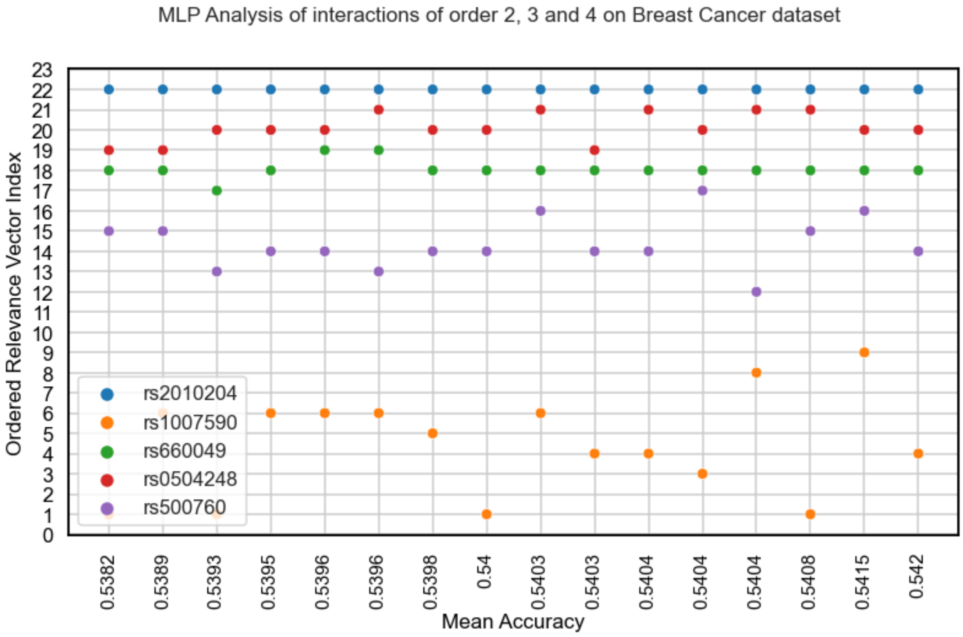


Figure 4.28: MLP index of interacting SNPs of orders 2, 3 and 4 in the increasingly ordered relevance vector for every MLP network. The higher indexes represent the SNPs with higher relevances, thus, networks having the interacting SNPs in higher indexes have better performance.

Table 4.16: CNN performance analysis for multi-objective analysis on Breast Cancer dataset. The Top Most Relevant SNPs represent the number of most relevant SNPs to be considered in order to include all the interacting SNPs. The notation used to represent the SNPs is presented in Table A.5.

Order	Interacting SNPs	Top Most Relevant SNPs
3	[3, 13, 22]	4
3	[3, 16, 22]	6
4	[3, 8, 13, 22]	5
4	[3, 13, 16, 22]	6
4	[3, 13, 16, 18]	6
4	[3, 14, 18, 22]	7
4	[3, 13, 14, 22]	7
5	[3, 14, 16, 18, 22]	7
5	[3, 13, 14, 16, 18]	7
5	[3, 12, 14, 16, 18]	7
5	[3, 13, 14, 20, 22]	9
5	[3, 11, 13, 20, 22]	13
5	[3, 11, 16, 18, 22]	13
5	[3, 11, 13, 18, 22]	13
5	[3, 13, 16, 17, 18]	15
5	[3, 13, 17, 18, 22]	15
5	[3, 13, 17, 20, 22]	15
6	[3, 13, 16, 20, 21, 22]	9
6	[3, 12, 13, 14, 20, 22]	9
6	[2, 3, 13, 20, 21, 22]	10
6	[2, 3, 13, 18, 20, 22]	10
6	[3, 11, 13, 18, 20, 22]	13
6	[3, 11, 13, 14, 18, 22]	13
6	[3, 11, 13, 16, 18, 22]	13
6	[3, 10, 13, 16, 17, 18]	15
6	[3, 13, 17, 18, 20, 22]	15
6	[2, 3, 13, 17, 20, 22]	15
6	[3, 13, 17, 20, 21, 22]	15

effects and that increasing the order of the interactions causes a decrease in model performance.

Further, a detailed analysis of the impact of MAF and heritability was performed. In both networks, it was observed that increasing the heritability causes the network performance to improve. Also, for

Table 4.17: Multi-objective exhaustive search execution time.

Order	Execution Time
3	0h 20min
4	0h 10min
5	0h 50min
6	2h 45min

Table 4.18: DL interpretability execution time.

Architecture	Execution Time (one network)	Execution Time (grid search)
MLP	0h 3min 48sec	1h 19min 10sec
CNN	0h 5min 33sec	4h 26min 48sec

lower heritability values ($h^2 = 0.01$ and $h^2 = 0.05$), none of the networks was able to produce reliable networks, only with the increase of heritability did the results started to improve. No relation between MAF and interpretability could be observed.

Since the interpretability performance measure cannot be calculated on real datasets, a threshold relation between interpretability and accuracy was established. From the results across all datasets, an accuracy threshold value of 0.5478 and 0.5482 for CNN and MLP networks was identified. Thus, any network with accuracy values over the defined thresholds can be trusted and interpreted.

