UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

JOSÉ PEDRO SILVEIRA MARTINEZ

An Embedded Categorical Feature Selector applied to the Genotype-Phenotype Prediction Problem

Work presented in partial fulfillment
of the requirements for the degree of
Bachelor in Computer Science

Advisor: Prof. Dr. Márcio Dorn
Coadvisor: M.Sc Bruno Iochins Grisci

Porto Alegre
December 2019

ACKNOWLEDGMENTS

I send thanks to my advisors, Professor Márcio Dorn and Bruno Grisci, for introducing and discussing the theoretical aspects that supported this study with me, for making me believe I am capable of working as a scientist, and for highlighting stepping stones on the road of my own professional and personal improvement. To Dr. Eduardo Ávila for the attentive review of the work and precise corrections concerning the biological aspects of the study. To my sister, Tatiana, for providing me emotional support and strengthening my spiritual linkage with our common ancestors. To my colleagues at Meerkat for supporting me financially and trusting I am doing my best in spite of the new challenges I am facing. Last but far from least, thank you to the INCT Forense for providing the data analyzed in this study.

“The Road to Wisdom? Well, it’s plain
And simple to express:
Err
and err
and err again,
but less
and less
and less.”
— Piet Hein, The Road to Wisdom

ABSTRACT

One crucial preprocessing step in many machine learning algorithms is feature selection since many predictions tasks are defined without knowledge of which attributes lying on data are relevant to the given task. The burden of evaluating all possible features subsets is computationally intractable and many heuristics for choosing sub-optimal attributes sets have been proposed and evaluated over the last decades. With the advance of data storage and collection technology, databases posing novel predictions tasks are filled with spurious attributes, which enforces the urge for general and computationally inexpensive feature selection algorithms. Elimination of redundant attributes is capable of improving machine learning models predictive capability, while the discovery of relevant features has an important scientific value on domains in which knowledge about the relationship of collected data attributes is null or insufficient. This study proposes N3O-D, a novel feature selection algorithm based on neuroevolution and mutual information which automatically selects categorical attributes as it learns to solve a given learning task. The proposed method classification and selection capability are experimentally evaluated on the genotype-phenotype prediction of eye and skin color. Experimental results showed that the method has the potential of improving classification performance obtained from state-of-art feature selection frameworks, achieving it on some of the evaluated data sets.

Keywords: Neuroevolution, feature selection, metaheuristics, genetic algorithm, neural networks, machine learning, mutual information, information theory.

Um Seletor de Atributos Categóricos baseado em Neuroevolução aplicado ao problema de Predição de Fenótipo a partir de Genótipo

RESUMO

Uma etapa crucial do pré-processamento de muitos algoritmos de aprendizado de máquina é a seleção de atributos, visto que muitas tarefas preditivas são definidas sem um conhecimento prévio de quais atributos presentes nos dados são de fatos relevantes para o problema. A tarefa de avaliar todos os possíveis subconjuntos de atributos é computacionalmente intratável e heurísticas que propõe conjuntos de atributos sub-ótimos tem sido propostas e avaliadas nas últimas décadas. Com o avanço tecnólogico da coleta e armazenamento de dados em larga escala, bancos de dados apresentam tarefas preditivas inéditas contendo atributos supérfluos, reforçando a carência de algoritmos de seleção de atributos genéricos e computacionalmente baratos. A eliminação de atributos redundantes é capaz de melhorar a capacidade preditiva de algoritmos de aprendizado de máquina, enquanto a descoberta de atributos relevantes tem um valor científico importante para domínios onde o conhecimento sobre a relação dos dados coletados é nula ou insuficiente. Esta monografia propõe N3O-D, um seletor de atributos baseado em neuro-evolução e informação mútua que automaticamente seleciona atributos categóricos enquanto aprende a resolver uma tarefa de aprendizado. O método proposto é avaliado experimentalmente na predição dos fenótipos cor de olho e cor de pele a partir de dados genotípicos. Resultados experimentais demonstraram que o método é capaz de superar a capacidade preditiva obtida a partir de métodos de seleção de atributos no estado da arte, atingindo o objetivo em alguns dos conjuntos de dados analisados.

Palavras-chave: neuroevolução, seleção de atributos, metaheurísticas, algoritmo genético, redes neurais, aprendizado de máquina, informação mútua, teoria da informação.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

CNN	Convolutional neural network
CV	Cross-validation
DNA	Deoxyribonucleic acid
FS	Feature selection
FS-NEAT	Feature selective neuroevolution of augmenting topologies
FN	False negative
FP	False positive
GA	Genetic algorithm
INCT	Institutos Nacionais de Ciência e Tecnologia
KW	Kruskal-Wallis one-way analysis of variance
NN	Neural network
MI	Mutual information
mRMR	Minimum redundancy - maximum relevance
N3O	Three new operators
N3O-D	Three new operators, adapted to discrete data
NEAT	Neuroevolution of augmenting topologies
RBF	Radial basis function
RFE	Recursive feature elimination
RMSProp	Root mean square propagation
RNN	Recurrent neural network
SNP	Single nucleotide polymorphism
STR	Short tandem repeat
SVM	Support vector machine
SVMRFE	Support vector machine methods based on recursive feature elimination

TN	True negative
TP	True positive
UFS	Univariate feature selection
UFS(MI)	Univariate feature selection using mutual information as ranking criterion
XOR	Exclusive OR (logical operator)

LIST OF SYMBOLS

$\uplus$	Disjoint set union
e	Euler number
$tanh$	Hyperbolic tangent
$\|.\|_1$	L_1-norm
log	Logarithm
$p(x, y)$	Joint probability distribution of random variables x and y
ln	Natural logarithm
ϕ	Neural network activation function
ψ	Neural network aggregation function
Θ	Neural network weights
θ	Neural network weight
$\mathcal{N}$	Normal distribution
$p(x)$	Marginal probability distribution of random variable x
$\max$	Maximum
λ	Regularization Coefficient
$\cap$	Set intersection
$\|.\|$	Set length
$\in$	Set pertinency
$\cup$	Set union
$\sum$	Summation
$var(.)$	Variance (statistical measure)

CONTENTS

1 INTRODUCTION

Most predictive models assume that all of the specified attributes are useful to the predictive task, although synthesized and real-world problems might present inputs that are either not correlated to the learning objective (*irrelevant*) or very similar to other inputs (*redundant*). Including irrelevant features increases model induction and prediction computational cost and might harm the model's inference capability (LAZAR et al., 2012; AGGARWAL et al., 2014; MIAO; NIU, 2016; CILIA et al., 2019). On most posed tasks, the relevant attributes are unknown and feature selection has the potential of discovering knowledge of gathered data. Genomic selection results, for instance, indicate which parts of the genome influence phenotype characterization (YOUNG; BENONISDOTTIR; PRZEWORSKI, 2019) and disease treatment response (GRISCI; FELTES; DORN, 2018; LU; CHEN; YAN, 2017). On real-world tasks, collecting, maintaining and providing input data presents an economic cost to be minimized, as well as more strict performance requirements.

An example of a prediction problem in the biology field is genotype-phenotype mapping (YOUNG; BENONISDOTTIR; PRZEWORSKI, 2019), in which specific characteristics of a given individual are to be predicted from its genetic encoding. Since genomes yield a lot of data, which is mostly uninformative given a particular phenotype, feature selection (also referred to as genomic selection, given the nature of input data) is a natural application to this kind of problem. Data collection techniques, such as those employed to obtain single nucleotide polymorphisms (SNPs) and short tandem repeats (STRs), already focus on more informative parts of the genome; nevertheless, more aggressive filtering on previously collected data is required by most tasks in the biological world. A more detailed discussion of the genotype-phenotype mapping problem is presented in Chapter 3.

While much effort has been put on the aim of automatically finding optimal feature sets of machine learning data sets, current results are still unsatisfactory, especially for domains where knowledge about data is shallow (ANG et al., 2016; FELTES et al., 2019). Additional challenges include inherent noise in data distribution and strict assumptions made by feature selection algorithms (LAZAR et al., 2012; AGGARWAL et al., 2014). In biological areas, the discovery of relevant attributes may provide new insights on already established biological knowledge and can inspire more efficient data collection and storage techniques. Feature selective algorithms based on neuroevolution recently presented

promising results on reinforcement learning and classification tasks (WHITESON et al., 2005; TAN et al., 2009; SOHANGIR; RAHIMI; GUPTA, 2013; SOHANGIR; RAHIMI; GUPTA, 2014; GRISCI; FELTES; DORN, 2018). In particular, Grisci, Feltes and Dorn (2018) showed that the statistical relationship between attributes is an informative guide to evolution's stochasticity. This work was evaluated on microarray (gene expression) data sets labeled with different cancer conditions. The method could successfully select around 200 informative genes (roughly half of them already discussed in previous research) over multiple runs on distinct databases with a large number of features (typically tens of thousands of features and hundreds of samples). Each experiment constructed neural networks populations in which the best performing individual utilized 10 genes on average.

The present study introduces a feature selection algorithm based on neuroevolution that uses mutual information (COVER; THOMAS, 2006; BUNNEY et al., 2017; VERGARA; ESTÉVEZ, 2015) to guide inclusions of categorical attributes into neural networks that are evolved to optimize prediction accuracy concerning a classification task. The proposed method is evaluated on the genotype-phenotype prediction task of inferring eye and skin color of an individual given its SNPs gene markers.

This work's objective is to assess the proposed method's genomic selection capability, as well as its ability to solve the genotype-phenotype prediction problem on its own. To assess the proposed method's inference capability, its classification performance is compared to the baseline accuracy. The method's selection quality is investigated by being compared to a univariate feature selection algorithm, as well as the selection retrieved by its algorithmic basis which is also powered by neuroevolution.

The present study is organized in the following manner: Chapter 2 reviews the theoretical foundations of the study; Chapter 3 explains the genotype-phenotype prediction problem. The proposed method is presented on Chapter 4. Chapter 5 describes how gene selection quality is assessed, presents the dataset used on experiments and discusses results. Chapter 6 relates the research objectives with the obtained results, outlining the work's contributions.

2 THEORETICAL BASIS

The following chapter presents computational methods and theoretical definitions that are essential to the present work.

2.1 Genetic algorithms

Genetic algorithms (GAs) are optimization methods inspired by the biological process of evolution. They are essential to this work because the employed neuroevolution methods are supported by a GA.

A broader review of the core operators and parameters of GAs and its extensions is given by Beyer and Schwefel (2002). The more crucial to this work are detailed next. A genetic algorithm evolves a *population* of *individuals* throughout *generations*. Individuals represent solutions to the optimization problem and are encoded such as they can be combined to yield *offspring* individuals, through an operation referred to as *crossover*. Individuals might evolve in a stochastic manner through an operation called *mutation*. Mutations are applied randomly each generation; their change on individuals' encoding is also random, and is scaled by the hyper-parameter *mutation power*. The capability of a given individual to solve the task in hand is assessed by a pre-defined *fitness function*. The population evolves as individuals are mutated and offspring replace less fit parents.

Optimization occurs by applying selective pressure on most fit individuals. A common and efficient example is *elitism*, which copies the most fit individuals of a generation to the next one unchanged. Although the fitness of an individual is an informative guide to optimization, it often leads to sub-optimal solutions. The employment of randomness in diverse operations is the strategy adopted by GAs to avoid local minimums.

Next, it is presented a pseudo-code that describes the implementation of a simple GA. On the simply described evolution, $|P|$ and G are the chosen population size and number of generations, respectively, P_i represents the population at the i-th generation, g denotes the current evolution generation, M_i represents the mating pairs computed for generation i, O_i is the set of individuals generated by the application of crossover on computed mating pairs. For readability, essential operations are abstracted as functions. $marriage$ summarizes the computation of mating pairs (which might consider speciation), $mating$ and $mutation$ respectively abstract the application of the crossover and mutation operators on computed mating pairs and offspring individuals. $selection$

abstracts how individuals are passed onto the next generation, a process that usually involves fitness assessment and elitism; the disjoint set union is used for denoting which individuals are selected as the new population might be composed of offspring only. A more general procedure is presented in Beyer and Schwefel (2002).

Algorithm 1: General structure of a genetic algorithm.

Data: $|P|, G$

Result: P_G

$P_0 \leftarrow \{new\_individual()\}$ ▷ $|P_0| = |P|$;

$g \leftarrow 0$;

while $g < G$ **do**

  $M_g \leftarrow marriage(P_g)$;

  $O_g \leftarrow mating(M_g)$;

  $O'_g \leftarrow mutation(O_g)$;

  $P_{g+1} \leftarrow selection(P_g \uplus O'_g)$ ▷ $|P_{g+1}| = |P|$;

  $g \leftarrow g + 1$;

end

2.2 Neural networks

Neural networks (NNs) are universal function approximators that map vectors of $\mathbb{R}^i$ to $\mathbb{R}^o$, where i and o are the numbers of network input and output values, respectively (HORNIK, 1991; QU; WANG, 2019). NNs utilize neurons as an atomic processing unit. Neurons are connected through directed connections that individually weight the contribution of input signals to compute a single activation output. Weighted input values are aggregated and passed to an activation function. Equation 2.1 defines such process in a general way, where ϕ and ψ are respectively the aggregation and activation functions, I is the set of input nodes, and $W : I \to \mathbb{R}$ and $V : I \to \mathbb{R}$ are functions that map input nodes to real-values numbers, respectively representing connection weights and input activation values. Equation 2.2 describes the same operation for the case where average and identity are respectively chosen as aggregation and activation functions, $w_i = W(x_i)$ and $a_i = V(x_i)$.

$$\phi(\psi(I, W)) \tag{2.1}$$

$$\frac{\sum_{x_i \in I} x_i w_i}{|I|} \tag{2.2}$$

Networks applied to learning tasks mainly differ in their architecture, which specifies how neurons are connected. A short explanation of common employed NN architectures is presented next.

2.2.1 Architectures

Feed-forward neural networks: Also referred to as multi-layered perceptron, feed-forward neural networks have their neurons organized in ordered layers. Neurons from a given layer l are connected to (typically all) neurons from layers $l+1$ and $l-1$, except for input and output layers which are respectively connected to their successor and predecessor layer (RUCK et al., 1990).

Convolutional neural networks (CNNs): CNNs differ from conventional neural networks as they receive input values containing structure that is exploited by the network's aggregation function. Typical choices for aggregation on CNNs are convolutions that utilize spatial relations on input. Convolutional neural networks have been successfully applied in Computer Vision on tasks such as object detection, face recognition, and depth estimation (DRUZHKOV; KUSTIKOVA, 2016).

Recurrent neural networks (RNNs): Recurrent neural networks aim at processing sequential data. This is achieved by creating recurrent connections within the network's topology. The activation values for a given input sequence element x_i are stored in memory gates, which in turn are passed as input to the network along with the next input, x_{i+1}. Therefore the network output for a sequence element depends on the sub-sequence that precedes it, although one element is processed at a time. RNNs have been applied with success to acoustic modeling and natural language processing (MULDER; BETHARD; MOENS, 2015).

2.2.2 Optimization

Besides a network architecture and neuron aggregation and activation functions, there are other hyper-parameters, such as the number of neurons and layers. Neural network topologies are commonly designed by humans, while connection weights are opti-

mized by error propagation algorithms, such as backpropagation and RMSProp (RUMELHART; HINTON; WILLIAM, 1986; TIELEMAN; HINTON, 2012). On neuroevolution (introduced next in Section 2.3), evolutionary strategies are employed to optimize such hyper-parameters, and some attempt to evolve topology connections and weights simultaneously (STANLEY; MIIKKULAINEN, 2002; WHITESON et al., 2005). Neural networks are commonly applied to classification tasks (ZHANG, 2000), and their application is also extended to reinforcement learning tasks.

2.3 Neuroevolution

Neuroevolution is an area of evolutionary computation that employs evolutionary algorithms to discover and optimize neural networks (NNs) concerning a predictive task. Methods found in the literature differ on how neural networks are genetically encoded and which operators are used to improve already found solutions (STANLEY; CLUNE; LEHMAN, 2019; FLOREANO; DüRR; MATTIUSSI, 2008; DUFOURQ; BASSET, 2017; XIE; YUILLE, 2017). Genetic algorithms are defined and implemented in a very abstract manner (LUKE, 2013; BEYER; SCHWEFEL, 2002); neuroevolution naturally inherits this flexibility.

