UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO

Luiz Daniel Ramos França

Neuroevolution of Augmenting Topologies Applied to the Detection of Cancer in Medical Images

Recife, Brazil

February - 2018

Luiz Daniel Ramos França

Neuroevolution of Augmenting Topologies Applied to the Detection of Cancer in Medical Images

Trabalho de Conclusão de Curso apresentado ao Programa de Bacharelado em Ciência da Computação do Departamento de Estatística e Informática da Universidade Federal Rural de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Programa de Graduação

Supervisor: Péricles B. C. Miranda

Co-supervisor: Filipe Rolim Cordeiro

Recife, Brazil

February - 2018

MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO (UFRPE)
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

http://www.bcc.ufrpe.br

FICHA DE APROVAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO

Trabalho defendido por Luiz Daniel Ramos França às 14 horas do dia 08 de fevereiro de 2018, na Sala 05 do CEAGRI-02, como requisito para conclusão do curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, intitulado **Neuroevolution of Augmenting Topologies Applied to the Detection of Cancer in Medical Images**, orientado por Péricles Barbosa Cunha de Miranda e aprovado pela seguinte banca examinadora:

Péricles Barbosa Cunha de Miranda
DEINFO/UFRPE

Valmir Macário Filho
DEINFO/UFRPE

Ricardo Bastos Cavalcante Prudêncio
CIn/UFPE

Acknowledgements

II would like to thank everyone who has been a part of my journey through my graduation. In special, I would like to express my deepest gratitude my parents, Claudia Ramos França and Paulo Roberto França for all the support they have given me. I thank them for giving me everything so I can follow my dreams and for believing in me even when I did not. Moreover, I would like to thank my brother Paulo Roberto França Filho, my sister Paula Roberta de Araújo França, and all my family for their support.

I am so very grateful for my advisor, Péricles Miranda, for all the guidance and help he has given me. I would also like to express my gratitude to my co-advisor Filipe Rolim, for his lessons and inspirational classes.

I'm thankful for the Federal Rural University of Pernambuco for all the opportunities given to me during my time there. I am grateful for its faculty and staff, in special to Sandra Xavier for all the help and attention she gives to all the students.

I would like to express my gratitude to Larissa Lages, Giovanni Paolo, and Thays Silva, for all the help, support, and mentorship they gave me to make this work possible. To my friends Suzana Saraiva, Pedro Cavalcanti, Charles de Oliveira, and all the members of Bubble and B-hash, for all the times of joy and happiness, we experienced in the last few years.

I'd like to thank Perseu Bastos, Lucas Alencar, Bruno Tabosa, Carolina Alves, and all the people from the Icebreak, who gave me the opportunity to have my first job. Where we had so many great times together.

I'm very thankful for the time I spent at the University of Wisconsin Eau Claire, during my exchange program, where I had the opportunity experience many of my most memorable times. Where I had the opportunity to meet and travel with great friends such as Mahira Araujo and Caroline Lima.

Finally, I would like to thank each and everyone who has helped me be who I am today, and who has been a part of my journey through life.

" Sometimes it is the people no one can imagine anything of who do the things no one can imagine."

(Alan Turing)

Resumo

Câncer de mama é um dos tipos de câncer que mais mata mulheres no mundo. O câncer de mama é uma doença causada pelo crescimento descontrolado das células da mama que formam tumores e danificam os tecidos ao redor. O diagnostico prematuro da doença é um dos fatores mais importantes no sucesso do tratamento.

A mamografia é exame de imagem de radiografia da mama, bastante utilizada para detectar câncer de mama em mulheres que ainda não apresentam sinais ou sintomas da doença. A análise da mamografia geralmente é realizada pelo radiologista, que examina a mamografia em busca de anormalidades na mama. Dependendo da experiência do radiologista, qualidade da imagem e idade da mulher, existe a possibilidade do exame gerar falso-positivos, o que pode gerar ansiedade e estresse para o paciente, além da necessidade de outros exames que podem ser mais invasivos, ou falso-negativos, que deixaria a paciente sem diagnóstico por mais tempo, permitindo a evolução da doença para estágios mais graves.

O uso de algoritmos de aprendizagem de maquina podem auxiliar médicos no diagnostico mais preciso de pacientes. Tais programas são conhecidos por Diagnósticos Auxiliados por Computadores, do inglês, *Computer-Aided Diagnosis* (CAD). Para realizar tal tarefa, são extraídas cateterísticas das imagens de mamografia e esses exemplos são usados como entrada para os algoritmos de aprendizagem de máquina. Atualmente existem vários algoritmos de classificação que podem ser aplicados para este problema. Dentre estes existem algoritmos de neuroevolução. Estes algoritmos fazem uso de algoritmos genéticos para otimizar os hiper-parâmetros dos classificadores.

Este trabalho apresenta um estudo do algoritmo neuroevolutivo chamado Neuroevolução com Aumento Topológico, do inglês *Neuroevolution of Augmented Topologies* (NEAT), aplicado a detecção de câncer em imagens de mamografia. Para isso foram usados quatro *datasets* e os resultados foram comparados com três classificadores diferentes e com seis abordagens neuroevolutivas encontrados na literatura.

Os resultados motram que o NEAT apresenta uma performance semelhante ao de trabalhos encontrados na literatura, mesmo usando um número menor de gerações. Além disso, o NEAT alcançou resultados satisfatórios na análise binária da base de imagens IRMA, com *f-score* de 92,15%. Porém, o NEAT teve baixo *f-score* na análise multi-classe da base IRMA.

Palavras-chave: neuroevolução. aprendizagem de máquina. algoritmos genéticos.

Abstract

Breast cancer is one of the diseases that mostly affects women. Breast cancer is a disease caused when the breast cells grow out of control forming tumors that damage surrounding tissues. Diagnosis of the disease in the early stages increases the chances of success in treatments.

A mammogram is a breast imaging technique that uses low-dose x-rays, and it is widely used in the detection of breast cancer in women who have not shown symptoms of the disease. The mammogram is often analyzed by a radiologist that looks for abnormalities in the breast. Depending on the experience of the radiologist, the quality of the image and the age of the patient, it is possible to misdiagnose the disease, which could leave the disease untreated and allow it to evolve into more dangerous stages.

The use of machine learning algorithms can aid physicians to make more accurate diagnoses. Those group of programs is known as Computer-Aided Diagnosis (CAD). For those systems perform this task, it is necessary to extract features from the mammography images and feed them to the machine learning algorithm. Nowadays, there are many classification algorithms that could be used to solve this problem. Among those, there is a class of algorithms of neuroevolution. These methods use genetic algorithms to optimize the hyper-parameters of the classifiers.

This work presents a study of the NeuroEvolution of Augmenting Topologies applied in the context of detecting tumors in medical images. To assess the algorithms, it was used four datasets, and the results are compared to three different classifiers and six neuroevolution approaches found in the literature.

The results show that NEAT presents a performance similar to those algorithms found in the literature, even when executing a much smaller number of generations. The NEAT algorithm reached satisfactory results on the binary analyses of the IRMA dataset, with an f-score of 92.15%. Although NEAT obtained a low f-score on the multi-class analyses of IRMA.

Keywords: neuroevolution. machine learning. genetic algorithms.