To conclude and confirm the findings, tests on a real Breast Cancer dataset were performed. The results were compared with a recent study that performed an exhaustive analysis on the same dataset. From the results, only one CNN was able to pass the desired threshold. The identified SNPs in the top 30% most relevant, belonged to interactions of orders two three and four. These are positive results since they confirm that the networks can detect valid SNPs as most relevant. Additionally, it highlights the possibility of using DL networks as filters to select relevant SNPs and perform exhaustive analysis on the selected SNPs.

Chapter 5

Conclusions and Future Work

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 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.

First, an analysis of the most recent works using deep learning in epistasis detection revealed that CNNs and MLPs were the most used types of DL models. The network architectures from the most cited papers [66–69, 73] were selected and the hyperparameters evaluated, allowing the definition of manageable search spaces for both CNNs and MLPs.

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. These relevance scores were calculated throughout the entire dataset and added. The final result was a vector of increasingly ordered SNPs according to their relevance values. 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 the experimental stage, CNN and MLP networks were evaluated under a wide variety of epistasis scenarios. Due to the increased difficulty of datasets without marginal effects, these were used to prune the initial search spaces. The pruned search space was applied on datasets having marginal effects, including both pairwise and higher-order datasets. Additionally, a detailed analysis of the impact of heritability and minor allele frequency MAF was performed.

During the experimental stage, the relation between interpretability and other performance measures (accuracy, F1 score, precision, sensitivity and AUC) was compared, and it was concluded that there was a threshold relation between interpretability and accuracy. Every network having accuracy above the threshold would have maximum interpretability. Thus, accuracy and interpretability were selected as the two performance measures to evaluate networks.

It was confirmed that both CNN and MLP networks have less difficulty detecting interacting SNPs in the presence of marginal effects, achieving higher accuracy and interpretability values. When increasing to higher-order interactions (order 4), there was a drop in performance. This was caused by the fact that networks need to detect a higher number of SNPs as being the most relevant to achieve maximum interpretability. Still, the results were very encouraging since several networks were able to achieve maximum interpretability. For NME datasets, limitations regarding heritability were detected. For lower heritability values ($h^2 = 0.01$ and $h^2 = 0.05$) networks did not fit the data, however, with the increase of heritability ($h^2 > 0.1$) networks started having an accuracy threshold from which maximum interpretability is achieved. No conclusions on the impact of MAF could be made.

In a real epistasis dataset, 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. A CNN network able to pass the accuracy threshold was selected and the relevance values for each SNP were evaluated. The results were compared with a previous study in which an exhaustive search method was applied on the same datasets. The network 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%).

To conclude, with this dissertation the identified gap in the state-of-art methods regarding network interpretability was filled. The developed methodology allows interpreting network decisions for detecting the interacting SNPs and evaluating network limitations in terms of interpretability. These results allowed the definition of an accuracy threshold from which networks can be trusted. Given the result on a real Breast Cancer dataset, it is believed that this dissertation consists of valuable work to overcome the black-box nature of deep learning models.

5.1 Future Work

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 [54]. 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 [10, 83], 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.

Bibliography

[1] I. C. Gray, D. A. Campbell, and N. K. Spurr. Single nucleotide polymorphisms as tools in human genetics. *Human molecular genetics*, 9(16):2403–2408, 2000.

[2] G. P. Consortium et al. A global reference for human genetic variation. *Nature*, 526(7571):68–74, 2015.

[3] 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*, 101(1):5–22, 2017.

[4] D. S. W. Ho, W. Schierding, M. Wake, R. Saffery, and J. O'Sullivan. Machine learning snp based prediction for precision medicine. *Frontiers in Genetics*, 10, 2019.

[5] R. J. Klein, C. Zeiss, E. Y. Chew, J.-Y. Tsai, R. S. Sackler, C. Haynes, A. K. Henning, J. P. SanGiovanni, S. M. Mane, S. T. Mayne, et al. Complement factor h polymorphism in age-related macular degeneration. *Science*, 308(5720):385–389, 2005.

[6] W. T. C. C. Consortium et al. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447(7145):661, 2007.

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

[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*, 14(2):251–260, 2013.

[9] 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*, 15(2):599–612, 2016.

[10] G. Eraslan, Ž. Avsec, J. Gagneur, and F. J. Theis. Deep learning: new computational modelling techniques for genomics. *Nature Reviews Genetics*, 20(7):389–403, 2019.

[11] M. Pérez-Enciso and L. M. Zingaretti. A guide on deep learning for complex trait genomic prediction. *Genes*, 10(7):553, 2019.

[12] J. H. Moore and M. D. Ritchie. The challenges of whole-genome approaches to common diseases. *Jama*, 291(13):1642–1643, 2004.

[13] D. Harold, R. Abraham, P. Hollingworth, R. Sims, A. Gerrish, M. L. Hamshere, J. S. Pahwa, V. Moskvina, K. Dowzell, A. Williams, et al. Genome-wide association study identifies variants at clu and picalm associated with alzheimer's disease. *Nature genetics*, 41(10):1088, 2009.

[14] J. C. Barrett, S. Hansoul, D. L. Nicolae, J. H. Cho, R. H. Duerr, J. D. Rioux, S. R. Brant, M. S. Silverberg, K. D. Taylor, M. M. Barmada, et al. Genome-wide association defines more than 30 distinct susceptibility loci for crohn's disease. *Nature genetics*, 40(8):955, 2008.