Neuroevolution is capable of relaxing restrictions common to usual neural network optimization methods. In principle, neuroevolution algorithms do not require NNs activation and loss functions to be differentiable, in contrast to backpropagation based optimization methods (RUMELHART; HINTON; WILLIAM, 1986) (TIELEMAN; HINTON, 2012). It is also possible to evolve network topologies (better detailed when discussing NEAT in Section 2.3.1) which is a harder problem as it increases the size of the search space. A main drawback of neuroevolution when compared to usual neural network optimization methods is the necessity of iterating over all training data for a single individual's fitness update, although recent work suggests this burden may be diminished (MORSE; STANLEY, 2016).

2.3.1 Neuroevolution of augmenting topologies

As discussed above, neuroevolution is a family of methods that applies concepts of evolutionary computation to optimize neural networks. Neuroevolution of augmenting

topologies (NEAT) (STANLEY; MIIKKULAINEN, 2002) is a neuroevolution method that is capable of evolving network weights and topologies. NEAT defines mutation and crossover operators on top of a genetic algorithm framework.

An advantage of automatically evolving topologies is that humans do not need to know in advance which NN architecture is best suited for a given task. For instance, empirical evidence showed that NEAT can discover recurrent neural network architectures from a simple feed-forward topology at tasks where recurrence was known to be a desirable property (WHITESON et al., 2005; TAN et al., 2009).

NEAT has been validated on abstract problems such as the computation of the non-linearly separable logical function XOR and the reinforcement learning task of double pole balancing (STANLEY; MIIKKULAINEN, 2002). Its feature selective variations have been successfully applied to supervised (GRISCI; FELTES; DORN, 2018; SOHANGIR; RAHIMI; GUPTA, 2013; SOHANGIR; RAHIMI; GUPTA, 2014) and reinforcement learning tasks (WHITESON et al., 2005). The rest of this section details NEAT most important extensions to a regular genetic algorithm, as well as specifies its mutations and crossover operators.

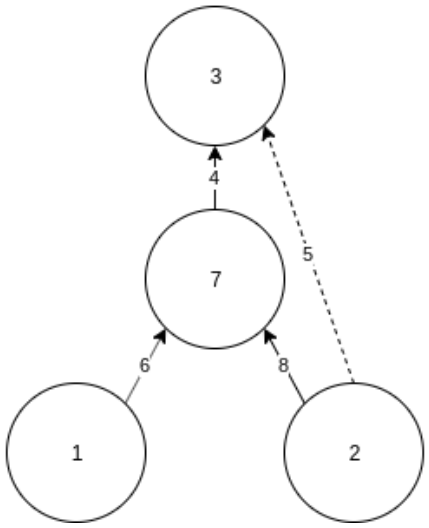
Genetic encoding: A NEAT individual is directly represented by a list of nodes and a list of edges that represent their topology, which denotes the individual's phenotype, in the context of evolutionary algorithms. In computational terms, genomes are directed acyclic graph representations. Given an individual I, its genome g may be partitioned into $g^{nodes} \cup g^{edges}$. Such partition is important because node genes possess different attributes than in comparison to connection genes. While node genes are encoded only by its type (input, hidden or output) and by its bias weight, an edge gene is encoded by the two nodes it connects, its connection weight, and whether it is enabled. A disabled connection gene is ignored on the individual's phenotypical representation.

Topological innovation via speciation: Although evolution strategies yield a vast toolbox for extending the simplest genetic algorithms (see Beyer and Schwefel (2002) for a wider review) NEAT relies mainly on speciation. As a given topology evolves, it needs mutation steps to fine-tune new weights. During this process, it is unfair that a unique, new promising individual must compete with more straightforward but already tuned individuals. Such a problem is approached with speciation and fitness sharing, where individuals from different species do not compete for offspring, and individuals from the same species influence each other's fitness. To group individuals in species, the competing conventions problem and an individuals' difference function must be addressed.

Figure 2.1: Depiction of a NEAT individual's genome and phenotype.

Node Gene 1	Node Gene 2	Node Gene 3	Node Gene 4
type: input	type: input	type: hidden	type: output
marker: 1	marker: 2	weight: 2	weight: 5
		marker: 7	marker: 3

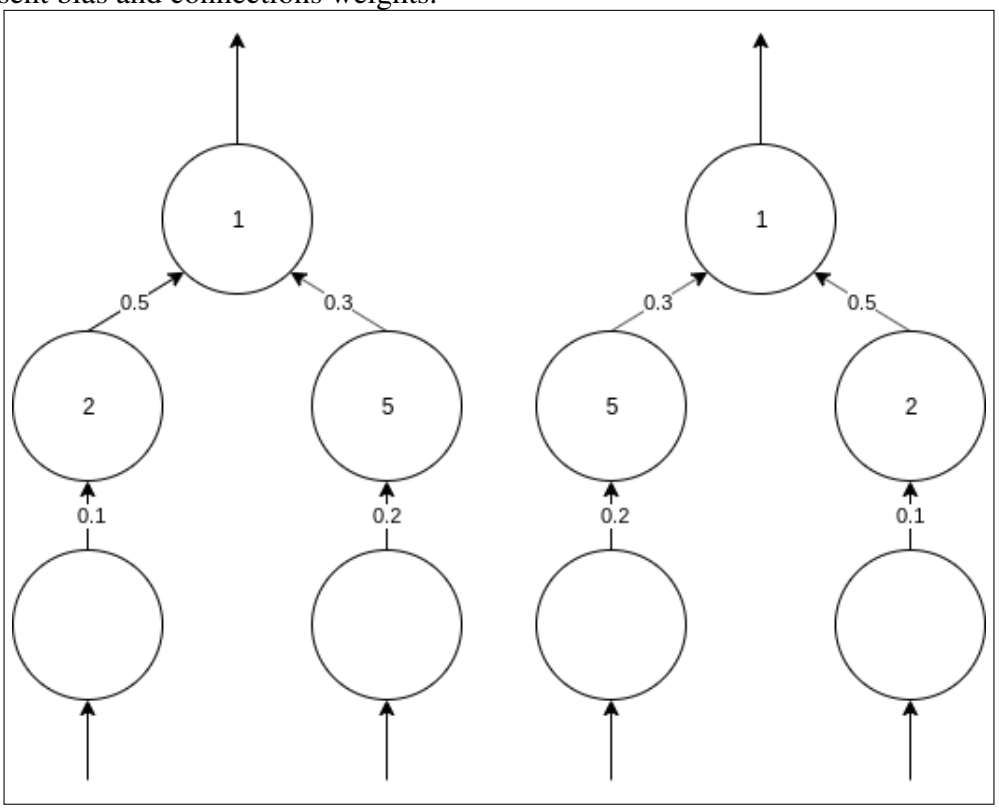
Connection Gene 1	Connection Gene 2	Connection Gene 3	Connection Gene 4
incoming node: 1	incoming node: 2	incoming node: 3	incoming node: 2
outgoing node: 3	outgoing node: 3	outgoing node: 4	outgoing node: 4
weight: 0.1	weight: 0.5	weight: 0.2	weight: 0.3
disabled: no	disabled: yes	disabled: no	disabled: no
marker: 6	marker: 8	marker: 4	marker: 5



Historical innovation markers: An issue common to neuroevolution algorithms that evolve topologies is the competing conventions problem. Individuals with apparently distinct genomes might produce the same output. The simplest case, when an individual's genome is a permutation of the other, is depicted in Figure 2.2. Although the depicted networks compute the same function, a naive representation might be unable to reveal their similarities in reasonable computation time. Such distinction is important because individuals must be distinguished by their behavior, not by their genome.

NEAT handles the competing conventions problem by attributing a global innovation number to newly added genome elements (topology nodes or connections). These innovation markers induce a natural order for individuals' genes, allowing the construc-

Figure 2.2: **Depiction of competing conventions.** Circles with loose incoming edges represent input nodes. Circles with loose outgoing edges represent output nodes. Numbers represent bias and connections weights.



tion of a difference function able to distinguish between matching, exceeding, and disjoint genes. This way, mutated individuals will surely present an encoding similar to their prior, and offspring receives an encoding similar to their parents. Innovation markers induce the concepts of matching, exceeding, and disjoint genes, as well as supports the definition of a difference function between two arbitrary individuals, which is detailed next.

Individuals difference: Given genes yield historical markers, two different genomes are easily comparable. Given a genome g_1, its genes might be partitioned into $g_1 = g_1^M(g_2) \cup g_1^E(g_2) \cup g_1^D(g_2)$ given another individual's genome. Matching genes $g_1^M(g_2)$ represents genes in g_1 which historical marker is found in g_2. Exceeding genes $g_1^E(g_2)$ represent genes in g_1 that are more recent than all of those in g_2, while disjoint genes $g_1^D(g_2)$ are genes in g_2 whose historical marker is not present in g_1, but are newer than at least one gene in g_1. This genome partition is explicitly used by the crossover operator and the difference function. The genome difference of two individuals I_1, I_2 respectively with genomes g_1, g_2 is defined by the following equation, where E, D represents the set of

exceeding and disjoint genes of g_1 with respect to g_2, respectively.

$$d(g_1, g_2) = \frac{c_1|E| + c_2|D|}{max\{|g_1|, |g_2|\}} + c_3\hat{W} \tag{2.3}$$

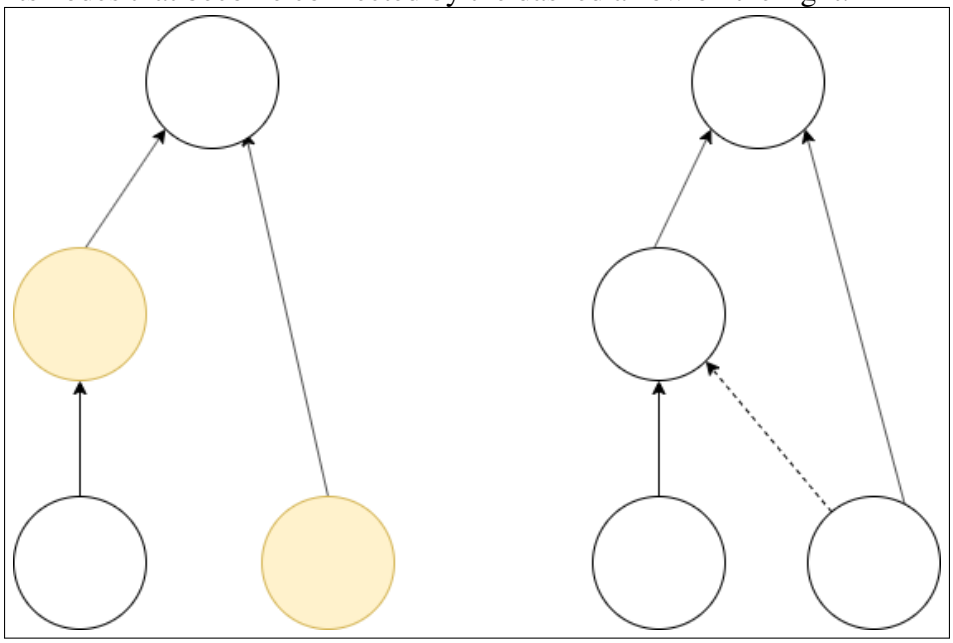
$$D(I_1, I_2) = d(g_1^{nodes}, g_2^{nodes}) + d(g_1^{edges}, g_2^{edges}) \tag{2.4}$$

The coefficients c_1, c_2, c_3 balance the importance of evolved topologies and weights. In a scenario where $c_1 = c_2$, the difference function is symmetric. Larger values of c_3 enable a more refined distinction between individuals that learned to approximate different functions with similar topologies.

Mutation Change Weight: There is a chance that a given node's bias or connection's weight is perturbed during optimization. Such alteration follows the normal distribution and is scaled by the mutation power employed on evolution. This operator is the responsible for fine-tuning topological innovations, given that other mutations only update weight values in case new elements are added to the topology.

Mutation Add Connection: A new connection might be created between two unconnected nodes. When the aim is to evolve a feed-forward network (which is the interest of this study), the addition of connections that create cycles within the topology is prohibitive.

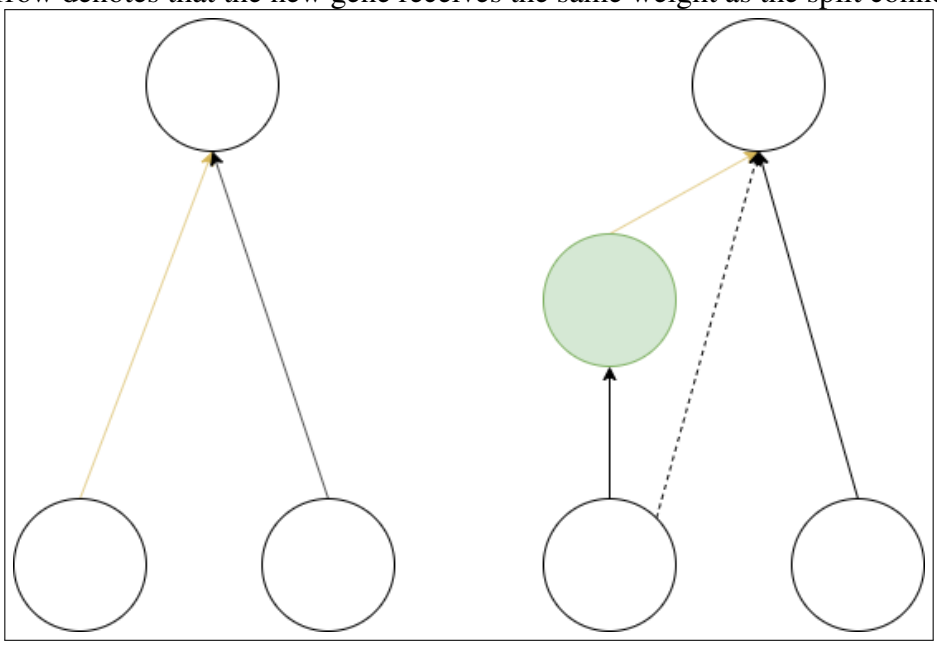
Figure 2.3: **Depiction of mutation Add Connection.** Two unconnected nodes become connected by the addition of an edge in the topology. On the left, yellow-colored ellipsis represents nodes that become connected by the dashed arrow on the right.



Mutation Add Node: An existing connection within the topology might be disabled in

order to make room for a novel hidden node, connected to the nodes that composed the target edge. The connection incoming to the new node receives a weight of 1, while the outgoing edge receives the same weight as the disabled one.

Figure 2.4: **Depiction of mutation Add Node.** An existing connection is split by the inclusion of a hidden node in the topology. On the left, the yellow-colored arrow represents the split edge. On the right, the green-colored node represents the node added by the mutation, the dashed arrow indicates that the split connection is disabled, and the yellow-cored arrow denotes that the new gene receives the same weight as the split connection.



Crossover: Given two individuals I_1, I_2, where $fitness(I_1) > fitness(I_2)$ the crossover operator specifies how their offspring are created. Matching genes are inherited randomly, while disjoint and exceeding genes concerning I_1 are inherited. The new individual inherits genes from both its parents, in the case $fitness(I_1) = fitness(I_2)$. NEAT also specifies that an inherited disabled connection gene might become active in case it is enabled in one of the parents, which turns the crossover operator into a stochastic process. Figure 2.5 depicts a crossover scenario and a possible outcome.

Fitness Sharing: Competition is avoided by allocating a fraction of the offspring population to each of the computed species, proportional to the fitness contribution held by the species. Individuals fitness are transformed by Equation 2.5 to compensate for the lack of diversity caused by large species, where f_i and f'_i respectively represent the original and adjusted fitness of individual i and $S(i)$ is the set of individuals which were attributed the same species as i. The number of offspring individuals generated by a particular species

is proportional to the sum of the adjusted fitness of its individuals.

$$f'_i = \frac{f_i}{|S(i)|} \tag{2.5}$$

2.3.2 Feature selective neuroevolution of augmenting topologies

NEAT assumes all of the available input is relevant to the given task once every input node is present on new individuals, and its operators do not support the exclusion of nodes or edges. Although evolution can nullify the weights of input nodes, such a phenomenon is not explicitly guided by the employed optimization and therefore consumes unnecessary compute resources. Feature selective neuroevolution of augmenting topologies (FS-NEAT) modifies how novel individuals are created by including only one of the possible input nodes. To evolve networks capable of combining information from multiple inputs, a new mutation, *Add Input*, is added to the genetic algorithm that powers evolution.