List of Figures

List of Tables

List of abbreviations and acronyms

ANN	Artificial Neural Network
BI-RADS	Breast Imaging Reporting and Data System
BP	Backpropagation
CAD	Computer-Aided Diagnosis
CGP	Cartesian Genetic Programming
DDSM	Digital Database for Screening Mammography
EC	Evolutionary Computing
FNA	Fine Needle Aspirate
IRMA	Image Retrieval in Medical Applications
ISIC	International Skin Imaging Collaboration
GA	Genetic Algorithm
GLCM	Gray-level Co-occurrence Matrix
GP	Genetic Programming
GONN	Genetically Optimized Neural Network
GS	Grid-Search
LBP	Local Binary Pattern
LLNL	Lawrence Livermore National Laboratory
LTEM	Laws Texture Energy
LRBC	Ljubjana Recurrence Breast Cancer
PSO	Particle Swarm Optimization
PSOWNN	Particle swarm Optimized Wavelet Neural Network
MIAS	Mammographic Image Analysis Society Digital Mammogram Database
MLP	Multilayer Perceptron

MODE	Multi-objective Differential Evolution
MSE	Mean Square Error
NE	Neuroevolution
NEAT	NeuroEvolution of Augmenting Topologies
NN	Neural Network
RF	Random Forest
RBF	Radial Basis Function
ROI	Region of Interest
RS	Randomized Search
RWTH	Rheinisch-Westfälische Technische Hochschule
SVM	Support Vector Machine
TWEANNs	Topology and Weight Evolving Artificial Neural Networks
WBCD	Wisconsin Breast Cancer Dataset
WDBC	Wisconsin Diagnosis Breast Cancer
WPBC	Wisconsin Prognostics Breast Cancer
WNN	Wavelet Neural Network

Contents

1 Introduction

Breast cancer is a disease that begins when the cells of the breast start reproducing out of control. According to the American Cancer Society[1], breast cancer can start in the ducts that carry the milk to the nipple, called Invasive Ductal Carcinoma, or in the glands that make the milk, called Invasive Lobular Carcinoma. On the later stages, the disease can spread through the body in a process called metastases, which in later cases can lead to death. According to the Center for Disease Control and Prevention[2], besides some kinds of skin cancer, breast cancer is the most common cause of death by cancer in women in the United States. Moreover, it is estimated that 266,120 new cases of invasive breast cancer to be diagnosed in women in the U.S., according to the Breast Cancer Organization[3].

It is crucial to diagnose breast cancer in the early stages to allow a greater probability of survival of the patient. According to a study made by the Office for National Statistics[4] from 2014 to 2015, 95% of the women diagnosed with breast cancer on the third stage survived the first year after the diagnoses against 63% of the women diagnosed on the fourth stage. Besides, according to the American Cancer Society, 100% of the women diagnosed with stage 1 survived at least five years after the diagnoses, whereas only 22% women diagnosed with stage 4 survived the same period.

Medical images are used to help specialists to detect breast cancer in women that don't have any symptoms of the disease. Depending on the experience of the specialist, the quality of the image and the age of the woman, it is possible that the specialist misdiagnoses the patient with a false-positive which may induce stress and anxiety on the patient, besides the necessity to perform more invasive exams to confirm the diagnosis. Or in the worst case, the specialist might even misdiagnose the patient with a false-negative, which would leave the disease untreated for longer, allowing its development. Over 50% of women that are screened annually will be misdiagnosed with a false-positive, according to the National Cancer Institute[5].

1.1 Research Problem and motivation

The diagnosis of breast cancer in the early stages can increase the chance of success in treatments. However, the diagnosis is frequently mistaken, which leads the patients to

[1] https://www.cancer.org
[2] https://www.cdc.gov/
[3] http://www.breastcancer.org
[4] https://www.ons.gov.uk
[5] https://www.cancer.gov

take longer to discover the disease or the need to perform more invasive exams according to the National Cancer Institute[5]. To mitigate those mistaken diagnosis, several researches have been conducted on systems known as Computer-aided Diagnosis (CAD) to help doctors and radiologists better diagnose each patient. Many CAD systems have been proposed in the past years, for instance, (IBRAHIM et al., 2015; TURABIEH, 2016; BELCIUG; GORUNESCU, 2013). However, the inconsistency and low accuracy of those algorithms lower the acceptance of those kinds of CAD systems (DHEEBA; SINGH; SELVI, 2014). This problem happens because classifiers' performance depends heavily on the fine-tuning of its parameters.

Although many GA algorithms have been proposed to optimize the parameter of artificial neural networks in the context of detecting tumors in medical images, most of them have some limitation or constraint regarding the topology, like a maximum number of inputs for each node or the maximum number of nodes. On the other hand algorithm of NeuroEvolution of Augmented Topology (NEAT) does not present most of those limitations. Furthermore, NEAT has been tested on several problems reaching great results, where NEAT achieved better results than popular classifiers and GA solutions applied in their context. Some instances of those works are crash warning systems (STANLEY et al., 2005), feature selection for cancer detection in mammographic images (TAN; PU; ZHENG, 2014), and in predicting protein structural features (GRISCI; DORN, 2016). Although NEAT has been tested on several problems, it has not been thoroughly studied in the context, with only two papers found to the best of the author's knowledge.

Taking into account the aforementioned, this project seeks to answer the following questions:

- How does the solution NEAT generates compares to an optimal MLP and an MLP optimized by randomized search?
- How NEAT compares against popular classifiers on the problem at hand?
- How NEAT performs on the various datasets combined with different descriptors?
- Where NEAT stands when compared to other state of the art algorithms in the context of detecting tumors in medical images?

1.2 Research Goals

1.2.1 General Goals

- This work proposes a study of the efficiency of the algorithm of Neuroevolution of Augmenting Topologies (NEAT) compared against other classifiers to detect breast

cancer in medical images.

1.2.2 Specific Goals

- Investigate NEAT's performance in the context of tumor detection in medical images;
- Compare NEAT with MLP optimized by grid-search and randomized search;
- Compare NEAT with commonly used classifiers;
- Compare NEAT with state of the art neuroevolution algorithm found in the literature.

1.3 Document Structure

- Chapter 2 presents the concepts of evolutionary algorithms, learning algorithms and fundamentals of digital imaging.
- Chapter 3 presents the literature review, the related works and the difference between them and the proposed work.
- Chapter 4 describes the NeuroEvolution of Augmenting Topologies algorithm.
- Chapter 5 describes the methodology adopted in this project, the experiments setups, datasets and metrics for evaluation.
- Chapter 6 presents the results of the experiments and discusses it.
- Chapter 7 concludes the work and presents paths for further investigations.

2 Background

This chapter provides the theoretical basis for understanding this work. This chapter presents basic concepts regarding evolutionary computing, learning algorithms and fundamentals in digital imaging.

2.1 Evolutionary Computing

Evolutionary computing (EC) is the field of computer science that seeks to study and develop algorithms inspired by Darwin's theory of natural selection. In general, evolutionary algorithms take a population of individuals and uses the natural selection, using a fitness function, and evolve the best-fitted individuals (EIBEN; SCHOENAUER, 2002). This kind of algorithms is great for optimization problems, where the algorithm has to navigate through a great search space. There are three main different subfields of EC: evolutionary programming, evolution strategies, and genetic algorithms (EIBEN; SMITH et al., 2015). In the next section, it is going to be explained how genetic algorithms work.

2.1.1 Genetic Algorithms

As mentioned in the previous section, genetic algorithms (GA) is a subfield of EC. GA was firstly introduced by Holland in his book *Adaptation in Natual and Artificial Systems* (HOLLAND, 1975). The algorithm 1 shows the basics of a GA.

Algorithm 1: Pseudo-code for Genetic algorithm adapted from (EIBEN; SMITH et al., 2015)

```
1 pop = initialize population;
2 Evaluate organisms;
3 while termination condition has not been met do
4 |   Select parents;
5 |   Crossover selected parents;
6 |   Mutate offspring;
7 |   Evaluate new population;
8 |   Select individuals to survive to the next generation;
9 end
```

GA has six main components, they are representation, fitness evaluation, initialization, crossover, selection, and mutation.

- *Representation:* this component is the way the possible solutions can be encoded. In

the simple GA, the solutions are typically encoded as a binary string. The encoded solution is called genome, and each part of the genome represents a phenotype.

- *Fitness evaluation:* To compare one solution with another, it is necessary to implement a fitness function. The fitness function works as a heuristic to how close the solution is from becoming optimal.
- *Initialization:* At the beginning of a GA, a population of solutions is generated,this process is called initialization. In traditional GA, the solutions are generated randomly.
- *Crossover:* This is the process of combining the genome of two parents to generate an offspring. The idea is that different combining different organisms may result in an offspring that share the qualities of their parents, therefore improving their fitness.
- *Selection:* This component selects the individuals in the population to mate. In general, these organisms have the highest fitness in the population. Depending on the selection method, the higher the fitness, the higher the chances of the organisms pass on its genes.
- *Mutation:* To maintain the diversity of the population, this component mutates the genome of the offspring with the goal of introducing new behaviors, previously nonexistent in the population.