[15] G. Thomas, K. B. Jacobs, M. Yeager, P. Kraft, S. Wacholder, N. Orr, K. Yu, N. Chatterjee, R. Welch, A. Hutchinson, et al. Multiple loci identified in a genome-wide association study of prostate cancer. *Nature genetics*, 40(3):310, 2008.

[16] W. Bateson. Mendel's principles of heredity. cambridge university press. *März 1909; 2nd Impr*, 3: 1913, 1909.

[17] R. A. Fisher. Xv.—the correlation between relatives on the supposition of mendelian inheritance. *Earth and Environmental Science Transactions of the Royal Society of Edinburgh*, 52(2):399–433, 1919.

[18] P. C. Phillips. Epistasis—the essential role of gene interactions in the structure and evolution of genetic systems. *Nature Reviews Genetics*, 9(11):855–867, 2008.

[19] V. J. Vieland and J. Huang. Two-locus heterogeneity cannot be distinguished from two-locus epistasis on the basis of affected-sib-pair data. *The American Journal of Human Genetics*, 73(2): 223–232, 2003.

[20] 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*, 7:13497–13509, 2019.

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

[22] X. Jiang, R. E. Neapolitan, M. M. Barmada, and S. Visweswaran. Learning genetic epistasis using bayesian network scoring criteria. *BMC bioinformatics*, 12(1):89, 2011.

[23] 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*, 5(1):1–14, 2012.

[24] 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*, 21(1): 1–9, 2020.

[25] J. Marchini, P. Donnelly, and L. R. Cardon. Genome-wide strategies for detecting multiple loci that influence complex diseases. *Nature genetics*, 37(4):413–417, 2005.

[26] X. Wan, C. Yang, Q. Yang, H. Xue, X. Fan, N. L. Tang, and W. Yu. Boost: A fast approach to detecting gene-gene interactions in genome-wide case-control studies. *The American Journal of Human Genetics*, 87(3):325–340, 2010.

[27] L. S. Yung, C. Yang, X. Wan, and W. Yu. Gboost: a gpu-based tool for detecting gene–gene interactions in genome–wide case control studies. *Bioinformatics*, 27(9):1309–1310, 2011.

[28] G. Yang, W. Jiang, Q. Yang, and W. Yu. Pboost: a gpu-based tool for parallel permutation tests in genome-wide association studies. *Bioinformatics*, 31(9):1460–1462, 2015.

[29] X. Zhang, S. Huang, F. Zou, and W. Wang. Team: efficient two-locus epistasis tests in human genome-wide association study. *Bioinformatics*, 26(12):i217–i227, 2010.

[30] A. Gyenesei, J. Moody, C. A. Semple, C. S. Haley, and W.-H. Wei. High-throughput analysis of epistasis in genome-wide association studies with biforce. *Bioinformatics*, 28(15):1957–1964, 2012.

[31] D. Gola, J. M. Mahachie John, K. Van Steen, and I. R. König. A roadmap to multifactor dimensionality reduction methods. *Briefings in bioinformatics*, 17(2):293–308, 2016.

[32] J. H. Moore, F. W. Asselbergs, and S. M. Williams. Bioinformatics challenges for genome-wide association studies. *Bioinformatics*, 26(4):445–455, 2010.

[33] G. Shmueli et al. To explain or to predict? *Statistical science*, 25(3):289–310, 2010.

[34] B. A. McKinney, D. M. Reif, M. D. Ritchie, and J. H. Moore. Machine learning for detecting gene-gene interactions. *Applied bioinformatics*, 5(2):77–88, 2006.

[35] 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*, 2013, 2013.

[36] K. Van Steen. Travelling the world of gene–gene interactions. *Briefings in bioinformatics*, 13(1): 1–19, 2012.

[37] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[38] R. Jiang, W. Tang, X. Wu, and W. Fu. A random forest approach to the detection of epistatic interactions in case-control studies. *BMC bioinformatics*, 10(1):S65, 2009.

[39] D. F. Schwarz, I. R. König, and A. Ziegler. On safari to random jungle: a fast implementation of random forests for high-dimensional data. *Bioinformatics*, 26(14):1752–1758, 2010.

[40] L. De Lobel, P. Geurts, G. Baele, F. Castro-Giner, M. Kogevinas, and K. Van Steen. A screening methodology based on random forests to improve the detection of gene–gene interactions. *European journal of human genetics*, 18(10):1127–1132, 2010.

[41] H.-Y. Lin, Y. Ann Chen, Y.-Y. Tsai, X. Qu, T.-S. Tseng, and J. Y. Park. Trm: A powerful two-stage machine learning approach for identifying snp-snp interactions. *Annals of human genetics*, 76(1): 53–62, 2012.

[42] M. Yoshida and A. Koike. Snpinterforest: a new method for detecting epistatic interactions. *BMC bioinformatics*, 12(1):469, 2011.

[43] X. Chen and H. Ishwaran. Random forests for genomic data analysis. *Genomics*, 99(6):323–329, 2012.

[44] Q. Pan, T. Hu, J. D. Malley, A. S. Andrew, M. R. Karagas, and J. H. Moore. Supervising random forest using attribute interaction networks. In *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, pages 104–116. Springer, 2013.

[45] A. Staiano, M. D. Di Taranto, E. Bloise, M. N. D'Agostino, A. D'Angelo, G. Marotta, M. Gentile, F. Jossa, A. Iannuzzi, P. Rubba, et al. Investigation of single nucleotide polymorphisms associated to familial combined hyperlipidemia with random forests. In *Neural nets and surroundings*, pages 169–178. Springer, 2013.