With the modifications above, FS-NEAT automatically performs feature selection as it searches for networks with excellent prediction performance concerning the given task. Intuitively, included irrelevant or spurious features will not yield any performance gain. Consequently, individuals mutated this way are not likely to survive in the next generations.

Mutation Add Input There is a stochastic event in which an arbitrary input node, not present in an individual genome, becomes connected to a hidden or output node present in the given individual's topology. The new connection properties are the same as in the *Add Connection Mutation*. Such a mutation is necessary due to FS-NEAT minimalist start. Topologies containing input nodes that do not contribute to the given task are not likely to survive, therefore only individuals with relevant inputs persist throughout evolution.

2.3.3 Three new operators

Grisci, Feltes and Dorn (2018) proposed three core modifications to FS-NEAT in an algorithm named after the introduced alterations, N3O. Proposed modifications were motivated by the large number of features in data that were approached by the method. The same work demonstrated that the proposed alterations improve the exploration of

the search space. The method relies on statistics extracted from numerical training data in order to better guide the evolutionary process. The requirement of continuous data posed by employed statistics motivated the proposal of a novel feature selection method based on neuroevolution, able to provide similar guidance on categorical data, which is presented in Chapter 4. Since N3O is a core basis for the method proposed in this study, its extensions on FS-NEAT are detailed in the next sections.

Input Relevance Assessment: One main improvement of N3O over FS-NEAT is that it incorporates information about input features into evolution. In Grisci, Feltes and Dorn (2018), this is done by the employment of the Kruskal-Wallis one-way analysis of variance (KW) (KRUSKAL; WALLIS, 1952). KW tests whether ranked samples from multiple groups G belong to the same statistical distribution. The test is defined by Equation 2.6, where G denotes the sample groups to be tested, N represents the total number of samples, r_x denotes the rank attributed to a given sample x, while $\bar{r}_g$ and $\bar{r}$ are defined by Equations 2.7 and 2.8, respectively. The test outputs p-values representing the probability of groups in G belonging to equal distributions. Being non-parametric is one important characteristic of KW, as it does not assume that input samples belong to a given distribution. A drawback of the test is that it is designed for real-values numbers, being inapplicable to categorical data.

$$H(G) = (N-1)\frac{\sum_{g \in G} |g|(\bar{r}_g - \bar{r})^2}{\sum_{g \in G} \sum_{x \in g} (r_x - \bar{r})^2} \tag{2.6}$$

$$\bar{r}_g = \frac{\sum_{x \in g} x}{|g|} \tag{2.7}$$

$$\bar{r} = \frac{N+1}{2} \tag{2.8}$$

Statistical Filtering: Input features that present a non-negligible probability of belonging to the same distribution as the target class are excluded from training data before optimization begins. This processing step prior to training was motivated by the large number of features of data in which the method was evaluated. Such filtering might be interpreted as a filter feature selection algorithm, as described in Section 2.9.

Mutation Guided Add Input: p-values output from KW are normalized into probabilities by the transformation $-log_{10}(.)$ and scaled by the softmax function (Equation 2.13), which results in a probability distribution over input features. Such distribution is used to sample disconnected input nodes when choosing a feature to be included in the topology

by the Add Input mutation.

Mutation Swap Input: Another mutation proposed in N3O is the Swap Input, which attaches neighbors of an input node present in an individual's topology to a disconnected input. Such a step is a random and unbiased manner of exploring a part of the search space that is not guided by the Kruskal-Wallis one-way analysis of variance, and is depicted in Figure 2.7.

Crossover Modification: Originally on NEAT crossover, resulting offspring do not inherit excess genes concerning the least fit parent. Grisci, Feltes and Dorn (2018) proposes that the least fit parent's input nodes that are connected to nodes present in the fittest parent's topology have a chance to be inherited during the crossover. This alteration is motivated by the intuition that combining feature selection from multiple promising networks yields a better selection. The restriction that only inputs that fit the usually inherited genome is principled, as it minimizes the topological extension of the most fit parent.

2.4 k-nearest neighbors

The k-nearest neighbors algorithm (KNN) is a computational method that outputs, given a query sample $x_i \in X$, a set of samples Y representing the k points in X most similar to x_i, where $x_i \notin Y$. The method is parametrized by the integer k, a set of data points X, and a similarity function $f : X^2 \to \mathbb{R}$ which compares samples from X. Due to the flexibility of the similarity function's definition, the method can handle both numerical and categorical data points. KNN is commonly applied to classification tasks (NOI; KAPPAS, 2017; HAN; KARYPIS; KUMAR, 2001), where an unlabeled sample is queried and its labeled nearest neighbors are considered to infer the queried sample's class. The algorithm is also employed on data imputation, in which a sample containing missing or invalid values is queried, and the values of its neighbors are used to estimate attributes that are not valid (BATISTA; MONARD, 2003).

2.5 Support vector machine

Support vector machine (SVM) is a supervised machine learning algorithm that is suited for both classification and regression tasks (CORTES; VAPNIK, 1992). It follows the assumption that training data is well separable by hyper-planes residing on the features

space. The method supports the application of non-linear transformations (kernels) to training data to represent original features in a more separable space. SVM naturally handles binary classification and only approaches multi-class classification tasks given algorithmic extensions, such as training an ensemble where each classifier separates a particular class from the others.

Among state-of-the-art classifiers, SVM is computationally inexpensive (in particular when compared to deep neural networks) and presents low generalization error to a broad variety of data sets. Nevertheless, its parametrization is very sensitive, including kernel choice. SVM is essential to this study as it is the classifier that assesses feature selections quality on experimental evaluation.

2.6 Mutual information

Mutual information (MI) is a measure of dependency between categorical variables, defined within information theory (SHANNON, 1948; STEUER et al., 2002; GRAY, 1990). MI (Equation 2.9) is defined in terms of the marginal and conditional probabilities concerning the variables analyzed.

Given two discrete variables X, Y able to respectively assume values in V_X, V_Y, the mutual information between X and Y is defined as:

$$I(X,Y) = \sum_{x \in V_X} \sum_{y \in V_Y} p(x,y) log \frac{p(x,y)}{p(x)p(y)} \tag{2.9}$$

When X, Y are not correlated, $I(X,Y) = 0$. Higher values of $I(X,Y)$ indicate a stronger correlation between input variables.

Mutual information has been utilized on feature selection methods as a strategy for handling categorical inputs (VERGARA; ESTÉVEZ, 2015; BUNNEY et al., 2017). It has also been successfully applied to discretized noisy continuous data (DING; PENG, 2003). The present study adopts MI for assessing input features relevance concerning the target class, as well as measuring input attributes redundancy with respect to each other. Such a strategy is presented in the next section.

2.7 Minimum redundancy - maximum relevance

Minimum redundancy - maximum relevance (mRMR) (DING; PENG, 2003) is a sequential forward feature selection algorithm that ranks and greedily selects a pre-defined number of attributes employing criteria based on Mutual Information (presented in Section 2.6).

Let S be features already selected by the algorithm and let h denote the target attribute. The redundancy and relevance assessment of the selection set S is respectively defined by the following equations.

$$W(S) = \frac{1}{|S|^2} \sum_{i,j \in S} I(i,j) \tag{2.10}$$

$$V(S) = \frac{1}{|S|} \sum_{i \in S} I(h,i) \tag{2.11}$$

One way of composing redundancy and relevance measurements in a single criterion (to be maximized) is the quotient composition, defined by Equation 5.3. (DING; PENG, 2003) presented evidence where such composition outperforms others, such as the difference, in a cancer biomarker selection dataset.

$$\frac{V(s)}{W(s)} \tag{2.12}$$

The search for all selection sets S is computationally intractable, as it is an NP-Complete problem, so a sequential forward feature selection method is proposed on top of the aforementioned criterion. Starting from an empty set of selected features S and the set of all features Ω, mRMR employs a greedy strategy to select remaining features $\Omega_S = \Omega - S$. The attribute that maximizes the chosen criteria composition is added to S. The selection terminates when the pre-defined number of features to be chosen have already been inserted to S.

Since attributes redundancy assessments are defined as a function of S it has to be updated every time a new feature is selected. Whole selection is computed in $O(|S|N)$ steps, N being the number of samples used to infer the relationship between attributes.

2.8 Softmax

The softmax function is a mathematical transformation that normalizes a vector of real-values numbers into a probability distribution, and is defined by Equation 2.13, being $X = X_1, ..., X_n$ the vector to be transformed and X' its normalized output. Such an operation is important, for instance, to transform a vector of ordered scores into a vector of the same size containing non-negative elements and whose $|L_1|$-norm is 1. Therefore resulting elements can be interpreted as probabilities and the whole vector as a probability distribution.

$$X_i = \frac{e^{X_i}}{\sum_{i \in [n]} e^{X_i}} \tag{2.13}$$

2.9 Feature selection

Feature selection (FS) is part of dimension reduction, one of the data preprocessing tasks which are essential to machine learning models construction. While some models, such as Decision Trees, measure feature relevance and use it as induction criteria, most machine learning models assume all features are relevant to the given task. In both cases, the model's induction computational cost increases significantly with the number of features. Spurious or irrelevant attributes might also harm the model's prediction ability, as there is evidence that the same learning algorithm achieves better performance by inducting on a subset of the same data where some of the original features are discarded (UTANS; MODDY; REHFUSS, 1995; DASH; LIU, 1997; CILIA et al., 2019, 2019).

Lazar et al. (2012) divides feature selection algorithms into four categories: filter, wrapper, embedded, and ensemble. Ang et al. (2016) describes a fifth category: hybrid. *Filter* selection methods use statistical measures to analyze features relevancy w.r.t target data. These algorithms do not depend on any classifier to perform the selection, therefore they represent the category that induces less bias. Filter methods are also computationally less expensive than other feature selection methods, and some do not rely on labeled data as they focus on redundancy criteria. An example of a filter method is univariate feature selection (UFS) (AGGARWAL et al., 2014), in which attributes are ranked given a criterion and the best are retrieved. Features are ranked in a univariate manner, as in the correlation of each input feature concerning the target class is measured independently.

The employed criterion and the number of features to be retrieved are parameters to this algorithm. Univariate feature selection is relevant to this study as its selection quality is compared to the proposed method.

Wrapper methods perform a heuristic search on the space of subsets of the original feature set using a predefined evaluator to measure the quality of candidate subsets. The evaluation of any point in search space requires the induction and validation of a machine learning model, which is computationally expensive. Wrappers selection output suffers from any bias induced by the base evaluator, which diminishes the interpretation of selected features as the relevant features to the task. The only restrictions for the quality assessment of proposed search points are the ones presented by the base evaluator, enabling the whole FS algorithm to be applied to unsupervised and reinforcement learning tasks. An example of a wrapper feature selection algorithm is the utilization of Recursive Feature Elimination (RFE) and Support Vector Machines (SVMRFE) (GUYON et al., 2002), in which SVM is the classifier employed to assess the quality of inputs subsets.

Embedded methods perform feature selection by construction while approaching the prediction task. A classical example is Decision Trees as it computes attributes' relevance and redundancy, and might ignore some of the features initially present on training data. Another embedded feature selection algorithm is FS-NEAT (WHITESON et al., 2005) which is discussed in detail in Section 2.3.2 as it is a basis for the method proposed in this study. SVMRFE is classified as an embedded approach as well, since it utilizes RFE, which is a filter method, to propose selections to SVM in a wrapper-based approach.

Ensemble algorithms handle the sensitivity of most existing methods to noise present in data. An FS algorithm is applied to sampled subsets of original data, and results are aggregated into a more reliable selection output.

Hybrid feature selection composes methods of at least two of the aforementioned categories. They present strong potential since they combine already established methods strengths in order to diminish their weaknesses.

Filter methods are commonly used appended to hybrid frameworks (GRISCI; FELTES; DORN, 2018; AGGARWAL et al., 2014) due to their low computational cost. Their application is preferable on learning tasks that present a large number of features, or on data domains where correlation present in data is straightforward. Wrapper methods are best applicable when the main objective is predictive accuracy and the search space is reasonably small. Embedded feature selection algorithms are less common since they perform (most times implicitly) multi-objective optimization which is harder to con-

struct. They are not applicable when data violates the embedded selection assumptions. Ensemble FS is employed when data present variance that is harmful to the chosen base selector. All mentioned categories may handle unlabeled data, although filter and embedded methods commonly rely on the assessment of features relevance concerning a target class.

2.10 Cross-validation

A problem common to learning algorithms is over-fitting (LAWRENCE; GILES, 2000; BELKIN et al., 2019), characterized by the internalization of statistical patterns lying in training data that does not represent the natural data distribution. Such patterns are usually introduced by data collection bias, for instance, when collected samples represent a particular sub-population of data that is to be analyzed. This problem is detected by testing induced learning models on test samples not present in training data, originated by a different sampling procedure. Nevertheless, some predictive tasks (especially real-world ones) are described by a single data set or collection methodology, which forces the algorithm to learn on data subject to the same bias. A strategy for dealing with this problem is k-fold cross-validation (CV), where available data is partitioned into k equally sized sets called folds. Every fold is used to test the induced model's predictive capability as the same learning algorithm runs using the remaining $k - 1$ folds as training data. Finally, k learning models are induced and tested on different partitions of the available data and computed metrics are aggregated, usually by their mean and standard deviation. When data is partitioned into folds that respect the original class distribution, the experiment is called **stratified** cross-validation.

2.11 Confusion matrix

The confusion matrix is a square matrix that summarizes a given model's predictions by relating the ground-truth of annotated samples with its predicted class (SOKOLOVA; JAPKOWICZ; SZPAKOWICZ, 2006). The diagonal represents successful predictions, while non-diagonal elements account for samples in which the model failed to predict the correct class. The sum of all matrix entries amount to the total number of predicted samples. The confusion matrix is a common way of visualizing machine learning models

classification performance, although it becomes cumbersome for multi-class classification problems.

Table 2.1 depicts a confusion matrix for binary classification. In binary classification tasks, in which one class is taken as positive and the other as negative, the 4 matrix entries have a distinct meaning and serve as a basis for common machine learning metrics (FLACH, 2003; SOKOLOVA; JAPKOWICZ; SZPAKOWICZ, 2006). *True positives* (TP) represent predictions in which samples were annotated as positive and the model predicted the positive class; *False positives* (FP) are positive predictions on samples annotated as negative; *False negatives* (FN) are negative predictions on samples annotated as positive; and *True negatives* (TN) represent negative predictions on negative samples.

Table 2.1: Example of confusion matrix. Columns indicate predicted classes and rows indicate annotated classes. TP, FP, FN and TN are abbreviations for true positive, false positive, false negative and true negative, respectively.

	P	**N**
P	TP	FN
N	FP	TN

Accuracy is a common classification metric derived from a given confusion matrix. It does not distinguish errors between different classes and is defined by Equation 2.14. Precision and recall are metrics that focus on evaluating the model's performance concerning a particular class, and are respectively defined by Equations 2.15 and 2.16. The recall is also mentioned as sensitivity and is widely employed on biological, medical and image processing studies (SOKOLOVA; JAPKOWICZ; SZPAKOWICZ, 2006).

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.14}$$

$$precision = \frac{TP}{TP + FP} \tag{2.15}$$

$$recall = \frac{TP}{TP + FN} \tag{2.16}$$

2.12 Cross-entropy

The cross-entropy function (also referred to as log loss) measures the classification error of a neural network prediction concerning a target class. Such a measure is also applied to multi-class classification, by aggregating prediction errors for each class through summation (BOER et al., 2005; ALPAYDIN; JORDAN, 1996). It is defined by Equation 2.17, where C is a partition of the training set, each element $c \in C$ contains training samples belonging to class c, y is binary value denoting that training sample x is in fact labeled concerning c, and $p(x)$ represents the network prediction for a c-annotated sample x.

$$CE(p, C) = -\frac{1}{|C|} \sum_{c \in C} \sum_{x,y \in c} ln(p(x))y + ln(1 - p(x))(1 - y) \tag{2.17}$$

2.13 Chapter conclusion

This chapter presented the theoretical foundation researched while developing the presented study, as well as computational methods employed on experiments. The next chapter introduces the problem approached on the experimental evaluation, human phenotype prediction.