Genetic algorithms have some parameters that need to be set before the algorithms start. The population size defines the number of organisms in the population. The mutation rate defines the likelihood of a mutation occurs. If the mutation rate is too low, the population will tend to have low diversity, if the mutation rate is too high the population will not converge. The crossover rate defines how often the parents will mate. Some genetic algorithms introduced the idea of elitism. Elitism is when the best organisms are kept for the next generation.

2.1.2 Neuroevolution Algorithms

Neuroevolution (NE) is the evolution of artificial neural networks (NN). NE searches through a space of behaviors for a neural network with the goal of improving its performance on problem (STANLEY; MIIKKULAINEN, 2002). Traditional NE algorithms have a fixed fully connected topology, where there is a single hidden layer of neurons (STANLEY; MIIKKULAINEN, 2002). In this traditional NE, the algorithm optimizes the network weights. However, other NE approaches may also search through a space of topologies or

other hyper-parameters of an NN. These NEs that evolve both, topology and weights are called *Topology and Weight Evolving Artificial Neural Networks* (TWEANNs).

There are many techniques to perform TWEANNs encoding. The binary encoding also called direct encoding, is like the traditional GA representation shown in the Subsection 2.1.1. This type of encoding has some disadvantages because the number of genes is fixed, the limit for the number of nodes and connections for this type of approach has to be input by the user. Some TWEANNs use a graph encoding, where the connections are stored as a graph. This approach also has the limit on the number of nodes in the network. On this type of encoding, the crossover is usually done by swiping subgraphs. Other researchers believe that, in general, a crossover between organisms with different topologies to losses of behaviors. In contrast with the direct encoding, there is indirect encoding (STANLEY; MIIKKULAINEN, 2002). An example of indirect encoding is cellular encoding (GRUAU, 1993). Cellular encoding uses a specialized graph transformation language. The transformations are inspired by nature where it specifies cell divisions which can form the connections. The same gene can be used to form more than one connection. Thus the genome is more compact.

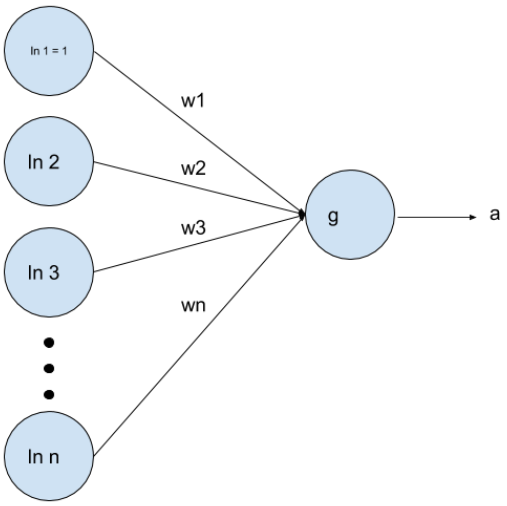
One of the challenges faced when developing TWEANNs is the competing conventions problem. This is when there is more than one way to represent the same solution. During a crossover, it is possible to generate a damaged organism (STANLEY; MIIKKULAINEN, 2002). An example is if two organisms had two hands and two feet, and during the crossover, the offspring ended up with four legs and the other with four hands. This problem is especially challenging to NE when trying to mate organisms with different topologies or even genome sizes. Some works propose a fixed or constrained topology (STANLEY; MIIKKULAINEN, 2002).

Another challenge faced by TWEANNs algorithms is how to protect a new structure from being removed from the population. When a new node or connection is added to a topology, usually the fitness tends to drop because the network has not had enough time to optimize the weights for this new structure. Thus, it is necessary to protect this innovation to give it time to be optimized before it can be compared against other topologies (STANLEY; MIIKKULAINEN, 2002).

2.2 Learning Algorithms

This section presents the supervised learning algorithms used in this work. Supervised learning works by training a model using a labeled database, that is a database where for every instance it is known to which class it belongs. Afterward, this model should be able to generalize which features lead to which class and be able to classify unseen examples. The classifiers used in this work are Multilayer Perceptron (MLP), Support

Figure 1 – A simple model of a perceptron



Source: The author

Vector Machines (SVM), and Random Forest (RF).

2.2.1 Multilayer Perceptron

Multilayer Perceptron is a class of artificial neural network(ANN) widely used in the literature for several types of problems. ANNs are inspired in some aspects of the brain, and it tries to mimic its structure regarding neurons, activation, and synapses (NICOLAS, 2015). One of the simplest ANN is a perceptron. (RUSSELL; NORVIG, 2010). A perceptron is an ANN with all the inputs connected to a single output node. The Figure 1 shows a simple model of a perceptron. The Equation 2.1 shows the formula of how a perceptron works. Where a is the output, g is the activation function, w is the weight of the link, and in is the input node.

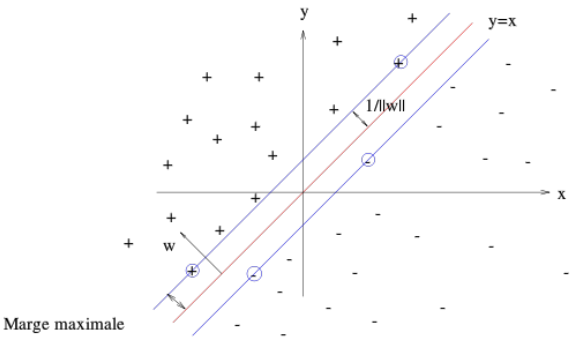
$$a = g(\sum_{i=1}^{n} w_i \cdot in_i) \tag{2.1}$$

A single perceptron is limited to compute a single linear combination of its weights and inputs. With the goal of allowing ANNs to solve more complex problems, MLPs uses intermediate layers of perceptrons called hidden layers (NICOLAS, 2015). Where each new layer is the input to the next layer.

2.2.2 Support Vector Machines

A support vector machine is a supervised classifier that tries to maximize the margin between the classes. SVM was first described in (BOSER; GUYON; VAPNIK, 1992), and it can resolve either linear or non-linear problems (BELL, 2014). The idea

Figure 2 – Optimal hyperplane with margins to illustrate SVM method



Source: (SYLENIUS, 2016)

behind an SVM is to create an optimal hyperplane that splits the data with the highest distance between the supporting vectors. The supporting vectors are the sample that is placed in the boundary region between the classes. The algorithm finds the supporting vectors and then it searches for the hyperplane that optimally divides them.

2.2.3 Random Forests

Random Forest is a learning model that consists of a combination of decision trees, where each tree is trained using a random sample of the date, then each tree classifies the input into a class, the class with the majority of the votes is chosen (PAL, 2005). It does that by using a technique called bootstrapping. This technique is used to generate a number of subsets by taking the n random samples from the dataset, where n is the number of samples in the dataset, but allowing for repetition (SUTHAHARAN, 2016). In the testing phase, the random forest algorithm uses a technique called bagging to average the prediction of the decision trees to generate a single response for the random forest. Given a new input x, the new data is classified by each of the N decision trees results in N predictions. The bagging algorithm suggests the random forest result to the class which has the majority of votes (SUTHAHARAN, 2016).

2.3 Fundamentals of Digital Images

This section presents the fundamentals of digital images. It is going to be addressed how images are represented in a computer, and the descriptors used in the image datasets used in this work.

2.3.1 Image Representation

A digital image is defined by sampling continuous data and storing those samples in rectangular arrays of pixels in the form of (x, y, u) where (x, y) are the location of the

point and u is the value. The value of u can be a vector of channels where each channel can represent the intensity of each primary color as the case of RGB color space, or brightness or contrast (KLETTE, 2014).