[46] S.-H. Chen, J. Sun, L. Dimitrov, A. R. Turner, T. S. Adams, D. A. Meyers, B.-L. Chang, S. L. Zheng, H. Grönberg, J. Xu, et al. A support vector machine approach for detecting gene-gene interaction. *Genetic Epidemiology: The Official Publication of the International Genetic Epidemiology Society*, 32(2):152–167, 2008.

[47] Y. Shen, Z. Liu, and J. Ott. Detecting gene-gene interactions using support vector machines with l 1 penalty. In *2010 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)*, pages 309–311. IEEE, 2010.

[48] S. Marvel and A. Motsinger-Reif. Grammatical evolution support vector machines for predicting human genetic disease association. In *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*, pages 595–598, 2012.

[49] H. Zhang, H. Wang, Z. Dai, M.-s. Chen, and Z. Yuan. Improving accuracy for cancer classification with a new algorithm for genes selection. *BMC bioinformatics*, 13(1):298, 2012.

[50] H.-J. Ban, J. Y. Heo, K.-S. Oh, and K.-J. Park. Identification of type 2 diabetes-associated combination of snps using support vector machine. *BMC genetics*, 11(1):26, 2010.

[51] Y. Wang, X. Liu, K. Robbins, and R. Rekaya. Antepiseeker: detecting epistatic interactions for case-control studies using a two-stage ant colony optimization algorithm. *BMC research notes*, 3 (1):117, 2010.

[52] Y. Wang, X. Liu, and R. Rekaya. Antepiseeker2. 0: extending epistasis detection to epistasis-associated pathway inference using ant colony optimization. *Nature Precedings*, pages 1–1, 2012.

[53] J. Shang, J. Zhang, X. Lei, Y. Zhang, and B. Chen. Incorporating heuristic information into ant colony optimization for epistasis detection. *Genes & Genomics*, 34(3):321–327, 2012.

[54] 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*, 31(5):634–641, 2015.

[55] C. S. Greene, B. C. White, and J. H. Moore. Ant colony optimization for genome-wide genetic analysis. In *International Conference on Ant Colony Optimization and Swarm Intelligence*, pages 37–47. Springer, 2008.

[56] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[57] J. Zhou and O. G. Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10):931–934, 2015.

[58] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 33(8):831–838, 2015.

[59] A. A. Motsinger-Reif, S. M. Dudek, L. W. Hahn, and M. D. Ritchie. Comparison of approaches for machine-learning optimization of neural networks for detecting gene-gene interactions in genetic epidemiology. *Genetic Epidemiology: The Official Publication of the International Genetic Epidemiology Society*, 32(4):325–340, 2008.

[60] M. D. Ritchie, B. C. White, J. S. Parker, L. W. Hahn, and J. H. Moore. Optimization of neural network architecture using genetic programming improves detection and modeling of gene-gene interactions in studies of human diseases. *BMC bioinformatics*, 4(1):28, 2003.

[61] A. A. Motsinger, S. L. Lee, G. Mellick, and M. D. Ritchie. Gpnn: Power studies and applications of a neural network method for detecting gene-gene interactions in studies of human disease. *BMC bioinformatics*, 7(1):39, 2006.

[62] M. D. Ritchie, A. A. Motsinger, W. S. Bush, C. S. Coffey, and J. H. Moore. Genetic programming neural networks: A powerful bioinformatics tool for human genetics. *Applied Soft Computing*, 7(1): 471–479, 2007.

[63] N. E. Hardison and A. A. Motsinger-Reif. The power of quantitative grammatical evolution neural networks to detect gene-gene interactions. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 299–306, 2011.

[64] Y. Tomita, S. Tomida, Y. Hasegawa, Y. Suzuki, T. Shirakawa, T. Kobayashi, and H. Honda. Artificial neural network approach for selection of susceptible single nucleotide polymorphisms and construction of prediction model on childhood allergic asthma. *BMC bioinformatics*, 5(1):120, 2004.

[65] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[66] P. Bellot, G. de los Campos, and M. Pérez-Enciso. Can deep learning improve genomic prediction of complex human traits? *Genetics*, 210(3):809–819, 2018.

[67] S. Uppu, A. Krishna, and R. P. Gopalan. A deep learning approach to detect snp interactions. *JSW*, 11(10):965–975, 2016.

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

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

[70] 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*, 69(1):138–147, 2001.

[71] 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)*, pages 1–8. IEEE, 2018.

[72] S. Uppu and A. Krishna. Convolutional model for predicting snp interactions. In *International Conference on Neural Information Processing*, pages 127–137. Springer, 2018.

[73] 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)*, pages 226–232. IEEE, 2018.

[74] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[75] P. Fergus, A. Montanez, B. Abdulaimma, P. Lisboa, C. Chalmers, and B. Pineles. Utilising deep learning and genome wide association studies for epistatic-driven preterm birth classification in african-american women. *IEEE/ACM transactions on computational biology and bioinformatics*, 2018.

[76] B. Abdulaimma, P. Fergus, and C. Chalmers. Extracting epistatic interactions in type 2 diabetes genome-wide data using stacked autoencoder. *arXiv preprint arXiv:1808.09517*, 2018.

[77] P. Waldmann. Approximate bayesian neural networks in genomic prediction. *Genetics Selection Evolution*, 50(1):70, 2018.