Figure 2.5: **Depiction of crossover operator.** At the top, two individuals depicted by their phenotype; at the bottom, one of the possible offspring generated by the crossover operator. The leftmost topology is owned by the most fit parent. Dashed arrows represent disabled connections and numbers represent genes' historical markers. Blue-colored circles and edges represent matching genes in I_1, while green-colored circles and edges represent matching genes in I_2. Yellow represents disjoint genes and red represents exceeding genes. In this example, matching genes with markers 1, 3 and 6 are inherited from the least fit parent, while genes with markers 2, 4, 5, 7 and 8 are inherited from I_1. Gene 5 could be enabled because it is an active connection in I_2, although its properties are inherited from I_1.

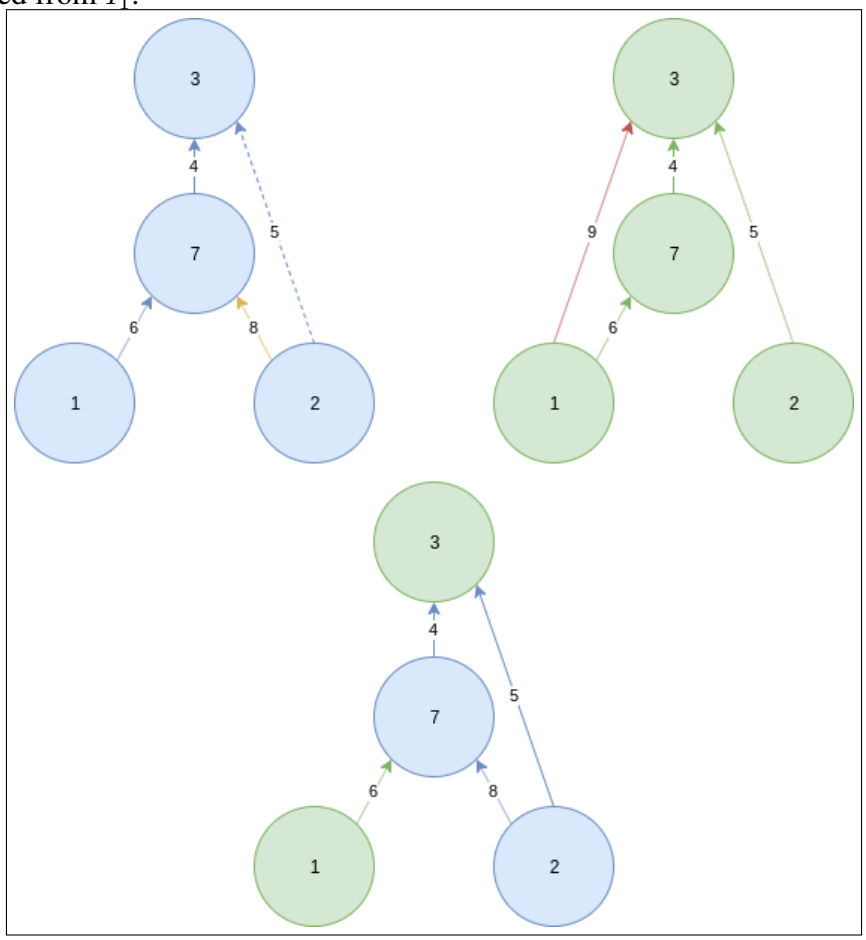


Figure 2.6: **Depiction of mutation Add Input.** An input node, not present in an individual's genome, gets connected to a hidden or output node.

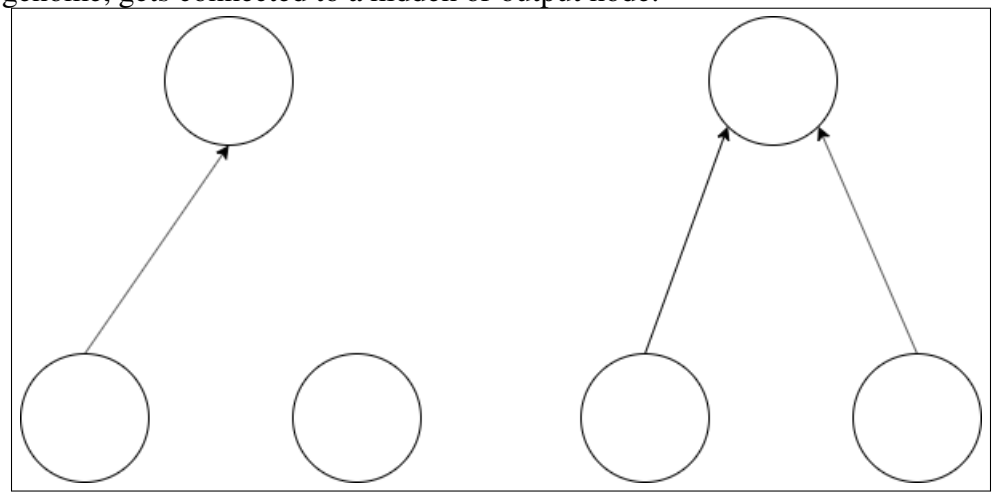
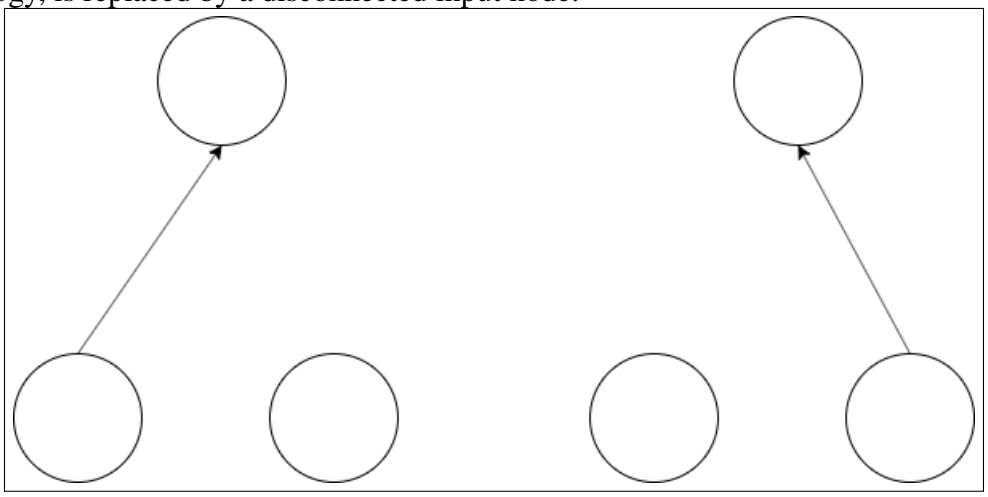


Figure 2.7: **Depiction of mutation Swap Input.** An input node, present in an individual's topology, is replaced by a disconnected input node.



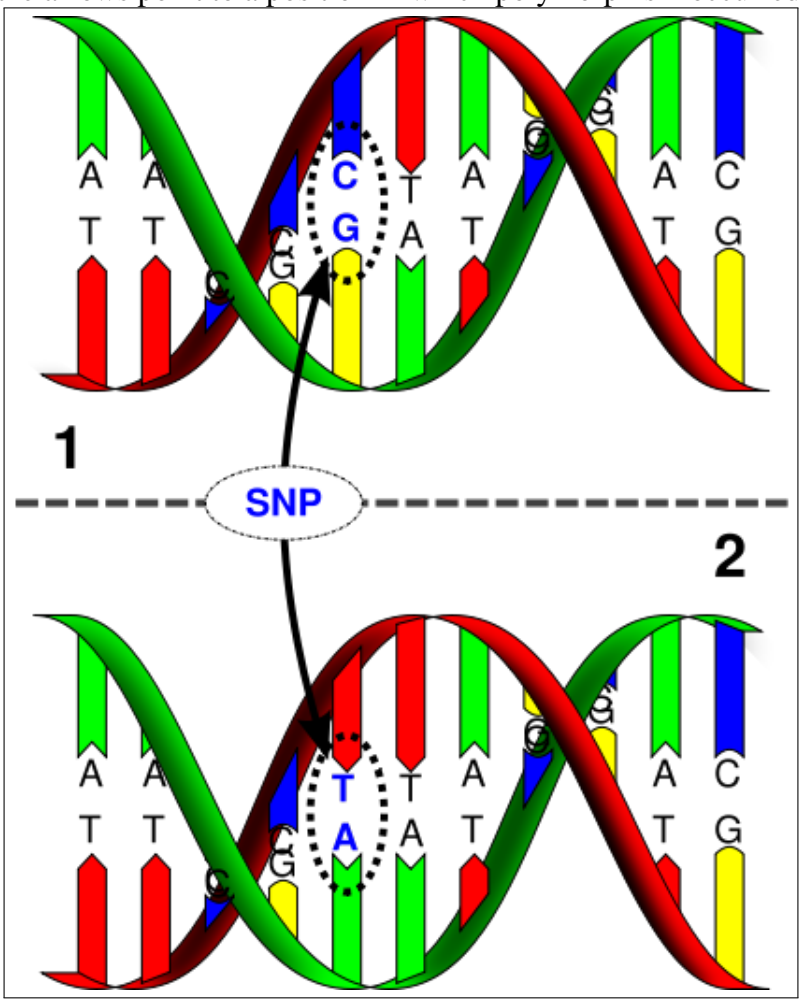
3 THE HUMAN PHENOTYPE PREDICTION PROBLEM

An open problem in biology is the prediction of phenotypical characteristics given genotypical data. Applications for such tasks include animal improvement (CALUS; VEERKAMP, 2007) and control, forensics (JOBLING; GILL, 2004; OGDEN; LINACRE, 2015; WILLIAMS; WIENROTH, 2017; ARENAS et al., 2017) and study of disease genetics (GRISCI; FELTES; DORN, 2018; DING; PENG, 2003; LAZAR et al., 2012; LOPEZ-RINCON et al., 2018). Available data for such tasks typically contain few samples and, at early stages of preprocessing, many dimensions are reduced mostly through statistical filters. Gene selection is an example of the employed dimension reduction techniques and is a particular type of genomic selection, where each input dimension corresponds to information derived from a whole gene.

Since interactions between genes themselves are too complex to be analyzed purely by biological and statistical methodologies, machine learning has been applied to problems derived from biological data (GRISCI; FELTES; DORN, 2018; MATUKUMALLI et al., 2006; LAZAR et al., 2012). It is important to notice that the genotype-phenotype relationship is susceptible to significant variations on different populations (AVILA et al., 2019; ZAORSKA; ZAWIERUCHA; NOWICKI, 2019; OGDEN; LINACRE, 2015), therefore results obtained from statistical or machine learning methods may be tightly coupled to the population represented in data.

While most of an individual's genes are similar to others from the same species, nuclear deoxyribonucleic acid (DNA) of autosomes presents reasonable difference for approaching identification by sequencing genome data (CORTE-REAL; VIEIRA, 2015). Single nucleotide polymorphisms (SNPs) are powerful biomarkers for capturing distinctions between individuals. Recent work showed that SNPs are informative to kinship determination (AVILA et al., 2019) and prediction of complex phenotypes, such as skin and eye color (ZAORSKA; ZAWIERUCHA; NOWICKI, 2019; LIU et al., 2009; WHITE; RABAGO-SMITH, 2011). The present work follows this evidence and applies the proposed method to an SNPs dataset to construct an informative biomarkers panel. The following section discusses how previous work related genotypical data analysis relates to feature selection, a core theme in this study.

Figure 3.1: **Depiction of a single nucleotide polymorphism.** While most of the genome between individuals of the same species is similar, particular locations on the nucleotide sequence present polymorphism. The letters in the image indicate nucleotides in the genome and the arrows point to a position in which polymorphism occurred.



Picture from Academia.edu (2015)

3.1 Genomic selection

The term genomic selection is employed when referring to a selection of attributes that correspond to information derived from biological genes (ANG et al., 2016). Figure 3.2 depicts how data obtained from different analysis techniques are useful when comparing genomes with a different granularity of genetic distance.

It is important to highlight that there is much uncertainty in biological data and its collection is a complex task, prone to error from diverse sources. Such errors might introduce bias on available data, which in turn yield prediction bias when applying machine learning algorithms to it. Therefore genomic selection results should be carefully validated by domain experts in order to reject erroneous results due to systemic and statistical

bias possibly introduced during data collection and processing (FELTES et al., 2019).

Figure 3.2: **Depiction of distinct data derived from genomic samples.** When comparing genomes, different reads on samples are suited for different magnitudes of genetic distance.

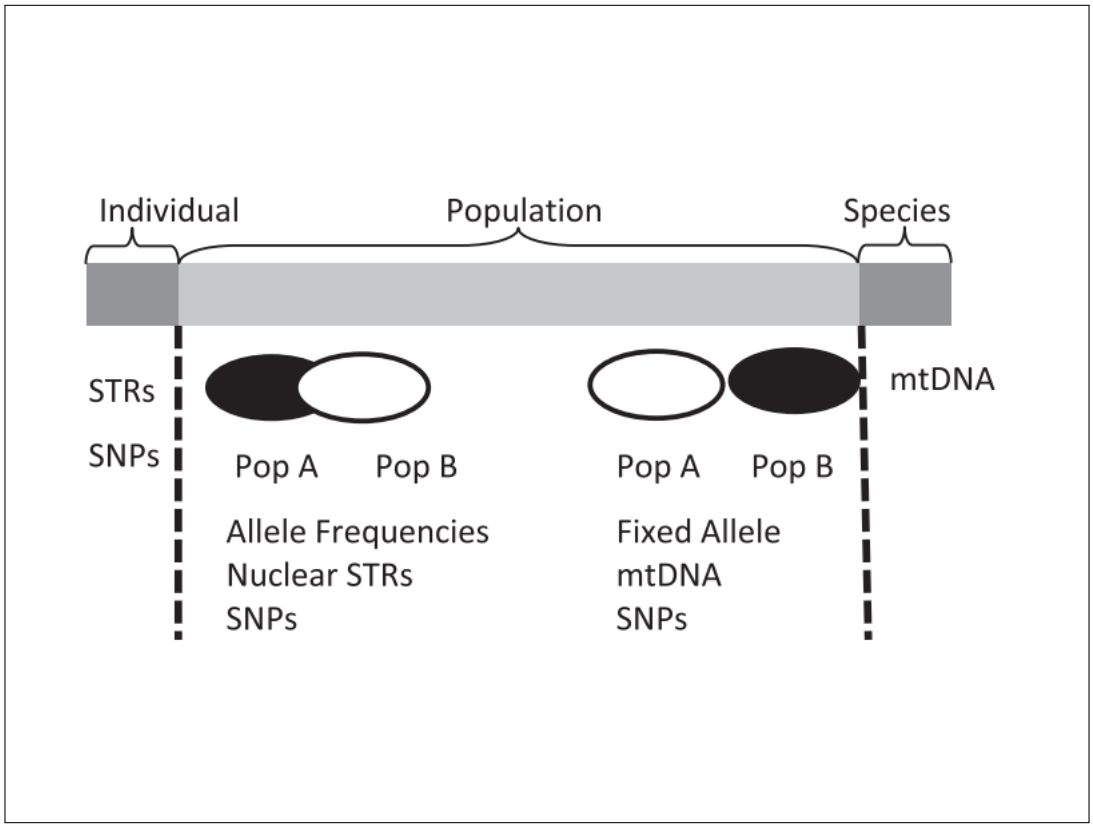


Figure from Ogden and Linacre (2015)

One important application of gene selection is the construction of biomarkers panels (AVILA et al., 2019; OGDEN; LINACRE, 2015; WILLIAMS; WIENROTH, 2017; ARENAS et al., 2017). Once relevant markers to a given task are discovered, future data collections might focus on the selected features only. The panel description not only improves data analysis capability (as it is the case in general feature selection) but cheapens the gathering and storage of new data. On animal improvement, for instance, data collection equipment must be distributed to farms along a possibly wide-area. In this case, spurious markers increase the cost of individual collection devices, dampening the process of obtaining data on large scale.

The application of feature selection to SNPs data is also mentioned as genomic selection, although the relation between single nucleotide polymorphisms and genes is not straightforward. An SNP represents less information than a gene, but it does not always reside inside coding regions of the genome. SNPs are often related to genes by physical proximity only. Furthermore, single nucleotide polymorphisms spread throughout the

entire genome and may not be related to any known gene.

3.2 Chapter conclusion

This chapter introduced the human phenotype prediction problem and related it to feature selection, a core theme in this work. The following chapter presents the proposed algorithm in detail, emphasizing its essential differences to the already established feature selection methods discussed in Chapter 2.