An image can be represented using different color spaces. There are many different color spaces in the literature. Each of them their advantages and disadvantages depending on the purpose of its use. The RGB color space is one of the most commonly used color space. It consists of three channels, each one of them representing the intensity of the colors red, green, and blue. The HSI color space uses cuts in the RGB cube orthogonally. The HSI stands for Hue, Saturation, and Intensity, where the hue is the angle on the disk, then the saturation is the grey-level diagonal of the RGB cube (KLETTE, 2014).

2.3.2 Image Descriptors

This section describes the image descriptors addressed in this work. An image descriptor is a set of computed property values of an image (KLETTE, 2014).

2.3.2.1 Gray-level Co-occurrence Matrix

The Gray-level Co-occurrence Matrix (GLCM) is a second-order estimator proposed by (HARALICK; SHANMUGAM et al., 1973). GLCM is a matrix computed with the frequency of a gray-level occurring next to another. The probability of going form a gray level i to a gray level j separated by a distance d and the direction θ (normally θ is 0°, 45°, 90° or 135°) is the value of co-occurrence matrix elements. This probability is computed by scanning the image in direction θ and the co-occurrence accumulated in the GLCM (GRAÑA, 2012). For instance, the GLCM table with a distance d equal to two and a θ of zero can be computed by making a matrix LxL where L is the number of different gray-levels present in the image. Each cell of the matrix is filled with the number of times a pixel with the gray-level corresponding to its row number appeared to the right of a pixel with the gray-level corresponding to its columns number (HARALICK; SHANMUGAM et al., 1973). From this table, several statistical analysis can be made. In (HARALICK; SHANMUGAM et al., 1973) proposed 14 statistical analysis such as angular second moment, contrast, correlation among others.

2.3.2.2 Local Binary Pattern

The Local Binary Pattern (LBP) was first presented by (HE; WANG, 1990). Initially, LBP was proposed to describe the texture. However, LBP became very popular because of its good performance and computational simplicity. The LBP operator labels each pixel with an integer. Each of these labels is called LBP pattern, and they are computed by comparing each of its adjacent pixels in a 3x3 area regarding intensity (FAN ZHENHUA WANG, 2015). The center pixel is compared to its neighboring pixels, if the

pixel intensity is grater or equal to the center pixel, then it's labeled as 1, and if it is smaller than the center pixel's intensity it is labeled as 0. This returns a binary string with eight elements, making a total of 256 different possible labels. This bit string is the converted to decimal. Afterward a histogram is computed with the resulting label for each pixel (AWAD; HASSABALLAH, 2016).

2.3.2.3 Zernike's Moments

Zernike's moments, proposed in (ZERNIKE, 1934), are a class of orthogonal moments. It can be defined using an arbitrary order. The higher the order, the finer the details carried from the image. The Zernike's moments are invariant to rotation and reflection (HSE; NEWTON, 2004). Zernike polynomials are defined over the interior of a disk unit.

$$V_{nm}(x, y) = V_{nm}(\rho, \theta) = R_{nm}(\rho)e^{jm\theta} \tag{2.2}$$

$$R_{nm}(\rho) = \sum_{s=0}^{\frac{n-|m|}{2}} (-1)^s \frac{(n-s)!}{s!(\frac{n+|m|}{2}-s)!(\frac{n-|m|}{2}-s)!}\rho^{n-2s} \tag{2.3}$$

Where n-|m| is even, |m|≤n, and ρ=$\sqrt{x^2+y^2}$. With the image function projected onto the basis set, the Zernike moment of order n with repetition m is:

$$Z_{n,m} = \frac{n+1}{\pi}\sum_x\sum_y f(x, y)V_{mn}(x, y), x^2 + y^2 \leq 1 \tag{2.4}$$

Since Z_{00} and Z_{11} are the same for normalized image, they are not used, so the features extracted of order n starts on the second order moments and go up to the later order moments (HSE; NEWTON, 2004).

3 Literature Review

Among the algorithms already used to classify tumors in medical images, artificial neural networks (ANN), in particular, MLPs have been widely used (IBRAHIM et al., 2015). Even though ANNs have been successfully used in different applications, its performance depends on the fine tune of its hyper-parameters, like the number of hidden nodes, weights, among others. Thus, many works have used genetic algorithms to optimize the classifiers to increase its performance in detecting tumors in medical images.

The work proposed by (IBRAHIM et al., 2015) proposes the adoption of an MLP with a multi-objective differential evolution (MODE) algorithm to diagnose breast cancer. In this paper, MODE is used to optimize the accuracy and complexity of the topology (for instance the number of hidden nodes) by choosing the pareto optimal. It was used the Wisconsin Breast Cancer Datasets to test it, which classifies between benign and malignant lesions. This database has 699 sequences, 65.5% of which are benign and 34.5% are malignant. The goal of the optimization algorithm is to minimize the complexity of the structure of the MLP and the error rate. The objective functions were the mean square error (MSE) and the number of hidden nodes. They experimented their algorithm using 10-fold cross-validation and the maximum of 1000 interactions. They achieved 97.51% of accuracy with 1.69 of standard deviation. The mean number of hidden nodes was 4.0 with a standard deviation of 1.24.

DHEEBA; SINGH; SELVI (2014) proposed the use of Particle Swarm Optimized Wavelet Neural Network (PSOWNN) to detect abnormalities in digital mammograms. The method is tested on a real clinical database of 216 mammograms collected from 54 patients, four of each. The database was classified as normal (without abnormalities), benign or malignant lesion. The particle swarm optimization (PSO) is used to improve the classification accuracy of the wavelet neural network(WNN), decreasing the error rate. For the preprocessing of the images, it was applied a technique of global limiarization, to split the region of interest (ROI) from the rest of the image. The Laws Texture Energy Measures (LTEM) was used to extract 25 features from the images. The PSO is used to search through a space of topologies, learning rates and term momentum. Afterward, every individual of the population is trained using backpropagation to compute the fitness. The optimization of the WNN was done through 100 generations with a population size of 50 individuals. The number of hidden neurons varied from 31 to 200, the learning rate and term momentum varied from 0 to 1. The WNN generated had 116 hidden nodes, a learning rate of 0.00127 and a term momentum of 0.9283. According to (DHEEBA; SINGH; SELVI, 2014), the propose achieved an accuracy of 93.671%, and the misclassification rate was 0.063291. Depending on the problem, PSOWNN may have a high computational cost, as

it has to execute a backpropagation on every individual from the population for every generation.

ZOHRA; NACÉRA (2013) uses Multi-Population Genetic Algorithm (MPGA) to optimize radial base function neural networks (RBF NN) to detect tumors in mammographic images. The K-means is used to find the centroids for the RBF NN, and the MPGA is used to evolve the weights of the RBF's second layer. The fitness used was the mean squared error. To evaluate the performance of the proposed algorithm, it was used a dataset of 20 mammographic images. According to (ZOHRA; NACÉRA, 2013), the accuracy reached by the algorithm was of 100%.

In the work of (BHARDWAJ; TIWARI, 2015), they propose a Genetically Optimized Neural Network (GONN). They used genetic programming (GP) to generate neural networks capable of classifying breast lesions as benign or malignant. The fitness function used was the mean squared error. To evaluate the algorithm, it was used 10-fold cross-validation. The algorithm was tested on the Wisconsin Breast Cancer Database (WBCD). GONN achieved an accuracy of 99.26% with a standard deviation of 0.602 and a maximum of 100% of accuracy.

The work of (AHMAD et al., 2015) adopts the genetic algorithm for feature selection and the optimization of the topology of a neural network multilayer-perceptron (MLP). The training was done using standard backpropagation. The stop condition was when the error on the validation dataset grew for more than six iterations. The experiment used a population size of 15, and an elitism size of 3. The data was split in 50% for training, 25% for validation and 25% for testing. Each experiment was executed three times. They reached an accuracy of 98.29% with a maximum of 15 generations. This proposal has the same disadvantage as the work of (DHEEBA; SINGH; SELVI, 2014), as it has to execute the training algorithm multiple times for each individual of the population.