[78] C. A. C. Montañez, P. Fergus, C. Chalmers, N. A. H. Malim, B. Abdulaimma, D. Reilly, and F. Falciani. Saerma: Stacked autoencoder rule mining algorithm for the interpretation of epistatic interactions in gwas for extreme obesity. *arXiv preprint arXiv:1908.10166*, 2019.

[79] S. Uppu and A. Krishna. A deep hybrid model to detect multi-locus interacting snps in the presence of noise. *International journal of medical informatics*, 119:134–151, 2018.

[80] G. M. Clarke, C. A. Anderson, F. H. Pettersson, L. R. Cardon, A. P. Morris, and K. T. Zondervan. Basic statistical analysis in genetic case-control studies. *Nature protocols*, 6(2):121–133, 2011.

[81] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(Feb):281–305, 2012.

[82] J. Liphardt. Deepevolve, 2017. URL `https://github.com/jliphard/DeepEvolve/`.

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

[84] F. Chollet et al. Keras: Deep learning library for theano and tensorflow. *URL: https://keras. io/k*, 7 (8):T1, 2015.

[85] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation OSDI 16)*, pages 265–283, 2016.

[86] 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*, 10(2):361–371, 2013.

[87] 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*, pages 613–628. Springer, 2020.

Appendix A

Experimental Results Appendix

Table A.1: List of architectures from pruned search space for MLP with input size N.

Architecture Nº	Nº Inputs	Neurons per Layer	Activation Function
0	N	[32]	elu
1	N	[32]	tanh
2	N	[64]	elu
3	N	[64]	tanh
4	N	[32 32]	elu
5	N	[32 32]	tanh
6	N	[64 64]	elu
7	N	[64 64]	tanh
8	N	[32 32 32]	elu
9	N	[32 32 32]	tanh
10	N	[64 64 64]	elu
11	N	[64 64 64]	tanh
12	N	[32 32 32 32 32]	elu
13	N	[32 32 32 32 32]	tanh
14	N	[64 64 64 64 64]	elu
15	N	[64 64 64 64 64]	tanh

Table A.2: List of architectures from pruned search space for MLP with Input size N.

Architecture Nº	Nº Inputs	Neurons per Conv. Layer	Kernel Size	Neurons per Class. Layer	Activation Function
0	N	[64 64]	2	[64 64 64]	elu
1	N	[64]	2	[32 32 32]	elu
2	N	[64]	1	[32 32 32]	elu
3	N	[64]	2	[64 64 64]	elu
4	N	[32]	1	[32 32 32]	elu
5	N	[16]	1	[32 32 32]	elu
6	N	[32 32]	2	[32 32 32]	elu
7	N	[64 64]	2	[32 32 32]	elu
8	N	[16]	2	[32 32 32]	elu
9	N	[32]	1	[64 64 64]	elu
10	N	[64]	1	[64 64 64]	elu
11	N	[16]	2	[64 64 64]	elu
12	N	[16]	1	[64 64 64]	elu
13	N	[32]	2	[64 64 64]	elu
14	N	[32]	2	[32 32 32]	elu
15	N	[32 32]	2	[64 64 64]	elu
16	N	[16 16]	2	[32 32 32]	elu
17	N	[16 16]	2	[64 64 64]	elu
18	N	[64 64]	1	[32 32 32]	elu
19	N	[16]	1	[32 32 32]	relu
20	N	[64 64]	1	[64 64 64]	elu
21	N	[64 64]	2	[32 32 32]	relu
22	N	[16]	1	[64 64 64]	relu
23	N	[32 32]	1	[64 64 64]	elu
24	N	[32 32]	2	[64 64 64]	relu
25	N	[64 64]	2	[64 64 64]	relu
26	N	[16 16]	1	[64 64 64]	elu
27	N	[16 16]	2	[64 64 64]	relu
28	N	[32 32]	1	[32 32 32]	elu
29	N	[16 16]	2	[32 32 32]	relu
30	N	[16]	2	[64 64 64]	relu
31	N	[32]	1	[32 32 32]	relu
32	N	[32]	1	[64 64 64]	relu
33	N	[64]	1	[64 64 64]	relu
34	N	[32]	2	[32 32 32]	relu
35	N	[32 32]	1	[32 32 32]	relu
36	N	[32 32]	2	[32 32 32]	relu
37	N	[16]	2	[32 32 32]	relu
38	N	[64]	1	[32 32 32]	relu
39	N	[16 16]	1	[32 32 32]	relu
40	N	[16 16]	1	[32 32 32]	elu
41	N	[32]	2	[64 64 64]	relu
42	N	[64 64]	1	[64 64 64]	relu
43	N	[16 16]	1	[64 64 64]	relu
44	N	[64]	2	[32 32 32]	relu
45	N	[32 32]	1	[64 64 64]	relu
46	N	[64 64]	1	[32 32 32]	relu
47	N	[64]	2	[64 64 64]	relu

Table A.3: Results for CNN performance in Breast Cancer dataset.