4 THE PROPOSED METHOD

N3O whole modifications to FS-NEAT rely on the Kruskal-Wallis one-way analysis of variance (Equation 2.6) which is inappropriate for categorical data. To overcome this restriction, the statistical test employed in N3O to guide the addition of new inputs is seen as a component that ranks features, similar to the scoring employed in filter feature selection methods. The present study leans on this abstraction to propose a novel feature selection algorithm based on neuroevolution which is specialized in categorical data, N3O-D.

In order to approach discrete input data, the criteria proposed in the mRMR method (Equation 2.12) is chosen to rank features that are not selected within individuals' topologies. Instead of greedily picking the feature with the higher score, remaining features scores are normalized to probabilities using the Softmax function (Equation 2.13), and then fed into the evolutionary algorithm mutation process through the mutation *Guided Add Input* (presented in Section 2.3.3). Such criterion is appropriate for categorical data since it interprets entries as distinct labels, as opposed to numerical quantities. It is appropriate for feature selection since it measures the relevancy of a given feature concerning the target class as well as the given feature redundancy concerning already selected attributes. In contrast to N3O attribution of probabilities to features, N3O-D needs to recompute probabilities every time a new feature is added to the individual topology (see Equation 2.10). Like most machine learning algorithms, the proposed method does not handle missing values naturally.

The following section presents a processing step that is extraneous to N3O's proposed modifications and is necessary due to the discrete nature of data only. The remaining sections detail a pipeline as an application example of the proposed method and conclude the chapter.

4.1 Encoding

For neural network topologies to distinguish different input values in a discrete manner rather than in a continuous one, categorical features should be strategically encoded into numerical values (POTDAR; TAHER; CHINMAY, 2017). Cedric (2018), Potdar, Taher and Chinmay (2017) presented evidence where one-hot encoding showed superior performance on comparative studies concerning other techniques and therefore

was chosen to encode input features. Such transform artificially increases the number of input variables and creates a relationship between input nodes. A mapping between the original attribute and its induced input nodes is kept, so feature selective variations of NEAT are capable of treating sets of input nodes as a whole attribute. Such mapping is crucial to the gene selection task, once the objective is to select whole markers rather than parts of its information. Figures 4.1 and 4.2 depict how some of N3O original mutations change when one hot encoding is applied.

Figure 4.1: **Depiction of mutation Add Input given one hot encoding of input nodes.** Inputs colored the same way belong to the same original input feature and therefore must be selected as a group.

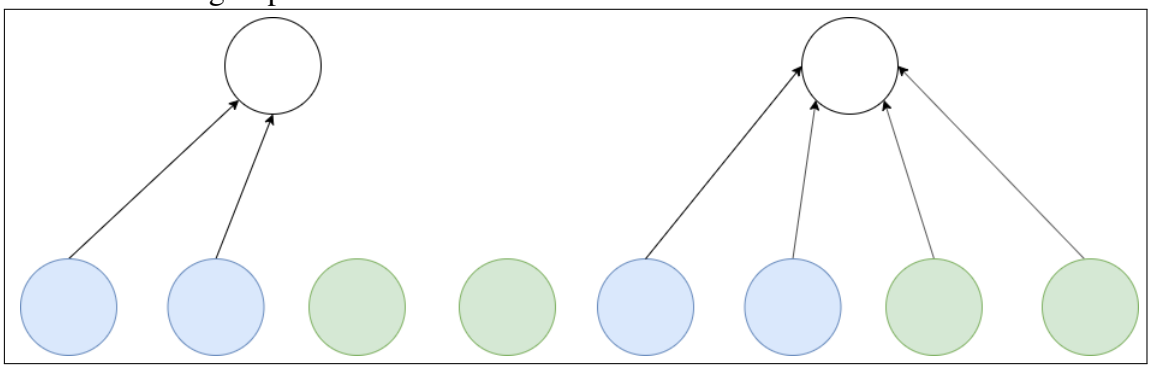
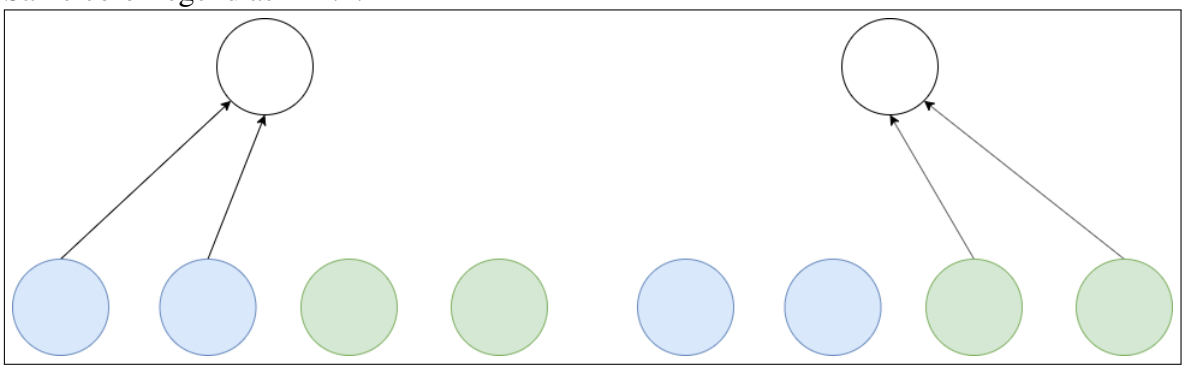


Figure 4.2: **Depiction of mutation Swap Input given one hot encoding of input nodes.** Same color legend as in 4.1.

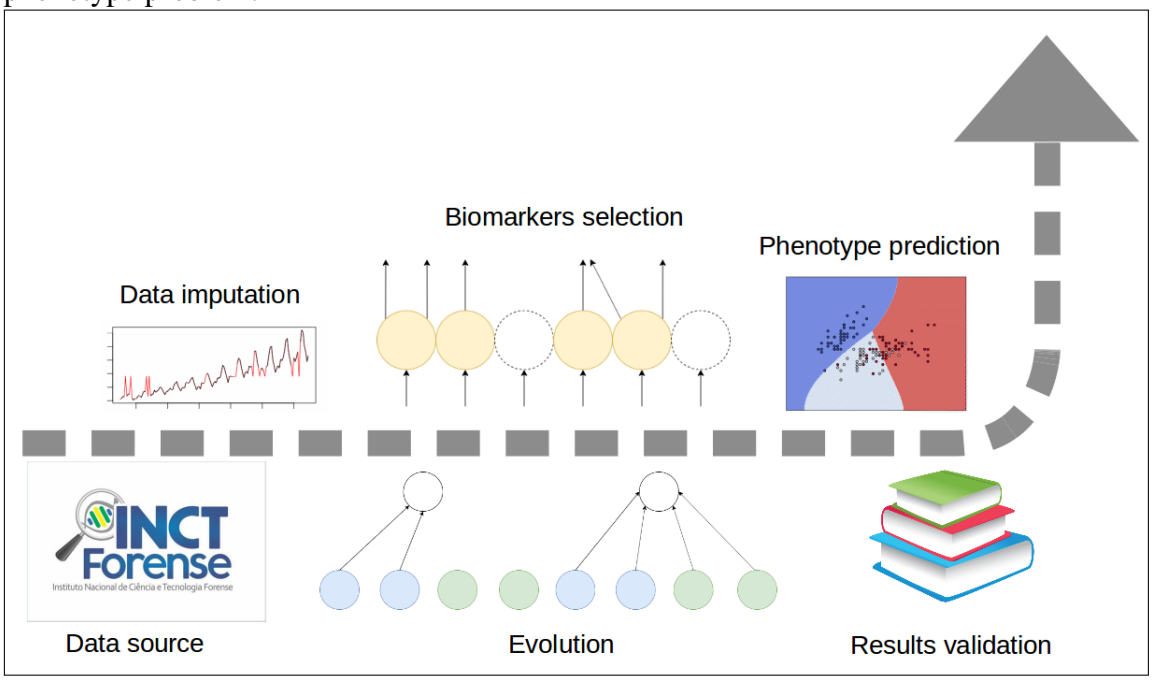


4.2 Pipeline example

Figure 4.3 depicts the pipeline employed on the experimental evaluation presented in Chapter 5 as an application example for N3O-D.

Data source: Data is provided by particular institutions that hold the technical knowledge required for collecting and storing genotypical data, as well as applying the first

Figure 4.3: Diagram depicting how the proposed method is applied to the genotype-phenotype problem.



preprocessing and filter stages.

Data imputation: Real-world data sets often contain missing or invalid values. The process of handling such values is called imputation and is a necessity given incomplete data.

Neuroevolution: Imputed data is fed onto the evolutionary algorithm which simultaneously selects relevant inputs and classifies samples according to their annotated phenotype.

Results validation: Feature selection results yield candidates for an SNPs panel. The relevancy of the selected biomarkers should be evaluated by the biology community. Appropriate validation might indicate methodology flaws in the conducted experiments that generated such a selection or inspire novel biological research. A simple validation of the SNPs selected on experimental evaluation is detailed and presented in Chapter 5.

4.3 Chapter conclusion

This chapter presented the feature selection method proposed by this study, as well as explained its necessity by highlighting weaknesses on the algorithms discussed in previous parts of the text. The following chapter is dedicated to the employed experimental evaluation. It introduces provided data and details the treatments applied to it, announces the evaluated algorithms while specifying their configurations, and finally presents and

discusses obtained experimental results.

5 EXPERIMENTS AND RESULTS

The prediction performance of N3O-D is experimentally compared on the human phenotype prediction problem to its base algorithm, FS-NEAT, which is also powered by neuroevolution, as well as a univariate feature selection that also assesses inputs relevance via mutual information. The prediction performance of neuroevolution algorithms is assessed through 10 evolution runs of 3-fold cross-validation. Feature selections of such methods are retrieved by picking the inputs selected by the best performing individual in each evolution, collected over the 30 executions. Retrieved features are ranked by the frequency in which they are selected throughout the multiple runs. The quality measurement of retrieved feature selections is better detailed in Section 5.2.

The following sections describe the data utilized in experiments, the feature selection methods evaluated, how N3O-D hyper-parameters were configured, and presents experimental results.

5.1 Data

INCT Forense [1] provided a data set with 439 samples of 67 single nucleotide polymorphisms (SNPs) genetic markers labeled with both eye and skin color phenotypical information, respectively showed in Figures 5.1 and 5.2. Table 5.1 depicts the data format. The values distribution for both target characteristics is presented in Tables 5.2 and 5.3.

Table 5.1: Raw data examples

ID	SNP_1	SNP_2	SNP_3	$phenotype_{eye}$	$phenotype_{skin}$
1	AA	CT	GA	blue	dark
2	AC	CT	..	green	pale
3	.C	CC	AG	hazel	light

Columns that represent SNPs (features) are represented by two nucleotides, one for each DNA strand at the particular genome position. Missing values might occur in the whole marker or partially. The last columns represent phenotype occurrences and depict that their values are not necessarily encoded for learning algorithms in a proper manner. Samples are identified by a specific column (the first, in this example), although this information is ignored when training machine learning models.

[1] <https://inctforense.wordpress.com/>

Figure 5.1: Picture of analyzed skin phenotypes.

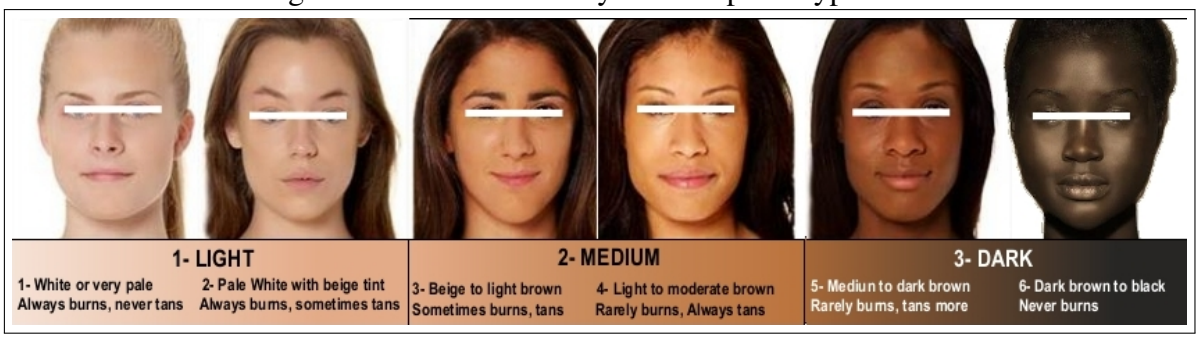


Image provided by INCT Forense

Figure 5.2: Picture of analyzed eye phenotypes.

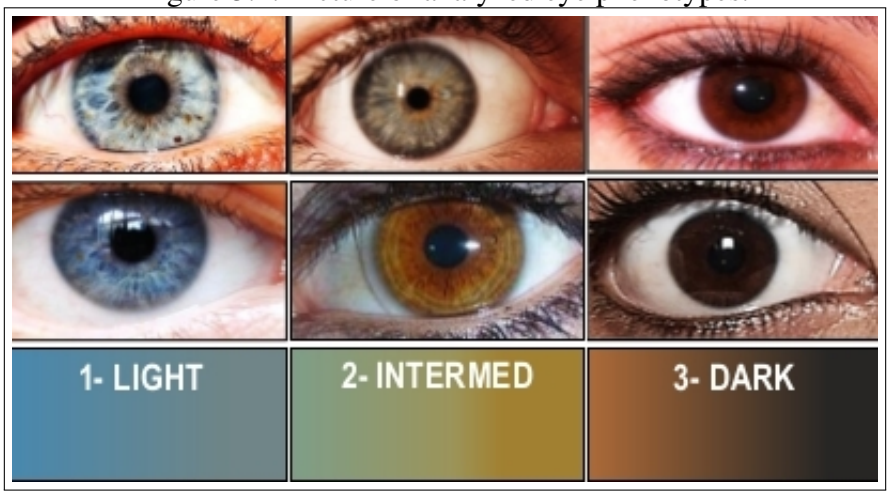


Image provided by INCT Forense

Table 5.2: Skin color class distribution

Classes	Representation (%)
White	25.79
Pale	31.73
Beige	14.38
Light Brown	9.58
Medium Brown	12.32
Dark Brown	6.16

Table 5.3: Eye color class distribution

Classes	Representation (%)
Blue	23.91
Green	8.2
Hazel	6.6
Light Brown	9.33
Dark Brown	37.12
Black	14.80

The provided data was not readily applicable as input to machine learning algorithms since it contained invalid entries. Class distributions are imbalanced, which is prejudicial for learning models. The next section outlines how these issues are approached throughout data preprocessing.

5.1.1 Imputation

Samples of provided data contained corrupt measurements for some SNPs. Since neither SVM nor NEAT handles invalid values, imputation had to be performed. The inference of invalid attribute values was employed through KNN. Samples containing missing values are queried using Equation 5.1 and the values to be imputed are estimated as the mode of retrieved neighbors. The described imputation technique is inspired by Batista and Monard (2003). Although mode imputation induces bias on learning algorithms, it was preferred over the exclusion of samples or features with invalid values since most attributes contained at least a few samples with incomplete entries.

$$D(s_1, s_2) = \frac{\sum_{a \in A_{s_1} \cap A_{s_2}} D^a(s_1, s_2)}{|A_{s_1} \cap A_{s_2}|} \tag{5.1}$$

$$D^a(s_1, s_2) = \begin{cases} 1, & \text{if } s_1^a = s_2^a \\ 0, & \text{otherwise} \end{cases} \tag{5.2}$$

$D(s_1, s_2)$ represents the difference of two distinct samples s_1 and s_2, s_k^a denotes value of attribute a for the k-th sample, A_{s_k} is the set of attributes of s_k that contains valid values, and $\cap$ denotes set intersection.

5.1.2 Data views

As Tables 5.2 and 5.3 showed, class distributions are imbalanced for both phenotypical characterizations, which encourages the specification of data views that aggregate original classes to construct a more homogeneous annotation of samples. Class imbalance is detrimental to classification algorithms as less representative classes do not contribute much to the patterns extracted to guide the model's induction, which results in models with a bias towards the classes that are most represented in training data. Furthermore, phenotype prediction use cases do not always require a distinction as refined as those presented originally. For instance, when identifying an individual with blue colored eyes, distinguishing between the multiple brown shades is irrelevant, hence the *Blue*-vs-*Non-Blue* view (presented in Table 5.5). Each data view represents a function that transforms the provided dataset into one more adapted to different applications by aggregating particular classes into significant partitions. Such transformations are capable of diminishing class imbalance as underrepresented classes may be merged into a more representative one. The attributed views for skin and color characterizations are respectively specified in Tables 5.4 and 5.5.