The work of (BELCIUG; GORUNESCU, 2013) proposes a genetic algorithm to optimize the weights of an MLP with a fixed topology. The algorithm is tested on four databases: the Wisconsin Prognostics Breast Cancer (WPBC1) with 683 sequences, Wisconsin Prognostics Breast Cancer (WPBC2) with 569, Ljubljana Recurrence Breast Cancer (LRBC) with 286 sequences and the Wisconsin Recurrence Breast Cancer (WRBC) with 198 sequences. They tested five different types of crossover. It was used a 10-fold cross-validation to evaluate the accuracy of the algorithm. The experiment was executed 106 times, for each database and each model. The accuracy of the algorithm varied from 80.43% on the LRBC to 93.58% on the WPBC2. The propose was tested on a variety of datasets, obtaining good results.

AHMAD et al. (2012) adopts Cartesian Genetic Programming (CGP) to evolve artificial neural networks to detect breast cancer. The algorithm uses a 2D vector to represent the nodes and connections. The CGP evolve not only the weights but also the

topology, and it chooses the most appropriated activation function. In the experiment, it was used a population of 10 organisms with a mutation rate of 10%. The fitness function was a sum of false-positives and false-negatives. The experiment was executed 24 times, running through 100,000 generations. The experiments varied in the maximum number of nodes and the maximum number of inputs per node. It was chosen 200 randomly picked patients for the training an testing group. The algorithm obtained a maximum accuracy of 98% on the testing dataset, although the method was tested on a very small dataset. The CGPANN presents some limitations. There is a maximum number of nodes and a maximum number of input per node. Furthermore, the algorithm does not execute a crossover on the population, which could lead to a local minimum according to (MANNING; WALSH, 2013).

KHAN et al. (2014) suggests the use of Wavelet Neural Networks (WNN) to be optimized by the Cartesian Genetic Programming (CGP). The algorithm was tested using a database of 200 mammographic images. The algorithm was evaluated using a training dataset with 70% of the sequences and 30% for testing. The method was also tested using 10-fold cross-validation. In this work, the CGPWNN reaches an accuracy of 89.57%. The authors also compare the technique with NEAT and CGPANN, which reach an accuracy of 89.11%. The CGPWNN was validated using a small dataset. Furthermore, it presents some of the same limitations as the proposed by (AHMAD et al., 2012).

In the work of (MANNING; WALSH, 2013), they propose an improvement on the CGPANN of (AHMAD et al., 2012). They introduced the Radial Basis Function neural network and crossover. They reached an accuracy of 97.19% against the 96.0% from the CGPANN proposed by (AHMAD et al., 2012). Although the algorithm has achieved a high accuracy, it presents some constraints regarding the maximum number of nodes and the maximum number of inputs per node.

In addition to the work presented previously, it should be highlighted the research developed by (TURABIEH, 2016). In his work, it was used the Wisconsin Breast Cancer Dataset from the UCI Machine Learning Repository to compare the performance of NEAT and an MLP trained using Backpropagation on the task of detecting cancer. Although NEAT reached better results than the MLP, the results of the experiment are inconclusive, as it was tested on only one small dataset.

The Table 1 presents the comparison between the proposed approach and the related work.

Adversely to the related works that used datasets with a small number of instances, this work makes use of a subset of IRMA dataset, with 2796 instances. Furthermore, the proposed approach does not have limits and constraints regarding the topology of the network, like the number of hidden nodes or number inputs per node, unlike several of the presented related works. When the approach has a fixed topology, it is necessary to find

Table 1 – Details of the difference between the related work and the proposal.

Reference	**Dataset**	**Approach**
(IBRAHIM et al., 2015)	WBCD	MLP optimized by a MODE
(DHEEBA; SINGH; SELVI, 2014)	Real clinical database	WNN optimized by PSO (Topology; BP parameters)
(ZOHRA; NACÉRA, 2013)	Does not state	MPGA to optimize weights of second layer of the RBF
(BHARDWAJ; TIWARI, 2015)	WBCD	GP to generate MLP topology
(AHMAD et al., 2015)	WBCD	GA for feature selection and to generate MLP topology
(BELCIUG; GORUNESCU, 2013)	WDBC WPBC1 WRBC LRBC	GA to optimize weight of an MLP with fixed topology
(AHMAD et al., 2012)	WDBC	CGP optimize topology and weights of an MLP
(KHAN et al., 2014)	Does not state	CGP optimize an WNN
(MANNING; WALSH, 2013)	WDBC	Same as (AHMAD et al., 2012) + RBF and crossover
The proposal	WBCD WDBC IRMA ISIC	GA to optimize topology and weights + uses speciation and historical markers

the optimal topology in the process of trial and error, which can be a costly process. The constraints on the number of hidden nodes and number of inputs per node may prevent the algorithm from finding the optimal solution, in the case when the optimal solution has more nodes than the constraints. Therefore, it is necessary to study techniques that present good results in the detection of tumor in medical images and test them on a variety of dataset with a great number of instances.

In the work of (STANLEY; MIIKKULAINEN, 2002), it was executed a series of experiments to assert NEAT's efficacy in the task of finding solutions close to the optimal with minimal topological complexity. During those experiments, NEAT's performance was better than traditional methods to solve problems like XOR (a nonlinear problem), pole balancing and balancing two poles. NEAT is considered one of the most popular constructive neural networks (TURABIEH, 2016), and it has been tested in several problems like feature selection for cancer detection in mammography images (TAN; PU; ZHENG, 2014), crash warning systems (STANLEY et al., 2005), and in predicting protein structural features (GRISCI; DORN, 2016).

Although NEAT has been performed with a variety of problems, it has not been thoroughly investigated in the context of detecting tumors in medical images. NEAT searches through a topology and weight space for a determined problem. Some works

approach this problem only optimizing its weight with a fixed topology, which has to be set by the user by trial and error. As mentioned earlier, NEAT does not have most of the limitations commonly found in other approaches. Therefore, NEAT presents qualities that suggest the capacity to produce good results in the context of detecting tumors in medical images.

4 NeuroEvolution of Augmenting Topologies

Several optimization algorithms have been studied to train neural networks for classification problems, as shown in the Chapter 3. Among the cited works, it is important to highlight neuroevolution algorithms (STANLEY; MIIKKULAINEN, 2002). These algorithms have been widely studied because of their power to optimize the topology and hyper-parameters of neural networks.

Neuroevolution algorithms are a subcategory of genetic algorithms that can optimize neural networks. Traditional neuroevolution algorithms are usually used to optimize hyper-parameters such as the weights of the links between nodes while using a fixed topology. Although the topology of the network plays a big role in the efficiency of the network.

NEAT is a neuroevolution algorithm that builds the topology of the ANN incrementally. The evolution begins with simple organisms, with only the input layer fully connected to the output layer. Those organisms then evolve through the generations, adding new nodes and links at the same time as it optimizes the weights of the network. To achieve this, NEAT needs to solve some problems:

- How to genetically represent a topology in a way that allows the crossover of different topologies?
- How to protect topological innovations that need time to be optimized, so they are not eliminated from the population prematurely?
- How to minimize the network‘s structure without the need to have restrictions or a fitness function that measures the topology’s complexity?

The algorithm 2 presents NEAT’s pseudo-code. Each part of this algorithm will be detailed in the next paragraphs, and the questions mentioned above will be answered.

To represent ANN’s genome, NEAT uses a direct coding (e.g., all links and nodes of the network are represented on the genome) as shown in the Figure 3. NEAT splits the genome into two lists, one representing the links between the nodes and the other representing the nodes. The node genes contain information if the node is a sensor, a hidden node or a node from the output layer. The link genes represent which node is connected to which, the weight of the connection, which is the output node, the innovation number and if the gene is active or not.