Architecture Nº	Mean Accuracy	Mean F1 Score	Mean Precision	Mean Sensivity	Mean ROC
47	**0.548999989**	0.5614247	0.546610452	0.587	0.5596069
44	0.547800004	0.562988267	0.544870607	0.5886	0.5600641
46	0.547500001	0.572753655	0.543713443	0.608	0.5607847
43	0.54749999	0.562886919	0.545141847	0.5892	0.5593997
45	0.547399997	0.554062193	0.547695084	0.564	0.5617455
42	0.547099996	0.566415149	0.543555468	0.5954	0.5607584
41	0.547000003	0.553631931	0.545614369	0.5642	0.5579454
40	0.546800017	0.556635987	0.545102936	0.5706	0.5573578
38	0.546800005	0.548023698	0.54807403	0.559	0.5586166
39	0.546800005	0.526805032	0.551991567	0.5166	0.5596866
37	0.546799994	0.552501497	0.546245146	0.5632	0.5578438
36	0.546600008	0.559686989	0.544547676	0.5816	0.5578752
35	0.546600008	0.54441289	0.547264074	0.5474	0.5601114
34	0.546600008	0.566225401	0.543033874	0.596	0.558551
33	0.546300006	0.569135707	0.542316047	0.6014	0.5586052
32	0.546300006	0.549109869	0.5458669	0.5608	0.5580873
31	0.546300006	0.556971624	0.54449919	0.5738	0.5589801
30	0.546000004	0.552477509	0.545157596	0.564	0.558834
29	0.545900011	0.541621729	0.547967627	0.5472	0.5573473
28	0.545700002	0.542915463	0.546458397	0.5422	0.5586151
27	0.545499992	0.556765949	0.543278203	0.571	0.5579312
26	0.545499992	0.565465497	0.542105491	0.5952	0.5584244
25	0.545400012	0.566085361	0.542022855	0.5992	0.5577591
24	0.545300007	0.552608821	0.543892373	0.563	0.5593616
23	0.545300007	0.544986803	0.545906972	0.55	0.5588722
22	0.545099998	0.556797885	0.54357194	0.5776	0.5566719
21	0.544999993	0.545907447	0.544909976	0.5492	0.5586792
20	0.543400002	0.536172114	0.544996188	0.5312	0.5537663
19	0.543400002	0.555352342	0.541477192	0.572	0.555798
18	0.543299997	0.539718954	0.544145937	0.5368	0.5537769
17	0.543299997	0.549089128	0.542304393	0.5564	0.5553821
16	0.5421	0.549041554	0.541005056	0.5582	0.5520686
15	0.541399992	0.550144209	0.539895587	0.5612	0.5513278
14	0.541099989	0.56353914	0.537670745	0.5978	0.5502525
13	0.541000009	0.542027392	0.540767966	0.5444	0.5502542
12	0.540999985	0.550576633	0.539397671	0.5626	0.5505221
11	0.540799999	0.552899057	0.538693697	0.569	0.550173
10	0.540499985	0.541986424	0.540377523	0.544	0.5503842
9	0.540199995	0.555177841	0.537560964	0.5766	0.5503457
8	0.540199995	0.547598718	0.538980368	0.5566	0.5492509
7	0.540100002	0.545184356	0.53949522	0.5526	0.549862
6	0.540100002	0.556424921	0.53746122	0.5776	0.5494439
5	0.540100002	0.5392379	0.540250023	0.5384	0.5511745
4	0.54000001	0.540099711	0.539840499	0.5422	0.5508388
3	0.5398	0.54756209	0.538388737	0.5576	0.5503328
2	0.539700007	0.545604687	0.53873092	0.5528	0.5513138
1	0.539499998	0.547317343	0.538324072	0.5576	0.5502254
0	0.538999999	0.543632588	0.538246403	0.5492	0.5497763

Table A.4: Results for MLP performance in Breast Cancer dataset.

Architecture Nº	Mean Accuracy	Mean F1 Score	Mean Precision	Mean Sensivity	Mean ROC
10	0.541999996	0.554632817	0.539788492	0.5712	0.5500945
6	0.541499984	0.559583504	0.538426949	0.5838	0.5500072
9	0.540799999	0.556053152	0.538247805	0.5758	0.5485336
0	0.540399992	0.546184373	0.539454197	0.5548	0.5496036
1	0.540399992	0.540528645	0.540438812	0.5426	0.5495486
7	0.540399992	0.547150518	0.539314711	0.556	0.5479382
14	0.5403	0.547435414	0.539818467	0.5614	0.5494495
2	0.540299988	0.544106141	0.539707787	0.5494	0.5506748
15	0.539999998	0.556444898	0.537312634	0.578	0.5488203
8	0.5398	0.55140004	0.537774902	0.5666	0.5500327
3	0.539600003	0.544388886	0.539684066	0.5542	0.5493491
5	0.539599991	0.555037989	0.537133709	0.575	0.5489614
11	0.539499986	0.549474499	0.537858818	0.5618	0.5491698
12	0.539299989	0.543673785	0.538931957	0.5504	0.549536
4	0.538900006	0.534608776	0.540138704	0.5462	0.5486338
13	0.538200009	0.549086502	0.536310612	0.5638	0.5480041

Table A.5: Simplified SNP labelling in Breast Cancer dataset. Interactions of order two are presented by SNPs 3 and 22, interactions of order three by SNPs 3, 16 and 22 and interactions of order four by SNPs 3, 13, 16 and 18.