Table 5.4: Skin color data views

Partition
White, Pale, Beige, Light Brown, Medium Brown, Dark Brown
Light (White and Pale), Intermediate (Beige and Light Brown), Dark (Medium and Dark Brown)
Light (White, Pale and Beige), Dark (Light, Medium and Dark Brown)

Table 5.5: Eye color data views

Partition
Blue, Green, Hazel, Light Brown, Dark Brown, Black
Blue, Intermediate (Green, Hazel and Light Brown), Dark (Dark Brown and Black)
Blue, Non-Blue
Brown (Light Brown, Dark Brown and Black), Non-Brown
Light (Blue and Green), Non-Light

5.2 Feature selection quality assessment

Since the main objective of the proposed method is to select inputs on training data, a quality measurement of retrieved selections is established and detailed in the present section. As a strategy for evaluating the contribution of the three operators proposed in Grisci, Feltes and Dorn (2018) on the proposed adaptation to categorical data, feature selections quality and classification performance of FS-NEAT are also assessed on target data. As an alternative to neuroevolution, mutual information (Section 2.6) is utilized in a univariate feature selection algorithm (explained in Section 2.9) to measure the contribution of input attributes to the task in hand. This method is mentioned as UFS(MI) in the following sections.

Given a rank of the available input dimensions, feature subsets are greedily formed by selecting the top-k attributes. $k \in [1, 10]$ was chosen as it is a typical use case specified by the data provider. Obtained selections serve as input to a cross-validation training, employing SVM as the classifier, whose aggregated accuracy indicates how excellent is the analyzed feature selection. For each evaluation replication, the selection with maximal accuracy is retrieved, preferring selections of minimal size in the case of ties. Since the classification performance concerning all phenotype classes should be accounted for, the terms in Equation 2.14 are adapted as follows: the numerator is replaced by the sum of true positive predictions concerning each target class and the denominator is the number of evaluated predictions.

On the univariate selection method, the ranking of attributes is an algorithmic by-product, and retrieved features are already ranked. Since UFS(MI) ranking is not deterministic, its selection capability is evaluated on 50 3-fold CV runs to mitigate its ranking variance. On methods based on neuroevolution, the attributes rank is computed by measuring the frequency in which available SNPs were selected throughout 30 executions, which resulted from the 10 3-fold cross-validation runs employed to evaluate the algorithms' classification performance through evolution alone (discussed at the beginning of this chapter). After the 30 executions, a single attribute rank is computed per neuroevolution algorithm, therefore a single CV is sufficient to measure the selection quality.

5.3 Baseline accuracy

Prior to any classification, a baseline accuracy might be established. It is defined by Equation 5.3 (where D denotes the whole training data set, C denotes target classes occurring in D, and D_c corresponds to samples in D annotated with class c) and translates to the percentage of samples annotated with the most representative class in training data. A classifier that matches the baseline accuracy is trivial to implement, as it constantly outputs the class represented by most samples, and finding such a class requires a single iteration on the data set. As in Grisci, Feltes and Dorn (2018), the baseline accuracy is compared to the classification performances obtained through experimental evaluation.

$$A(D) = max_{c \in C}\{\frac{|D_c|}{|D|}\} \tag{5.3}$$

5.4 Proposed method configuration

As an extension of FS-NEAT, the proposed method requires the specification of multiple hyper-parameters that heavily influence the optimization performance. As already discussed in Section 2.1, genetic algorithms provide a lot of room for micro-optimizations without violating the algorithm specification. References that propose neuroevolution of augmenting topologies (STANLEY; MIIKKULAINEN, 2002; WHITESON et al., 2005) present vague descriptions on how key components of the meta-heuristic should be implemented, especially how individuals are partitioned into species. It is important to highlight that existing implementations of NEAT employ extensions that are not prescribed by the original algorithm specification, and are beyond the scope of this study. Nevertheless, it is important to note that such discrepancies harden the comparison with previous results.

The following sections detail the hyper-parameters employed on experimental evaluation, which were selected based on literature revision rather than empirically. The feed-forward neural networks induced by individuals' phenotypes, the function employed to assess individuals' fitness and the evolution hyper-parameters are also specified.

5.4.1 Fitness function

An imbalanced version of the cross-entropy (presented in Section 2.12) is utilized as fitness function and is presented in Equation 5.4, where symbols' interpretation is the same as in Equation 2.17. The term $\frac{1}{|c|}$ weights the error induced per sample inversely proportional to its class representation within the training data. It was employed on Grisci, Feltes and Dorn (2018) to diminish the error estimation bias on imbalanced data sets.

$$CE(p, C) = -\frac{1}{|C|} \sum_{c \in C} \frac{1}{|c|} \sum_{x,y \in c} ln(p(x))y + ln(1 - p(x))(1 - y) \tag{5.4}$$

$$CE(p, C) + \lambda \frac{1}{2n|E|} \sum_{\theta \in \Theta} \theta^2 \tag{5.5}$$

The fitness of an individual whose induced phenotype computes p via weights Θ concerning a training set partitioned by C is defined by Equation 5.5, where E represents the set of edges within the individual's topology and n denotes the total number of samples in training data. The second term in the expression regularizes networks weights during optimization, under the motivation that connections and biases with small magnitude better generalize the patterns learned in training data. The employment of regularization turns the optimization into a multi-objective problem, whose objectives are weighted by the hyper-parameter λ. Since population individuals possibly yield a different number of edges, the regularization term is scaled by $\frac{1}{|E|}$ to avoid penalizing large networks that evolved useful structure.

In contrast to the experiment in Grisci, Feltes and Dorn (2018), this study's neural networks output a probability estimate for each possible occurrence of the target attribute, rather than classifying one class at a time. Such a decision is made upon the soft meaning attributed to label values. Furthermore, the goal of this study is to find a unique set of biomarkers able to explain the distinction between phenotypes, rather than characterizing a particular one. The explanation of specific phenotype occurrences is somehow approached by data views (discussed in Section 5.1.2).

5.4.2 Activation and aggregation functions

Neural networks neurons require the specification of activation and aggregations functions in order to compute their output. In this experiment, N3O-D was configured with activation functions that were highlighted in comparative studies (PAPAVASILEIOU; JANSEN, 2018). The hyperbolic tangent and the Gaussian function (with 0 mean and standard deviation of 1) are used by the network hidden and output nodes, respectively, and are defined by Equations 5.6 and 5.7. As in Grisci, Feltes and Dorn (2018), the mean was chosen as aggregation function to compensate for nodes with a different number of incoming edges.

$$\phi(x) = tanh(4.9\frac{x}{2}) \tag{5.6}$$

$$\phi(x) = e^{-\frac{5x}{2}} \tag{5.7}$$

5.4.3 Hyper-parameters

Neuroevolution algorithms based on NEAT requires the specification of several hyper-parameters that guide the evolution. The configuration chosen for this experiment is presented in table 5.6, where $\mathcal{N}$ denotes the normal distribution. It was based on the previous successful results of Grisci, Feltes and Dorn (2018), except c_3 and the population size, whose values were obtained from Whiteson et al. (2005). The parametrization of SVM was obtained via grid search optimization and is specified by Table 5.7.

Table 5.6: Chosen hyper-parameters for the proposed method configuration

Parameter	Value
Population size	100
Number of generations	100
Elitism proportion	10%
Inter-species mating rate	1%
Coefficient 1 (c_1)	1.0
Coefficient 2 (c_2)	1.0
Coefficient 3 (c_3)	3.0
Chance mutation Add Node happens	3%
Chance mutation Add Connection happens	5%
Chance mutation Change Weight happens	4%
Chance mutation Add Input happens	5%
Chance mutation Swap Input happens	5%
Chance disabled gene becomes enabled on crossover	50%
Chance input from least fit parent is inherited	75%
Initial weights distribution	$\mathcal{N}(0, 1)$
Initial biases distribution	$\mathcal{N}(0, 1)$
Weight mutation power	0.03
Bias mutation power	0.03
Regularization coefficient	0.1

Table 5.7: Chosen hyper-parameters for SVM grid search optimization

Parameter	Grid Coordinates
C	0.001, 0.01, 0.1, 1, 10
gamma	0.001, 0.01, 0.1, 1, scale, auto [2]
kernel	RBF

5.5 Results and discussion

The following tables present experimental results concerning input features ranks retrieved by the evaluated methods, as detailed in Section 5.2. Tables 5.8 and 5.9 present the size of selections retrieved by the 3 evaluated FS algorithms.

Table 5.8: Evaluated feature selections sizes on skin color data views

Data View	UFS(MI)	N3O-D	FS-NEAT
White, Pale, Beige, Light Brown, Medium Brown, Dark Brown	**5.92 ± 1.84**	10 ± 0	6.33 ± 2.86
Light, Intermediate, Dark	**5.09 ± 2.47**	6 ± 2.82	7.33 ± 3.77
Light, Dark	**6.27 ± 1.97**	8.66 ± 0.94	6.66 ± 2.62

Table 5.9: Evaluated feature selections sizes on eye color data views

Data View	UFS(MI)	N3O-D	FS-NEAT
Blue, Green, Hazel, Light Brown, Dark Brown, Black	7.35 ± 2.79	7.66 ± 3.29	**6.66 ± 2.05**
Blue, Intermediate, Dark	7.84 ± 1.83	**6.66 ± 2.86**	10 ± 0
Blue, Non-Blue	**1.98 ± 2.74**	6 ± 2.16	9 ± 1.41
Brown, Non-Brown	**1.7 ± 1.98**	2 ± 0	8 ± 0.81
Light, Non-Light	**1 ± 0**	4.33 ± 3.29	9 ± 1.41

Tables 5.10, 5.11, 5.12 and 5.13 make explicit the SNPs selected by the evaluated feature selection methods, as well as their occurrence frequency throughout the multiple

[2] *scale* and *auto* infer *gamma* concerning input features, respectively attributing $var(X) * M^{-1}$ and M^{-1} (where var denotes variance, the statistical measure) (<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>).

execution runs. SNPs that were related to the target phenotypes on previous studies are marked with *. For visualization purposes, SNPs selected with a frequency lower than 5% are not presented.

Table 5.10: SNPs selected by UFS(MI) on skin color data

Data View	Selection
White Pale Beige Light Brown Medium Brown Dark Brown	rs1426654$^*(93.60\%), rs16891982^*(80.67\%), rs1448484^*(64.00\%)$, rs11230664$^*(63.87\%), rs28777^*(39.87\%), rs183671^*(32.20\%)$, rs1129038$^*(28.80\%), rs12913832^*(28.53\%), rs6497271^*(18.93\%)$, rs16950987$^*(16.53\%), rs10777129^*(13.33\%), rs2240203^*(11.00\%)$, rs7948623 (9.20%), rs8039195$^*(7.13\%), rs1375164^*(6.07\%)$, rs642742$^*(5.73\%), rs6119471^*(5.20\%)$
Light Intermediate Dark	rs1426654$^*(97.47\%), rs16891982^*(82.20\%), rs11230664^*(70.93\%)$, rs1448484$^*(58.53\%), rs28777^*(38.33\%), rs183671^*(30.27\%)$, rs1129038$^*(28.07\%), rs12913832^*(26.20\%), rs6497271^*(22.87\%)$, rs16950987$^*(15.40\%), rs7948623(13.53\%), rs1375164^*(11.40\%)$, rs2240203$^*(8.87\%), rs10777129^*(7.20\%), rs8039195^*(5.60\%)$, rs6119471$^*(5.53\%), rs895829^*(5.13\%)$
Light Dark	rs1426654$^*(96.00\%), rs11230664^*(84.53\%), rs1448484^*(74.27\%)$, rs16891982$^*(74.20\%), rs28777^*(45.40\%), rs183671^*(38.47\%)$, rs6497271$^*(37.47\%), rs10777129^*(30.93\%), rs7948623(10.07\%)$, rs16950987$^*(10.07\%), rs8039195^*(8.33\%), rs6119471^*(7.33\%)$, rs642742$^*(7.13\%), rs2240203^*(6.20\%), rs1375164^*(5.13\%)$

Table 5.11: SNPs selected by UFS(MI) on eye color data

Data View	Selection
Blue Green Hazel Light Brown Dark Brown Black	rs12913832*(95.00%), $rs1129038^*(95.00\%), rs7494942^*(50.47\%)$, rs3935591*(48.80%), $rs916977^*(45.80\%), rs1426654^*(37.87\%)$, rs7170852*(31.80%), $rs895829^*(25.20\%), rs7170989^*(24.53\%)$, rs16891982*(11.80%), $rs1448484^*(10.07\%), rs683^*(10.07\%)$, rs16950987*(9.20%), $rs4778138^*(9.13\%), rs11636232^*(9.13\%)$, rs1375164*(6.07%), $rs2240203^*(5.80\%)$
Blue Intermediate Dark	rs1129038*(95.13%), $rs12913832^*(94.87\%), rs3935591^*(55.80\%)$, rs7494942*(50.40%), $rs916977^*(45.33\%), rs1426654^*(38.27\%)$, rs7170852*(31.53%), $rs7170989^*(24.67\%), rs895829^*(23.13\%)$, rs683*(14.27%), $rs4778138^*(11.67\%), rs11636232^*(11.20\%)$, rs4778241*(8.53%), $rs16891982^*(6.27\%), rs16950987^*(5.47\%)$, rs1375164*(5.27%), $rs2240203^*(5.13\%)$
Blue Non-Blue	rs1129038*(95.00%), $rs12913832^*(95.00\%), rs3935591^*(61.47\%)$, rs7494942*(51.87%), $rs916977^*(44.53\%), rs7170852^*(35.20\%)$, rs7170989*(25.80%), $rs4778241^*(24.60\%), rs895829^*(23.60\%)$, rs683*(17.27%), $rs2238289^*(17.00\%), rs1426654^*(13.40\%)$, rs4778138*(8.53%), $rs16950987^*(6.40\%), rs1375164^*(5.20\%)$
Brown Non-Brown	rs1129038*(95.33%), $rs12913832^*(94.67\%), rs7494942^*(51.27\%)$, rs916977*(46.67%), $rs3935591^*(45.40\%), rs7170852^*(28.73\%)$, rs11636232*(26.87%), $rs895829^*(26.20\%), rs7170989^*(25.53\%)$, rs1426654*(25.13%), $rs4778138^*(21.47\%), rs683^*(11.53\%)$, rs16891982*(10.87%), $rs1375164^*(7.87\%), rs4778241^*(5.40\%)$, rs16950987*(5.13%)
Light Non-Light	rs1129038*(95.80%), $rs12913832^*(94.20\%), rs3935591^*(57.13\%)$, rs7494942*(52.00%), $rs916977^*(46.40\%), rs7170852^*(36.13\%)$, rs7170989*(25.67%), $rs895829^*(24.73\%), rs11636232^*(23.27\%)$, rs4778241*(20.13%), $rs683^*(17.07\%), rs4778138^*(16.33\%)$, rs1426654*(12.73%), $rs16950987^*(7.13\%)$