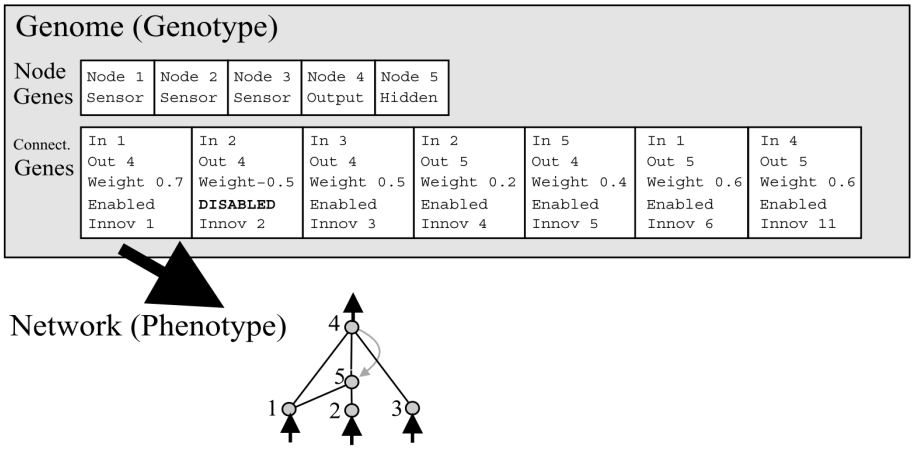
There are three kinds of mutation: one to mutate the weight of the link; other adds new nodes and another to add new links as shown in the Figure 4. The add link mutation

Algorithm 2: NEAT Steps

input : max_it, maximum number of generations
output : optimized population

```
1  pop = start population;
2  n_geracoes = 0;
3  while n_geracoes ≥ max_it do
4      calcular fitness de pop;
5      species = split population in species;
6      for each specie ∈ species do
7          select parents;
8          execute crossover;
9          mutate organisms;
10         add new organisms to pop;
11         select organisms to survive from pop;
12     end
13 end
```

Figure 3 – Translation from a genome to the topology of a NN



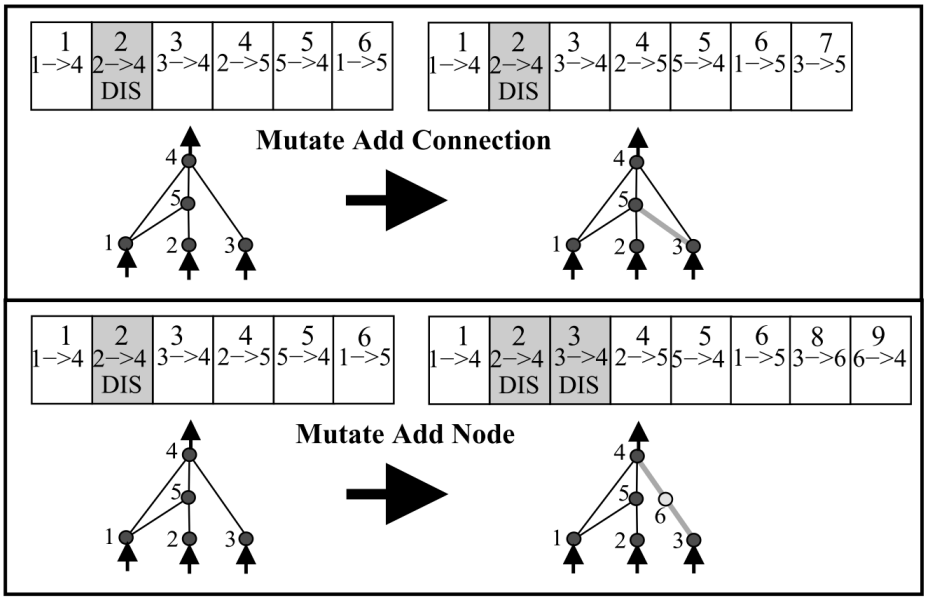
Source: (STANLEY; MIIKKULAINEN, 2002)

adds a new link between two nodes that weren't previously connected. In the add node mutation, an existing connection is split in two. Thus, the original connection is disabled, and the two new links are made connecting the input node from the old connection to the new node, setting the weight of the connection to one.

The other connects the new node to the output node from the old connection using the weight of the old connection. Therefore, avoiding creating nodes that are not linked to any other node.

To represent the topology in a way that allows the crossover between two different topologies, NEAT makes use of historical markings. NEAT tracks historical markers assigning an innovation number every time a new link between two previously unconnected

Figure 4 – Add link and add node mutation scheme



Source: (STANLEY; MIIKKULAINEN, 2002)

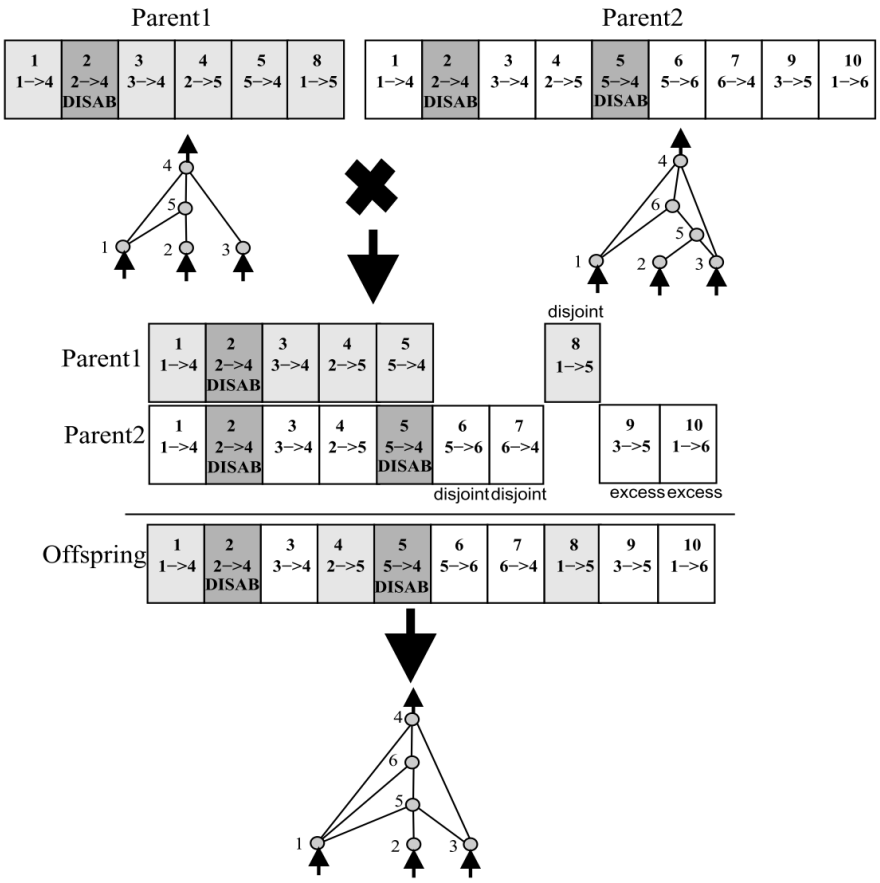
nodes is made. The innovation numbers are used to identify which genes can be matched, as two genes with the same innovation number share the same historical origin, therefore representing the same connection. NEAT uses historical markers to make the crossover of two organisms with different topologies. The crossover is made by aligning the genes that have the same innovation number. The genes that share the same innovation number in both parents are chosen randomly. Whereas the genes that are not common to both parents are selected from the parent with the highest fitness, as shown in the Figure 5.

NEAT uses speciation to protect topological innovations that need time to be optimized, avoiding them to be excluded from the population prematurely. When a new node or link is created, the fitness of the organism may drop, as the innovation has not had time to be optimized. Speciation split the population in niches (e.g., species), grouping similar organisms together to avoid organisms with innovations from being compared with all the population. To group similar organisms, NEAT uses the historical markers. The similarity between two organisms is calculated based on many historical markers they share together, as shown in the Equation 4.1; then a threshold is used to split them into species.

$$\sigma = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 \dot{W} \tag{4.1}$$

Where σ is the compatibility distance between the organisms, E is the number of excess, D is the number of disjoint, $\bar{W}$ is the weight difference of matching genes, c_1, c_2, c_3 are the importance of each of the three factors. The excess genes are those genes that do

Figure 5 – Crossover between two organisms



Source: (STANLEY; MIIKKULAINEN, 2002)

not match in the end and the disjoint are those that do not match in the middle.