SNP Nº	SNP Name
0	rs6169
1	rs4680
2	rs00046
3	rs2010204
4	rs1124692
5	rs0542404
6	rs2798577
7	rs1747651
8	rs1077647
9	rs1075898
10	rs9240799
11	rs0709081
12	rs9478149
13	rs0504248
14	rs521000
15	rs566250
16	rs660049
17	rs00570070
18	rs500760
19	rs858508
20	rs171418
21	rs858514
22	rs1007590

Table A.6: MLP ordered SNP relevance values in Breast Cancer dataset. The index in the increasingly ordered relevance vector of the interacting SNPs (orders two, three and four) is presented, with higher indexes representing higher relevance values. The notation used to represent the SNPs is presented in Table A.5.

				Index		
Architecture Nº	Increasingly Ordered Relevance Vector	SNP 3	SNP 22	SNP 16	SNP 13	SNP 18
10	[6 4 5 7 22 2 8 9 1 15 11 10 17 0 18 14 21 19 16 20 13 12 3]	22	4	18	20	14
6	[6 7 5 2 4 9 8 15 1 22 0 11 14 17 21 10 18 19 16 20 13 12 3]	22	9	18	20	16
9	[6 22 2 4 5 8 7 9 15 0 1 11 17 10 14 18 19 21 16 12 20 13 3]	22	1	18	21	15
0	[5 7 6 4 22 8 9 2 15 1 17 10 11 0 18 14 21 19 16 20 12 13 3]	22	4	18	21	14
1	[6 5 4 22 7 2 9 8 15 17 1 0 10 11 14 19 21 18 16 20 13 12 3]	22	3	18	20	17
7	[6 4 5 7 2 1 15 9 22 8 0 21 18 10 11 14 17 19 16 12 20 13 3]	22	8	18	21	12
14	[6 4 5 8 22 2 7 9 15 1 14 10 11 17 18 0 21 19 16 13 20 12 3]	22	4	18	19	14
2	[6 5 4 7 9 2 22 8 15 1 10 17 11 14 0 21 18 19 16 20 12 13 3]	22	6	18	21	16
15	[6 22 5 8 4 2 7 9 1 14 15 10 11 0 18 17 21 19 16 20 13 12 3]	22	1	18	20	14
8	[5 6 2 4 7 22 8 9 1 10 15 0 11 17 18 21 14 19 16 20 13 12 3]	22	5	18	20	14
3	[6 7 5 4 2 15 22 8 9 1 17 0 10 18 19 14 11 21 20 16 12 13 3]	22	6	19	21	13
5	[6 5 4 7 2 9 22 8 15 1 0 10 14 11 18 17 21 19 20 16 13 12 3]	22	6	19	20	14
11	[6 5 4 2 7 8 22 9 1 11 15 17 10 14 18 0 21 19 16 20 13 12 3]	22	6	18	20	14
12	[6 22 5 8 2 7 9 4 15 1 0 11 10 18 21 14 17 16 19 20 13 12 3]	22	1	17	20	13
4	[4 6 5 2 7 8 22 9 1 15 10 11 17 0 14 18 19 21 16 13 12 20 3]	22	6	18	19	15
13	[6 22 2 4 9 5 8 7 14 11 1 10 15 0 17 18 21 19 16 13 20 12 3]	22	1	18	19	15

Table A.7: CNN ordered SNP relevance values in Breast Cancer dataset. The index in the increasingly ordered relevance vector of the interacting SNPs (orders two, three and four) is presented, with higher indexes representing higher relevance values. The notation used to represent the SNPs is presented in Table A.5.