Table 5.12: SNPs selected by N3O-D on skin color data

Data View	Selection
White Pale Beige Light Brown Medium Brown Dark Brown	rs2835630*(23.33%), $rs2402130^*(20.00\%)$, $rs1724630^*(13.33\%)$, $rs2378249(13.33\%)$, rs1805005*(13.33%), $rs3768056^*(13.33\%)$, $rs10777129^*(10.00\%)$, $rs1042602^*(10.00\%)$, rs2733832*(10.00%), $rs6497271^*(10.00\%)$, $rs7170852^*(10.00\%)$, $rs1805009^*(10.00\%)$, rs1393350*(10.00%), $rs16891982^*(10.00\%)$, $rs12896399^*(10.00\%)$, $rs4778137^*(10.00\%)$, rs4778138*(10.00%), $rs183671^*(10.00\%)$, $rs1325127^*(10.00\%)$, $rs1800407^*(10.00\%)$, rs8039195*(10.00%), $rs11230664^*(10.00\%)$, $rs2424984^*(6.67\%)$, $rs1597196^*(6.67\%)$, rs1426654*(6.67%), $rs2594935^*(6.67\%)$, $rs6119471^*(6.67\%)$, $rs12913832^*(6.67\%)$, rs2070959*(6.67%), $rs10424065^*(6.67\%)$, $rs642742^*(6.67\%)$, $rs1800404^*(6.67\%)$, rs1805006*(6.67%), $rs1129038^*(6.67\%)$, $rs1900758^*(6.67\%)$, $rs1375164^*(6.67\%)$, rs9894429 (6.67%), rs10756819*(6.67%), $rs895828^*(6.67\%)$, $rs16950987^*(6.67\%)$, rs1110400*(6.67%), $rs7948623(6.67\%)$
Light Intermediate Dark	rs183671*(16.67%), $rs8039195^*(13.33\%)$, $rs885479^*(13.33\%)$, $rs1375164^*(13.33\%)$, rs12896399*(13.33%), $rs642742^*(13.33\%)$, $rs683^*(13.33\%)$, $rs2402130^*(13.33\%)$, rs7948623 (10.00%), rs3768056*(10.00%), $rs12913832^*(10.00\%)$, $rs3794606^*(10.00\%)$, rs3935591*(10.00%), $rs4932620^*(10.00\%)$, $rs11636232^*(10.00\%)$, $rs2835630^*(10.00\%)$, rs11230664*(10.00%), $rs4778241^*(10.00\%)$, $rs1448484^*(10.00\%)$, $rs6497271^*(10.00\%)$, rs13289*(10.00%), $rs9894429(6.67\%)$, $rs6119471^*(6.67\%)$, $rs2378249(6.67\%)$, rs1724630*(6.67%), $rs7170852^*(6.67\%)$, $rs1325127^*(6.67\%)$, $rs10424065^*(6.67\%)$, rs2240203*(6.67%), $rs1129038^*(6.67\%)$, $rs916977^*(6.67\%)$, $rs4959270^*(6.67\%)$, rs1805006*(6.67%), $rs1805009^*(6.67\%)$, $rs1426654^*(6.67\%)$, $rs1037208^*(6.67\%)$, rs4778137*(6.67%), $rs3212345^*(6.67\%)$
Light Dark	rs1042602*(23.33%), $rs11636232^*(20.00\%)$, $rs10777129^*(16.67\%)$, $rs7170989^*(16.67\%)$, rs1800404*(13.33%), $rs12913832^*(13.33\%)$, $rs16891982^*(13.33\%)$, $rs2424984^*(13.33\%)$, rs1129038*(13.33%), $rs2238289^*(13.33\%)$, $rs1800407^*(10.00\%)$, $rs2378249(10.00\%)$, rs1375164*(10.00%), $rs2733832^*(10.00\%)$, $rs4778241^*(10.00\%)$, $rs6497271^*(10.00\%)$, rs183671*(10.00%), $rs1110400^*(10.00\%)$, $rs683^*(10.00\%)$, $rs6119471^*(10.00\%)$, rs12896399*(6.67%), $rs1325127^*(6.67\%)$, $rs1900758^*(6.67\%)$, $rs9894429(6.67\%)$, rs1393350*(6.67%), $rs1448484^*(6.67\%)$, $rs3935591^*(6.67\%)$, $rs2402130^*(6.67\%)$, rs8039195*(6.67%), $rs3794606^*(6.67\%)$, $rs13289^*(6.67\%)$, $rs4778138^*(6.67\%)$, rs1805005*(6.67%), $rs2240203^*(6.67\%)$, $rs895829^*(6.67\%)$

Table 5.13: SNPs selected by N3O-D on eye color data

Data View	Selection
Blue Green Hazel Light Brown Dark Brown Black	$rs1448484^*(20.00\%), rs7170852^*(20.00\%), rs3794606^*(16.67\%), rs13289^*(16.67\%), rs2424984^*(16.67\%), rs4778137^*(16.67\%),$ rs4778232$^*(16.67\%), rs1900758^*(13.33\%), rs2036213^*(13.33\%), rs11230664^*(10.00\%), rs2594935^*(10.00\%), rs183671^*(10.00\%),$ rs13289810 (10.00%), rs2402130$^*(10.00\%), rs11636232^*(10.00\%), rs1800404^*(10.00\%), rs1805006^*(10.00\%), rs3212345^*(10.00\%),$ rs2733832$^*(10.00\%), rs1042602^*(10.00\%), rs10756819^*(10.00\%), rs1375164^*(10.00\%), rs1724630^*(10.00\%), rs642742^*(10.00\%),$ rs885479$^*(10.00\%), rs8039195^*(10.00\%), rs4778138^*(10.00\%), rs1800407^*(6.67\%), rs1393350^*(6.67\%), rs7494942^*(6.67\%),$ rs3768056$^*(6.67\%), rs7170989^*(6.67\%), rs2070959^*(6.67\%), rs4932620^*(6.67\%), rs895829^*(6.67\%), rs6119471^*(6.67\%),$ rs2378249 (6.67%), rs895828$^*(6.67\%), rs12896399^*(6.67\%), rs1426654^*(6.67\%), rs4959270^*(6.67\%), rs1805005^*(6.67\%),$ rs916977$^*(6.67\%), rs1805009^*(6.67\%), rs3935591^*(6.67\%), rs1037208^*(6.67\%), rs1129038^*(6.67\%)$
Blue Intermediate Dark	rs1129038$^*(16.67\%), rs12913832^*(13.33\%), rs4778138^*(13.33\%), rs7170852^*(10.00\%), rs1800407^*(10.00\%), rs3794606^*(10.00\%),$ rs1037208$^*(10.00\%), rs8039195^*(10.00\%), rs7948623(10.00\%), rs4778241^*(10.00\%), rs1448484^*(10.00\%), rs1597196^*(10.00\%),$ rs13289$^*(10.00\%), rs13289810(10.00\%), rs1393350^*(10.00\%), rs4932620^*(6.67\%), rs9894429(6.67\%), rs1800404^*(6.67\%),$ rs1805006$^*(6.67\%), rs1805005^*(6.67\%), rs885479^*(6.67\%), rs2070959^*(6.67\%), rs16950987^*(6.67\%), rs2733832^*(6.67\%),$ rs12203592$^*(6.67\%), rs4778137^*(6.67\%), rs10756819^*(6.67\%), rs11636232^*(6.67\%), rs3212345^*(6.67\%), rs10777129^*(6.67\%)$
Blue Non-Blue	rs7494942$^*(23.33\%), rs3794606^*(16.67\%), rs1129038^*(16.67\%), rs3935591^*(13.33\%), rs11230664^*(13.33\%), rs8039195^*(13.33\%),$ rs13289$^*(13.33\%), rs10424065^*(13.33\%), rs2424984^*(13.33\%), rs16891982^*(13.33\%), rs1037208^*(10.00\%), rs2733832^*(10.00\%),$ rs1805006$^*(10.00\%), rs4959270^*(10.00\%), rs6497271^*(10.00\%), rs1325127^*(10.00\%), rs4778232^*(10.00\%), rs1800407^*(6.67\%),$ rs11636232$^*(6.67\%), rs3212345^*(6.67\%), rs6119471^*(6.67\%), rs1800404^*(6.67\%), rs9894429(6.67\%), rs2378249(6.67\%),$ rs1375164$^*(6.67\%), rs13289810(6.67\%), rs1597196^*(6.67\%), rs2835630^*(6.67\%), rs4778137^*(6.67\%), rs1393350^*(6.67\%),$ rs1900758$^*(6.67\%), rs4932620^*(6.67\%), rs16950987^*(6.67\%), rs7170852^*(6.67\%), rs7948623(6.67\%), rs1110400^*(6.67\%),$ rs2238289$^*(6.67\%)$
Brown Non-Brown	rs7494942$^*(23.33\%), rs1129038^*(20.00\%), rs28777^*(16.67\%), rs3768056^*(13.33\%), rs2070959^*(10.00\%), rs11230664^*(10.00\%),$ rs895829$^*(10.00\%), rs4778137^*(10.00\%), rs1448484^*(10.00\%), rs2402130^*(10.00\%), rs13289^*(10.00\%), rs1800404^*(10.00\%),$ rs2378249 (6.67%), rs4778241$^*(6.67\%), rs916977^*(6.67\%), rs2835630^*(6.67\%), rs2240203^*(6.67\%), rs4778138^*(6.67\%),$ rs1805006$^*(6.67\%), rs1805009^*(6.67\%), rs1037208^*(6.67\%), rs6497271^*(6.67\%), rs2424984^*(6.67\%), rs6119471^*(6.67\%),$ rs1597196$^*(6.67\%), rs1375164^*(6.67\%), rs885479^*(6.67\%), rs3794606^*(6.67\%), rs7170852^*(6.67\%)$
Light Non-Light	rs7494942$^*(16.67\%), rs12913832^*(16.67\%), rs1129038^*(16.67\%), rs6497271^*(13.33\%), rs13289^*(13.33\%), rs16950987^*(13.33\%),$ rs1900758$^*(10.00\%), rs2733832^*(10.00\%), rs885479^*(10.00\%), rs1724630^*(10.00\%), rs7170852^*(10.00\%), rs11636232^*(10.00\%),$ rs2424984$^*(10.00\%), rs28777^*(6.67\%), rs12896399^*(6.67\%), rs4959270^*(6.67\%), rs9894429(6.67\%), rs3935591^*(6.67\%),$ rs1426654$^*(6.67\%), rs2238289^*(6.67\%), rs2036213^*(6.67\%), rs4778138^*(6.67\%), rs10777129^*(6.67\%), rs6510760(6.67\%),$ rs3794606$^*(6.67\%), rs3212345^*(6.67\%), rs10424065^*(6.67\%), rs2835630^*(6.67\%)$

Tables 5.14 and 5.15 present their respective measured qualities, in which bold entries indicate which algorithm retrieved the best feature selection and italic entries mark data views in which FS-NEAT outperformed N3O-D.

Table 5.14: Classification performance of selected features on skin color data views

Data View	Baseline	UFS(MI)	N3O-D	FS-NEAT
White, Pale, Beige, Light Brown, Medium Brown, Dark Brown	31.73	**44.96 ± 1.53**	38.58 ± 0.83	*42.46 ± 0.7*
Light, Intermediate, Dark	57.52	**79.07 ± 2.68**	73.29 ± 1.36	69.63 ± 1.13
Light, Dark	71.9	**93.98 ± 1.66**	89.49 ± 0.87	89.26 ± 0.32

Table 5.15: Classification performance of selected features on eye color data views

Data View	Baseline	UFS(MI)	N3O-D	FS-NEAT
Blue, Green, Hazel, Light Brown, Dark Brown, Black	23.91	**61.92 ± 2.84**	56.25 ± 0.51	*56.95 ± 0.83*
Blue, Intermediate, Dark	51.92	**81.20 ± 2.64**	77.93 ± 3.47	*78.14 ± 1.8*
Blue, Non-Blue	76.09	93.49 ± 1.68	**93.86 ± 1.41**	78.81 ± 0.92
Brown, Non-Brown	61.29	**89.19 ± 1.56**	89.06 ± 1.46	83.11 ± 3.34
Light, Non-Light	67.89	94.76 ± 0.8	**95.21 ± 0.94**	84.01 ± 3.61

Tables 5.16 and 5.17 present the confusion matrixes yielded by the evaluation of UFS(MI)'s feature selections, respectively for the original data views of skin and eye color. Tables 5.18 and 5.19 present the same information for feature selections retrieved by the proposed method. Entries represent the percentage of classifications concerning the number of evaluated predictions, rather than the absolute number classifications. Only predictions filtered to measure the classification performance of each algorithm were accounted.

Table 5.16: Confusion matrix for UFS(MI) skin color predictions

	White	**Pale**	**Beige**	**Light Brown**	**Medium Brown**	**Dark Brown**
White	9.55	14.68	0.62	0.56	0.36	0
Pale	10.37	18.68	1.72	0.74	0.2	0
Beige	1.21	9.02	1.95	1.31	0.85	0.01
Light Brown	0.05	2.62	0.9	2.79	3.01	0.19
Medium Brown	0.21	0.73	0.26	2.68	7.29	1.14
Dark Brown	0	0.1	0.03	0.36	3.89	1.85

Table 5.17: Confusion matrix for UFS(MI) eye color predictions

	Blue	**Green**	**Hazel**	**Light Brown**	**Dark Brown**	**Black**
Blue	22.49	0.08	0.05	0.15	1.13	0
Green	4.78	0.03	0.08	0.12	3.16	0
Hazel	0.46	0.09	0.1	0.3	5.6	0
Light Brown	0.07	0	0.8	0.34	8.75	0.06
Dark Brown	0.01	0.09	0.2	0.52	33.51	2.76
Black	0	0.01	0	0.05	11.22	3.51

Table 5.18: Confusion matrix for N3O-D skin color predictions

	White	**Pale**	**Beige**	**Light Brown**	**Medium Brown**	**Dark Brown**
White	0.22	25.57	0	0	0	0
Pale	0	31.05	0	0	0.68	0
Beige	0	13.92	0	0	0.45	0
Light Brown	0.68	6.16	0	0	2.73	0
Medium Brown	0.68	4.33	0	0	7.3	0
Dark Brown	0.22	0.91	0	0	5.02	0

Table 5.19: Confusion matrix for N3O-D(MI) eye color predictions

	Blue	**Green**	**Hazel**	**Light Brown**	**Dark Brown**	**Black**
Blue	22.09	0	0	0	1.82	0
Green	5.23	0	0	0	2.96	0
Hazel	1.82	0	0	0	4.78	0
Light Brown	3.18	0	0	0	6.15	0
Dark Brown	5.46	0	0	0	28.7	2.96
Black	0.22	0	0	0	9.11	5.46

Furthermore, Tables 5.20 and 5.21 describe the capability of N3O-D and FS-NEAT to solve the given classification task on its own, while Tables 5.22 and 5.23 depict the average input size of the best topology found in each neuroevolution run.

Table 5.20: Classification performance of champion topologies on skin color data

Data View	Baseline	N3O-D	FS-NEAT
White, Pale, Beige, Light Brown, Medium Brown, Dark Brown	**31.73**	18.08 ± 9.31	18.24 ± 8.7
Light, Intermediate, Dark	**57.52**	32.53 ± 16.91	14.02 ± 3.53
Light, Dark	**71.9**	46.69 ± 22.6	22.06 ± 3.86

Table 5.21: Classification performance of champion topologies on eye color data

Data View	Baseline	N3O-D	FS-NEAT
Blue, Green, Hazel, Light Brown, Dark Brown, Black	**23.91**	17.85 ± 10.6	18.12 ± 9.8
Blue, Intermediate, Dark	**51.92**	39.43 ± 13.91	32.79 ± 11.58
Blue, Non-Blue	**76.09**	52.29 ± 23.78	56.77 ± 23.03
Brown, Non-Brown	**61.29**	55.02 ± 18.06	51.73 ± 12.3
Light, Non-Light	**67.89**	48.06 ± 15.85	55.33 ± 19.56

Table 5.22: Input size of evolved topologies on skin color data

Data View	N3O-D	FS-NEAT
White, Pale, Beige, Light Brown, Medium Brown, Dark Brown	4.4 ± 2.33	**4.26 ± 2.51**
Light, Intermediate, Dark	4.23 ± 2.1	**3.53 ± 1.74**
Light, Dark	4.23 ± 2.41	**3.86 ± 2.56**

Table 5.23: Input size of evolved topologies on eye color data views

Data View	N3O-D	FS-NEAT
Blue, Green, Hazel, Light Brown, Dark Brown, Black	**5.16 ± 2.81**	5.16 ± 3.11
Blue, Intermediate, Dark	**3.53 ± 2.3**	3.93 ± 2.29
Blue, Non-Blue	4.03 ± 2.16	**3.6 ± 2.53**
Brown, Non-Brown	3.5 ± 1.97	**2.96 ± 1.74**
Light, Non-Light	**3.53 ± 2.29**	3.83 ± 2.46

To perform a functional analysis of retrieved selections, selected SNPs were queried on the public databases SNPedia [3], the United States National Library of Medicine [4] (USNBI) and GeneCards [5]. SNPs that were somehow associated with skin or color phenotypes (either because they reside nearby genes that are known for influencing the phenotypes or were previously related to skin or eye color pigmentation) are listed in Table

[3] <https://snpedia.com/>
[4] <https://www.ncbi.nlm.nih.gov/>
[5] <https://www.genecards.org/>

5.24. SNPs whose association with the analyzed phenotypes is unknown are reported in Table 5.25. Reports present the genetic location of each selected SNP if it is located nearby a known gene.