To minimize the ANN's topology without having a fitness that calculates complexity or using some constrains on the topology, NEAT begins with simple organisms with zero hidden nodes. Then, as the organisms evolve, they become more complex with the generations, adding new nodes and new links when needed. If a new connection or node represents a useful new behavior, it will be selected. That way, NEAT's minimize the complexity of the topologies and maximizes the performance of the classifier.

5 Material and Methods

This chapter presents the details of the methodology adopted to evaluate the classifiers described in Section 2.2 and Chapter 4 in the context of tumor detection in medical images.

5.1 Methodology

We have compared the neural network NEAT optimized against a multilayer perceptron (MLP) neural network optimized using grid search and randomized search. The grid search and randomized search optimized the MLP changing only the number of nodes on the hidden layer and keeping the learning rate and term momentum default. As NEAT optimizes not only the topology but also the weights of the links, it is possible to result in scores higher than those found by the grid-search optimized MLP. The highest number of nodes searched during the optimization process was the same as the highest number of nodes NEAT reached while optimizing the network.

NEAT was also compared to a support vector machine (SVM) and a Random Forest. It was used grid search on both algorithms to optimize their hyper-parameters.

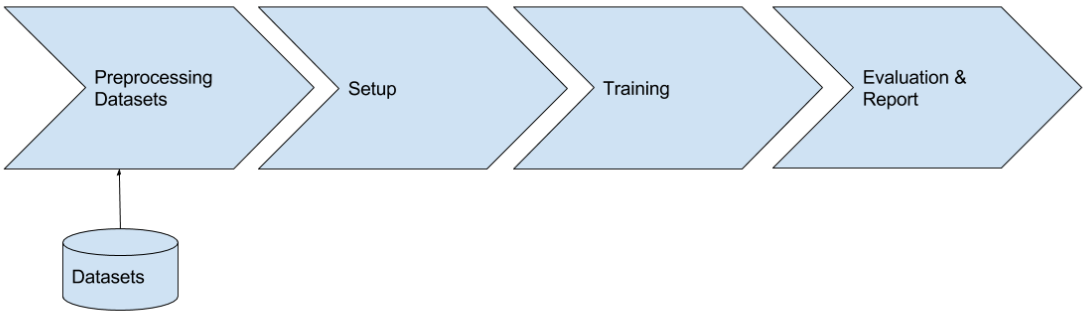
NEAT was performed on all datasets running through 1000 generations with the add new link rate of 10%, add new node rate of 1%, compatibility threshold of 6.0, and population of 50 organisms. On ISIC datasets, NEAT run 3000 generations.

We also compared NEAT against six neuroevolution methods found in the literature: MLP/GA Hibride, proposed by (BELCIUG; GORUNESCU, 2013), CGPANN-RBF crossover proposed by (MANNING; WALSH, 2013), CGPANN, proposed by (AHMAD et al., 2012), GONN, proposed by (BHARDWAJ; TIWARI, 2015), GAANN RP, proposed by (AHMAD et al., 2015), and Intelligent Multi-Objective classifier (IMOC), proposed by (IBRAHIM et al., 2015). Where the first three algorithms were tested using WDBC dataset, and the last three were performed on WBCD.

All the experiments were executed ten times, and the mean of each metric and standard deviations were taken. The MLP, SVM, and Random Forest was trained using Scikit-learn's library SKlearn in python (PEDREGOSA et al., 2011). It was used cross-validation with ten folds on all classifiers. In the following sections, it is going to be presented the results regarding each dataset (IRMA, ISIC, WDBC, WBCD).

To guarantee reproducibility and fairness of the evaluation of the techniques, all experiments followed the pipeline described in the Figure 6. The pipeline is divided into five modules, each of them will be described in the following subsections.

Figure 6 – Experiment's Pipeline



Source: The author

5.1.1 Preprocessing Datasets

All the dataset went through a preprocessing phase. In this phase, they were standardized. Standardization makes the data have zero-mean and unit variance. This process is widely used in machine learning to improve performances (DAWSON; WILBY, 2001). The Equation 5.1 present how the data is standardized. The IRMA dataset took an extra step in this module. As the IRMA dataset has three classes, it was created another dataset taking all the instances where the label was benign or malignant and counted as a single label "with lesion", and the remaining of the dataset was labeled "normal tissue". Then both versions of the dataset were standardized.

$$x = \frac{x - \mu}{\sigma} \tag{5.1}$$

Where x is the data, μ is the mean, and σ is the standard deviation.

5.1.2 Setup

In this step, the user sets up the experiment to be executed. In this module, the user chooses which classifier will be evaluated and what dataset will be used. The experiments done using NEAT algorithm has some extra parameters: the fitness function, the number of generations, the mutation probabilities and the population size.

5.1.3 Training

This step takes as a parameter which classifier will be tested with which dataset. The chosen classifier is then trained using k-fold cross-validation. The data is split into ten folds, and then the classifier is tested using ten combinations of these chunks of data, each time one of those folds is used for evaluation, and the other nine are used for training. The predictions of the models are then stored to allow the next module calculate the metrics for evaluation. All the classifiers were tested with all the datasets.

5.1.4 Evaluation and Report

This module is responsible for evaluating the classifiers and generate a report with the metrics calculated from each classifier. This module takes the predictions from each classifier collected in the Subsection 5.1.3 and calculates the metrics described in the following sections. Then a report is stored in a .csv file for further evaluation.

5.2 Datasets

In this section, it is presented the datasets used in the experiments. A total of four datasets was used: a dataset of mammography images, one dataset of digital skin images, and two datasets of features extracted from digital images. The Table 2 presents the details regarding each dataset.

Table 2 – Details about the datasets.

Dataset	Document Length	Classes
Image Retrieval in Medical Applications (IRMA)	2796	3
International Skin Imaging Collaboration (ISIC)	200	2
Wisconsin Diagnostic Breast Cancer (WDBC)	569	2
Wisconsin Breast Cancer Dataset (WBCD)	699	2

5.2.1 Image Retrieval in Medical Applications Dataset

The Image Retrieval in Medical Applications (IRMA) is a dataset containing the region of interest of mammograms classified by radiologists. It was first introduced in (OLIVEIRAA et al., 2008; DESERNO et al., 2012) and the dataset is classified as normal (without lesion), benign lesion and malignant lesion. This dataset is the union of four databases of mammographic images: the Mammographic Image Analysis Society Digital Mammogram Database (MIAS), the Digital Database for Screening Mammography (DDSM), the Lawrence Livermore National Laboratory (LLNL), and routine images from the Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen. The dataset was classified by tissue type, tumor staging, and lesion description and breast imaging reporting

and data system (BI-RADS) according to the American College of Radiology. The dataset is classified into four tissue types: adipose tissue (type I), fibroglandular tissues (type II), dense tissue (type III), and extremely dense tissue (type IV). The dataset contains 10,509 images. In this study it was used a subset of 2,796 images, containing images from all types of tissues. The Table 3 describes the division of the dataset.

Table 3 – Details about IRMA dataset.

Normal	Benign	Malignant	Total
932	932	932	2796

As mentioned in section 5.1.1, it was made two studies using the IRMA dataset. The first one, using the dataset as it is, with all three classes. The second study was made by turning the IRMA dataset into a binary problem. The dataset was split into two classes: normal (without lesion) and with lesion (counting the classes benign and malignant as one). The division of this dataset is shown in the Table 4.

Table 4 – Dataset with normal and a lesion.

Normal	Lesion	Total
932	1864	2796

5.2.2 International Skin Imaging Collaboration Dataset

The International Skin Imaging Collaboration Melanoma Project (ISIC) is a dataset from the International Society for Digital Imaging of the Skin. As the data from the dataset is unbalanced, it was used a balanced subset for the experiments. The dataset consists of digital images of skin lesions classified in benign lesion or malignant lesion. The ISIC dataset is classified in benign (Negative) and malignant(Positive) lesion, and it is distributed as shown in the Table 5.

Table 5 – Details about the ISIC dataset.