				Index		
Architecture Nº	Increasingly Ordered Relevance Vector	SNP 3	SNP 22	SNP 16	SNP 13	SNP 18
47	[15 5 9 0 6 7 4 17 10 11 1 19 2 20 21 14 16 8 13 12 18 22 3]	22	21	16	18	20
44	[5 6 7 0 4 9 15 11 10 17 1 20 2 19 12 21 16 14 18 13 8 22 3]	22	21	16	19	18
46	[10 18 17 5 15 21 19 16 1 6 4 7 9 0 12 11 2 20 14 3 13 8 22]	19	22	7	20	1
43	[15 5 10 17 18 21 7 1 6 19 16 4 9 0 11 12 2 20 13 14 8 3 22]	21	22	10	18	4
45	[5 15 6 17 10 4 7 1 16 18 9 19 21 0 11 12 2 20 13 14 8 3 22]	21	22	8	18	9
42	[5 15 6 10 21 4 7 1 17 18 9 16 19 0 12 11 2 20 14 13 8 3 22]	21	22	11	19	9
41	[5 6 7 15 9 4 0 1 10 17 11 16 2 20 19 21 14 18 13 22 12 8 3]	22	19	11	18	17
40	[6 5 4 15 7 9 11 1 2 0 17 10 14 21 20 19 12 8 18 16 13 22 3]	22	21	19	20	18
38	[5 7 15 9 6 17 4 18 1 10 16 19 21 0 11 20 2 12 14 8 13 22 3]	22	21	10	20	7
39	[5 6 15 7 4 9 1 17 10 18 19 0 21 11 16 12 2 20 13 8 14 3 22]	21	22	14	18	9
37	[5 15 6 9 4 0 7 1 10 17 11 2 19 21 20 16 14 18 22 13 12 8 3]	22	18	15	19	17
36	[5 6 15 0 10 7 1 17 4 16 9 19 18 2 21 11 12 20 13 22 8 14 3]	22	19	9	18	12
35	[15 5 18 6 17 10 21 16 7 4 9 1 19 0 11 2 12 20 14 8 3 13 22]	20	22	7	21	2
34	[5 6 15 9 4 7 0 11 1 17 2 10 19 20 21 8 14 16 12 18 22 13 3]	22	20	17	21	19
33	[5 6 9 7 15 4 17 18 10 0 1 11 2 16 21 19 20 12 8 14 13 22 3]	22	21	13	20	7
32	[5 15 6 9 17 7 4 18 1 16 19 10 0 21 11 2 12 20 14 8 13 22 3]	22	21	9	20	7
31	[5 9 7 15 6 4 17 0 1 10 11 18 2 16 19 21 20 12 8 14 13 22 3]	22	21	13	20	11
30	[5 15 6 9 0 4 17 7 1 10 11 19 2 16 21 20 12 18 13 14 8 22 3]	22	21	13	18	17
29	[0 5 6 15 17 10 1 7 4 9 16 19 11 21 18 12 2 20 13 8 14 22 3]	22	21	10	18	14
28	[6 5 4 15 7 9 0 11 1 2 17 14 20 19 21 10 12 8 13 16 18 22 3]	22	21	19	18	20
27	[5 6 15 0 9 10 1 7 4 16 19 11 17 21 18 2 13 20 22 8 12 14 3]	22	18	9	16	14
26	[6 4 5 9 0 2 7 11 15 1 17 14 21 10 20 12 19 8 13 18 16 22 3]	22	21	20	18	19
25	[5 15 6 0 4 7 1 10 17 9 2 16 11 19 18 21 20 22 12 8 14 13 3]	22	17	11	21	14
24	[5 6 0 15 1 7 17 4 10 9 16 11 2 19 21 18 12 20 13 8 14 3 22]	21	22	10	18	15
23	[6 5 4 15 7 9 11 0 1 2 17 10 19 14 20 21 12 8 18 16 13 22 3]	22	21	19	20	18
22	[5 7 6 15 9 4 1 10 18 17 21 0 16 19 11 2 8 20 12 14 22 13 3]	22	20	12	21	8
21	[5 0 6 15 4 10 1 7 17 9 16 11 2 19 21 18 12 20 8 14 13 22 3]	22	21	10	20	15
20	[5 6 4 7 15 9 2 17 11 0 1 14 8 10 19 21 20 18 12 16 22 13 3]	22	20	19	21	17
19	[5 9 7 6 4 15 17 11 0 1 10 18 2 19 16 21 8 12 20 14 22 13 3]	22	20	14	21	11
18	[5 6 4 9 7 15 2 17 8 11 1 0 10 14 21 18 19 22 20 16 12 13 3]	22	17	19	21	15
17	[5 6 4 15 7 9 0 17 11 2 1 10 14 22 19 20 21 12 18 16 8 13 3]	22	13	19	21	18
16	[6 5 22 4 7 9 2 15 8 0 11 1 10 17 21 18 14 19 16 20 12 13 3]	22	2	18	21	15
15	[5 6 7 22 4 9 8 2 15 17 0 11 1 10 14 21 18 19 16 20 12 13 3]	22	3	18	21	16
14	[5 9 22 8 7 6 4 17 15 2 11 0 10 21 18 14 1 19 16 12 20 13 3]	22	2	18	21	14
13	[6 7 22 4 5 9 2 15 8 10 0 11 18 17 14 1 21 19 16 20 13 12 3]	22	2	18	20	12
12	[5 7 6 9 8 4 22 2 11 15 17 19 0 16 1 14 18 10 21 12 20 13 3]	22	6	13	21	16
11	[5 22 6 4 7 9 8 15 2 0 1 17 11 10 14 18 21 19 20 16 12 13 3]	22	1	19	21	15
10	[5 6 22 7 4 2 9 8 15 1 17 11 0 10 14 18 21 19 16 20 13 12 3]	22	2	18	20	15
9	[5 22 4 7 8 6 9 15 2 11 1 18 17 14 10 0 19 21 13 16 20 12 3]	22	1	19	18	11
8	[5 6 22 7 2 4 9 8 15 17 10 1 0 14 18 19 11 16 21 20 12 13 3]	22	2	17	21	14
7	[5 6 22 7 9 4 8 2 15 1 0 11 17 18 14 10 21 16 19 20 12 13 3]	22	2	17	21	13
6	[5 6 7 22 8 2 9 4 15 1 0 10 11 17 18 19 21 14 16 20 12 13 3]	22	3	18	21	14
5	[5 6 7 9 22 4 2 8 15 17 1 10 19 14 18 11 0 21 16 20 12 13 3]	22	4	18	21	14
4	[5 6 9 22 7 4 2 15 8 17 0 10 11 14 18 1 19 21 16 20 12 13 3]	22	3	18	21	14
3	[5 6 22 7 9 4 8 2 15 17 1 0 10 18 11 14 21 19 16 20 12 13 3]	22	2	18	21	13
2	[6 7 5 4 9 2 8 15 1 17 10 0 18 11 19 21 22 14 16 20 12 13 3]	22	16	18	21	12
1	[6 5 22 7 9 4 8 2 15 0 17 18 11 14 1 10 21 19 16 12 20 13 3]	22	2	18	21	11
0	[5 22 6 7 4 9 8 2 15 0 1 17 11 10 18 14 21 19 16 20 12 13 3]	22	1	18	21	14