As Tables 5.20 and 5.21 showed, N3O-D and FS-NEAT evolved individuals did not present classification performance superior to the baseline, as well as a high variance on cross-validation results, which raises the suspicion that the optimization implicitly employed by evolution has not converged or is not properly configured. The comparison in Tables 5.14 and 5.15 demonstrated that the proposed method's selections, in general, did not improve on UFS(MI), except for the two least balanced eye data views, Blue vs Non-Blue and Light vs Non-Light. The proposed method retrieved inputs that consistently outperformed those of its base algorithm: except for the original data view for skin color, N3O-D only failed to improve on FS-NEAT by an accuracy difference within the measured standard deviation. In tables 5.16, 5.17, 5.18 and 5.19 it is observable that induced classifiers often err predictions by outputting labels of over-represented classes (*Blue* and *Dark Brown* for eye, and *Pale* for skin data). Such a trend is stronger on N3O-D's predictions.

As in Whiteson et al. (2005), FS-NEAT and N3O-D evolved topologies with an increasing number of selected features as evolution progressed, but at a much slower and inconsistent pace. In comparison to Grisci, Feltes and Dorn (2018), NE algorithms evaluated in this study evolved topologies with less connected inputs. It is important to highlight that N3O was proposed and evaluated on gene expression data that was subject to an intensive curation process (FELTES et al., 2019), while data applied to the experiments on this study were treated with imputation only, as described in Section 5.1.1. Furthermore, gene expression data is represented by real-valued numbers which can directly serve as input to neural networks, in contrast to SNPs data which, in this study, was transformed via one-hot encoding

5.6 Chapter conclusion

This chapter presented the experimental evaluation employed to assess N3O-D's feature selection capability and its ability to evolve networks that solve the phenotype prediction problem. The proposed method was compared to FS-NEAT, its base algorithm, as well as UFS(MI), a feature selection framework in the state of art. Although topologies found through neuroevolution could not surpass the baseline accuracy, the pro-

Table 5.24: **Report of relevant selected SNPs.** *na* entries correspond to SNPs whose location was not related to a known gene. All listed SNPs were previously associated to eye or skin color phenotypes.

SNP	Gene
rs1037208	OCA2
rs10424065	MFSD12
rs1042602	TYR
rs10756819	BNC2
rs10777129	KIKTLG
rs1110400	MC1R
rs11230664	ASIP
rs1129038	HERC2
rs11636232	HERC2
rs12203592	IRF4
rs12896399	*na*
rs12913832	HERC2
rs1325127	OCA2
rs13289	SLC45A2
rs1375164	OCA2
rs1393350	TYR
rs1426654	SLC24A5
rs1448484	OCA2
rs1597196	OCA2
rs16891982	SLC45A2
rs16950987	HERC2
rs1724630	MYO5A
rs1800404	OCA2
rs1800407	HERC2
rs1805005	MC1R
rs1805006	MC1R
rs1805009	MC1R
rs183671	SLC45A2
rs1900758	OCA2
rs2036213	OCA2
rs2070959	OCA2
rs2238289	HERC2
rs2240203	HERC2
rs2402130	SLC24A4
rs2424984	OCA2
rs2594935	ASIP
rs2733832	TYRP1
rs2835630	TTC3
rs28777	SLC45A2
rs3212345	*na*
rs3768056	LYST
rs3794606	OCA2
rs3935591	HERC2
rs4778137	OCA2
rs4778138	OCA2
rs4778232	OCA2
rs4778241	OCA2
rs4932620	HERC2
rs4959270	*na*
rs6119471	UGT1A6-10
rs642742	*na*
rs6497271	HERC2
rs683	TYRP1
rs7170852	HERC2
rs7170989	OCA2
rs7494942	HERC2
rs8039195	DDB1
rs885479	MC1R
rs895828	OCA2
rs895829	HERC2

Table 5.25: **Report of novel selected SNPs.** *na* entries correspond to SNPs whose location was not related to a known gene. All listed SNPs are not associated to eye or skin color phenotypes, given the employed functional enrichment.

SNP	Gene
rs13289810	*na*
rs2378249	PIGU
rs6510760	*na*
rs7948623	TMEM138
rs9894429	NPLOC4

posed method's retrieved selection yielded the best accuracy performance on some of the employed data views. Also, N3O-D presented a consistent improvement on FS-NEAT, matching previous results in which mutual information was pointed out as a useful guide to feature selection. The following chapter concludes the study, relating obtained results with the work's objectives and pointing directions for future work.

6 CONCLUSION

A novel feature selection method based on neuroevolution and adapted to categorical data was proposed and evaluated on the genotype-phenotype prediction problem, posed by a data set with SNPs biomarkers labeled with eye and skin color. The proposed method, named N3O-D, utilized mutual information to adapt a recent neuroevolution feature selection algorithm that was successfully applied to microarray data sets. N3O-D inference and selective capability were compared to a state-of-art framework that also used mutual information to guide feature selection. Given current experimental settings, the proposed method presented poor classification performance on its own, being unable to surpass baseline accuracy. Nevertheless, some of its selected attributes improved SVM's accuracy in comparison to univariate feature selection, in particular on imbalanced training sets labeled with eye color. The poor classification performance presented warrants further investigation on N3O-D's parameters configuration, especially on the data sets in which NEAT, FS-NEAT and N3O showed promising results.

Evaluated algorithms based on neuroevolution were developed from scratch on the Python [1] programming language, employing TensorFlow [2], scikit-learn [3] and NetworkX [4] as main dependencies. This venture was motivated by the lack of generalization provided by existing neuroevolution libraries. For instance, NEAT-Python [5] does not provide an implementation for FS-NEAT. Although the library is well documented and provides a configuration file interface, obtaining such an implementation requires the modification (rather than extension) of the source code, which is not a trivial task. The code developed in this study is to be improved and generalized to provide free, well-written and extensible computational tools for future neuroevolution research, through the release of an open-source library focused on neuroevolutive algorithms.

Evolutionary algorithms internally produce statistics concerning evolution runs, which provide useful insight on how the algorithm can be improved by tuning its parameters, besides better detailing the evolution process (GRISCI; FELTES; DORN, 2018). Examples are the progress of population's mean and incumbents' fitness; species rise, progress and extinction; graphical representations of evolved networks; frequency in which inputs are selected throughout evolution. In this study, such statistics were not

[1] <https://python.org>
[2] <https://tensorflow.org>
[3] <https://scikit-learn.org/>
[4] <https://networkx.github.io/>
[5] <https://neat-python.readthedocs.io/>

collected but are considered fundamental for the interpretation and improvement of neuroevolution algorithms. Another line of future work is the employment of quantification on continuous data as a noise reduction strategy (DING; PENG, 2003). Such a transformation turns a numerical data set into a discrete one, which motivates the application of N3O-D as a feature selector.

REFERENCES

ACADEMIA.EDU. **Image depicting a single nucleotide polymorphism**. 2015. Available from Internet: <https://knowgenetics.org/snps/>.

AGGARWAL, C. et al. Feature selection for classification: A review. **Data Classification: Algorithms and Applications**, 2014.

ALPAYDIN, E.; JORDAN, M. Local linear perceptrons for classification. **IEEE Transactions on Neural Networks**, 1996.

ANG, J. et al. Supervised, unsupervised, and semi-supervised feature selection: A review on gene selection. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, 2016.

ARENAS, M. et al. Forensic genetics and genomics: Much more than just a human affair. **PLOS Genetics**, 2017.

AVILA, E. et al. Forensic characterization of brazilian regional populations through massive parallel sequencing of 124 snps included in hid ion ampliseq identity panel. **Forensic Science International: Genetics**, 2019.

BATISTA, G.; MONARD, M. An analysis of four missing data treatment methods for supervised learning. **Applied Artificial Intelligence**, 2003.

BELKIN, M. et al. Reconciling modern machine learning practice and the classical bias-variance trade-off. **National Academy of Sciences of the United States of America (Proceedings)**, 2019.

BEYER, H.; SCHWEFEL, H. Evolution strategies - a comprehensive introduction. **Natural Computing**, 2002.

BOER, P. D. et al. A tutorial on the cross-entropy method. **Annals of Operations Research**, 2005.

BUNNEY, P. et al. Feature selection by optimizing a lower bound of conditional mutual information. **Physiology and Behavior**, 2017.

CALUS, M.; VEERKAMP, R. Accuracy of breeding values when using and ignoring the polygenic effect in genomic breeding value estimation with a marker density of one snp per cm. **Journal of Animal Breeding and Genetics**, 2007.

CEDRIC, S. An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing. **School of Electrical Engineering and Computer Science (EECS)**, 2018.

CILIA, N. et al. An experimental comparison of feature-selection and classification methods for microarray datasets. **Information (Switzerland)**, 2019.

CORTE-REAL, F.; VIEIRA, D. **Princípios de Genética Forense**. 1. ed. [S.l.: s.n.], 2015. ISBN 978-8576253600.

CORTES, C.; VAPNIK, V. Support-vector networks. **IEEE Expert-Intelligent Systems and their Applications**, 1992.

COVER, T.; THOMAS, J. **Elements of Information Theory**. 2. ed. [S.l.: s.n.], 2006.

DASH, M.; LIU, H. Feature selection for classification, in intelligent data analysis. **Elsevier Intelligent Data Analysis**, 1997.

DING, C.; PENG, H. Minimum redundancy feature selection from microarray gene expression. **Computation Systems Bioinformatics**, 2003.

DRUZHKOV, P.; KUSTIKOVA, V. A survey of deep learning methods and software tools for image classification and object detection. **Pattern Recognition and Image Analysis**, 2016.

DUFOURQ, E.; BASSET, B. Eden: Evolutionary deep networks for efficient machine learning. **Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)**, 2017.

FELTES, B. et al. Cumida: An extensively curated microarray database for benchmarking and testing of machine learning approaches in cancer research. **Journal of Computational Biology**, 2019.

FLACH, P. The geometry of roc space: Understanding machine learning metrics through roc isometrics. **Twentieth International Conference on Machine Learning Proceedings**, 2003.

FLOREANO, D.; DüRR, P.; MATTIUSSI, C. Neuroevolution: From architectures to learning. **Evolutionary Intelligence**, 2008.

GRAY, R. M. **Entropy and Information Theory**. 1. ed. [S.l.]: Springer, 1990. ISBN 978-1441979704.

GRISCI, B.; FELTES, B.; DORN, M. Neuroevolution as a tool for microarray gene expression pattern identification in cancer research. **Journal of Biomedical Informatics (JBI)**, 2018.

GUYON, I. et al. Gene selection for cancer classification using support vector machines. **Machine Learning**, 2002.

HAN, E.; KARYPIS, G.; KUMAR, V. Text categorization using weight adjusted k-nearest neighbor classificatio. **Lecture Notes in Computer Science (LNCS)**, 2001.

HORNIK, K. Approximation capabilities of multilayer feedforward networks. **Neural Networks**, 1991.

JOBLING, M.; GILL, P. Encoded evidence: Dna in forensic analysis. **Nature Review Genetics**, 2004.

KRUSKAL, W. H.; WALLIS, W. A. Use of ranks in one-criterion variance analysis. **Journal of the American Statistical Association**, 1952.

LAWRENCE, S.; GILES, C. Overfitting and neural networks: Conjugate gradient and backpropagation. **International Joint Conference on Neural Networks (Proceedings)**, 2000.

LAZAR, C. et al. A survey on filter techniques for feature selection in gene expression microarray analysis. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, 2012.

LIU, F. et al. Eye color and the prediction of complex phenotypes from genotypes. **Current Biology**, 2009.

LOPEZ-RINCON, A. et al. Evolutionary optimization of convolutional neural networks for cancer miRNA biomarkers classification. **Applied Soft Computing Journal**, 2018.

LU, H.; CHEN, J.; YAN, K. A hybrid feature selection algorithm for gene expression data classification. **Neurocomputing**, 2017.

LUKE, S. **Essentials of Metaheuristics**. 2. ed. [S.l.: s.n.], 2013. ISBN 978-1300549628.

MATUKUMALLI, L. et al. Application of machine learning in snp discovery. **BMC Bioinformatics**, 2006.

MIAO, J.; NIU, L. A survey on feature selection. **Procedia Computer Science**, 2016.

MORSE, G.; STANLEY, K. Simple evolutionary optimization can rival stochastic gradient descent in neural networks. **Genetic and Evolutionary Computation Conference (GECCO)**, 2016.

MULDER, W. D.; BETHARD, S.; MOENS, M. A survey on the application of recurrent neural networks to statistical language modeling. **Computer Speech and Language**, 2015.

NOI, P.; KAPPAS, M. Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using sentinel-2 imagery. **Sensors**, 2017.

OGDEN, R.; LINACRE, A. Wildlife forensics science: A review of genetic geographic origin assignment. **Forensic Science International: Genetics**, 2015.

PAPAVASILEIOU, E.; JANSEN, B. The importance of the activation function in neuroevolution with fs-neat and fd-neat. **IEEE Symposium Series on Computational Intelligence Proceedings (SSCI)**, 2018.

POTDAR, K.; TAHER, S.; CHINMAY, D. A comparative study of categorical variable encoding techniques for neural network classifiers. **International Journal of Computer Applications**, 2017.

QU, Y.; WANG, M. Approximation capabilities of neural networks on unbounded domains. **arXiv:1910.09293**, 2019.

RUCK, D. W. et al. The multilayer perceptron as an approximation to a bayes optimal discriminant function. **IEEE Transactions on Neural Networks**, 1990.

RUMELHART, D. E.; HINTON, G. E.; WILLIAM, R. J. **Learning internal representations by error propagation**. [S.l.]: Parallel Distributed Processing, 1986. ISBN 978-0262291408.

SHANNON, C. E. A mathematical theory of communication. **The Bell System Technical Journal**, 1948.

SOHANGIR, S.; RAHIMI, S.; GUPTA, B. Optimized feature selection using neuroevolution of augmenting topologies (neat). **Proceedings of the 2013 Joint IFSA World Congress and NAFIPS Annual Meeting**, 2013.

SOHANGIR, S.; RAHIMI, S.; GUPTA, B. Neuroevolutionary feature selection using neat. **Journal of Software Engineering and Applications**, 2014.

SOKOLOVA, M.; JAPKOWICZ, N.; SZPAKOWICZ, S. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. **AI 2006: Advances in Artificial Intelligence**, 2006.

STANLEY, K.; CLUNE, J.; LEHMAN, J. Designing neural networks through neuroevolution. **Nature Machine Intelligence**, 2019.

STANLEY, K.; MIIKKULAINEN, R. Evolving neural networks through augmenting topologies. **Evolutionary Computation**, 2002.

STEUER, R. et al. The mutual information: Detecting and evaluating dependencies between variables. **Bioinformatics**, 2002.

TAN, M. et al. Automated feature selection in neuroevolution. **Evolutionary Intelligence**, 2009.

TIELEMAN, T.; HINTON, G. Divide the gradient by a running average of its recent magnitude. **COURSERA: Neural Networks for Machine Learning**, 2012.

UTANS, J.; MODDY, J.; REHFUSS, S. Input variable selection for neural networks: application to predicting the u.s. business cycle. **IEEE/IAFE Conference on Computational Intelligence for Financial Engineering, Proceedings (CIFEr)**, 1995.

VERGARA, J.; ESTÉVEZ, P. A review of feature selection methods based on mutual information. **Neural Computing and Applications**, 2015.

WHITE, D.; RABAGO-SMITH, M. Genotype-phenotype associations and human eye color. **Journal of Human Genetics**, 2011.

WHITESON, S. et al. Automatic feature selection via neuroevolution. **Proceedings of the Genetic and Evolutionary Computation Conference**, 2005.

WILLIAMS, R.; WIENROTH, M. Social and ethical aspects of forensis genetics. **Forensic Science Review**, 2017.

XIE, L.; YUILLE, A. Genetic cnn. **IEEE International Conference on Computer Vision (ICCV)**, 2017.

YOUNG, A.; BENONISDOTTIR, S.; PRZEWORSKI, M. Deconstructing the sources of genotype-phenotype associations in humans. **Science**, 2019.

ZAORSKA, K.; ZAWIERUCHA, P.; NOWICKI, M. Prediction of skin color, tanning and freckling from dna in polish population: linear regression, random forest and neural network approaches. **Human Genetics**, 2019.

ZHANG, G. Neural networks for classification: A survey. **IEEE Transactions on Systems, Man and Cybernetics**, 2000.