Negative	Positive	Total
100	100	200

5.2.3 Wisconsin Diagnostic Breast Cancer Dataset

The Wisconsin Diagnostic Breast Cancer (WDBC) (WOLBER; STREET; MANGASARIAN, 1995) is a dataset containing features extracted from digitalized images using Fine Needle Aspirate (FNA). The dataset is classified in benign lesions and malignant lesions. The WDBC dataset has a total of 30 features. This dataset has been widely used

in the literature. This dataset was used by three related work as seen in the Chapter 3 and it was chosen to compare NEAT with these works. The WDBC dataset is classified in benign (Negative) and malignant (Positive) lesion, and it is distributed as shown in the Table 6.

Table 6 – Details about the WDBC dataset.

Negative	Positive	Total
357	212	569

5.2.4 Wisconsin Breast Cancer Dataset

The Wisconsin Breast Cancer Dataset (WBCD) was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. The WBCD contains 699 sequences with nine features. The dataset was introduced by (MANGASARIAN, 1990; WOLBERG; MANGASARIAN, 1990; MANGASARIAN; SETIONO; WOLBERG, 1990; BENNETT; MANGASARIAN, 1992). The WBCD is classified as a benign lesion and malignant lesion. Table 7 shows the distribution of the dataset. This dataset was used in several works in the literature, and it was chosen to compare NEAT against three of these works.

Table 7 – Details about the WBCD dataset.

Negative	Positive	Total
458	241	699

5.3 Descriptors

To use the image datasets IRMA and ISIC, it is necessary to use some descriptors to extract features from the images. These descriptors are explained in the Chapter 2. The Gray Level Co-Occurrence Matrix (GLCM) (HARALICK; SHANMUGAM et al., 1973) was used to extract 13 features from the images. The GLCM was used to extract features from the IRMA (both multi-class and binary) and ISIC datasets. The Local Binary Patterns (LBP) (HE; WANG, 1990) to extract 26 features. The LBP descriptor was used to extract features from the IRMA (both multi-class and binary) and ISIC datasets. Zernike's descriptor was used to extract 25 features from the images. The Zernike's descriptor was used with Otsu's thresholding method. This descriptor was used on the IRMA dataset.

5.4 Evaluation Metrics

This section introduces the evaluation metrics used to assert the quality of the classifiers. To assess the quality and compare different classifiers, it is necessary to use a

metric to understand how the classifiers differ from one another. The metrics adopted in the work are accuracy, f1-score, precision, and recall. To calculate the previously mentioned metrics, it is necessary to count the number:

- **True positive (TP):** the number of instances labeled as positive correctly classified.
- **True negative (TN):** the number of instances labeled as negative correctly classified.
- **False positive (FP):** the number of instances labeled as negative wrongly classified.
- **False negative (FN):** the number of instances labeled as positive wrongly classified.

Each metric used in this work is defined as follows:

- **Accuracy:** Accuracy is the percentage of the predictions correctly classified, independent of the class (NICOLAS, 2015). The accuracy is defined in the Equation 5.2.

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn} \tag{5.2}$$

- **Precision:** Precision is the percentage of the predictions of correctly classified as positive of the class considered positive (NICOLAS, 2015). Precision is defined by Equation 5.3.

$$Precision = \frac{tp}{tp + fp} \tag{5.3}$$

- **Recall:** Recall (also known as sensitivity) is the percentage of the sequences labeled as positive correctly classified as positive (NICOLAS, 2015). The definition of recall is shown in Equation 5.4.

$$Precision = \frac{tp}{tp + fn} \tag{5.4}$$

- **F1-Score:** F1-Score (also known as F-score or F-measure) is the harmonic measure the precision and the recall. F-score can be defined by Equation 5.5. This score ranges from 0 (worst score) to 1 (best score) (NICOLAS, 2015).

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{5.5}$$

F-score can also be used in multinomial classification. The precision and recall are calculated for all classes, and then the average of them is taken to produce one mean

precision and recall (NICOLAS, 2015). The precision and recall for each class are computed by taking the class at hand as the positive and all the others as negative. There are two formulas to compute precision and recall for a multi-class problem.

- **Macro:** the precision and recall are computed for each class, and then the average is taken
- **Micro:** the sum of the numerators and denominators for each class is taken, and then the precision and recall are computed.

This project is going to use the macro formulas of precision and recall for the multinomial classifications and the regular formulas for the binary classifications. The formulas for precision and recall macro can be defined in Equation 5.6 and Equation 5.7.

$$MacroPrecision = \frac{1}{c} \cdot \sum_{i=0}^{c-1} \frac{TP_i}{TP_i + FP_i} \tag{5.6}$$

$$MacroRecall = \frac{1}{c} \cdot \sum_{i=0}^{c-1} \frac{TP_i}{TP_i + FN_i} \tag{5.7}$$

Where c is the number of classes in the dataset.

5.5 Experimental Setup

This section describes the experimental setup for all the experiments. All the experiments were run ten times, and the mean of all metrics was taken. It was used the k-fold cross-validation. The k-fold cross-validation splits the data into k groups, called folds. Each fold consists of one Kth of all the sequences picked randomly. One of these folds is used for testing and the other $K - 1$ are used for training. Then the next fold is used for testing, and so on until all the folds have been used for testing (NICOLAS, 2015).

For this project, it was used ten folds for the cross-validation. All the experiments using NEAT was on Virtual Machines on Google Cloud [1] on a CPU with two cores and 1.80 GB of RAM. All the experiment using the other classifiers was on Virtual Machines on Google Cloud [1] on a CPU with eight cores and 7.20 GB of RAM. The NEAT algorithm was developed using the C++ programming language, and it was developed by (STANLEY; MIIKKULAINEN, 2002), and the algorithms to test NEAT on the datasets were developed by the author. The scripts for the processes the datasets and to test the other classifiers were developed using the Python programming language version 2.7, and it was used the libraries as described as follows:

[1] https://cloud.google.com

- *Pandas* is an open source Python library for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables and time series. (MCKINNEY, 2010).
- *Scipy* is an open source Python library used for scientific computing and technical computing (JONES et al., 2001).
- *Numpy* is a library for the Python programming language that adds support for large, multi-dimensional arrays and matrices, and several collections of high-level mathematical functions to operate on these arrays (WALT; COLBERT; VAROQUAUX, 2011).
- $Scikit-learn$ is a machine learning free open source library for the Python programming language. It was developed to interact with the Python numerical and scientific libraries NumPy and SciPy (PEDREGOSA et al., 2011). This library was used to train the models (MLP, SVM, and Random Forest) using grid-search and randomized search with k-fold cross-validation.

The MLP was optimized using grid-search and randomized search. Both algorithms optimized the number of hidden nodes. As NEAT does not have a limit on the number of hidden nodes, it was used a range from 1 to the maximum number of hidden nodes NEAT used for each dataset on the grid-search and randomized search. The Table 8 presents the parameters of the grid-search and randomized search for each dataset.

Table 8 – Parameter for the grid-search and randomized search for MLP

Dataset	Range of hidden nodes
Multi-class IRMA GLCM	$1-27$
Multi-class IRMA LBP	$1-30$
Multi-class IRMA Zernike	$1-26$
ISIC GLCM	$1-72$
ISIC LBP	$1-68$
Binary IRMA GLCM	$1-25$
Binary IRMA LBP	$1-30$
Binary IRMA Zernike	$1-27$

It was used grid-search to optimized the hyper-parameters of the SVM and Random Forest. Following the guidelines provided by (HSU; CHANG; LIN,), on the SVM the grid search on the parameters γ and C. The parameter γ assumed values from 2^{-15} to 2^{4} and C assumed the values from 2^{-5} to 2^{15}. The random forest grid search optimized the number of estimators, ranging from 10, 30 and then from 50 to 650, going 25 by 25. The parameters for the grid-search of the random forest was chosen after noticing that increasing the number of estimators did not improve significantly the performance of the algorithm and adding a much longer processing time.

The parameters used on NEAT was the default given with the code provided by (STANLEY; MIIKKULAINEN, 2002). These parameters are as shown in the Table 9. The number of generations used on all datasets was 1000 generations, except for ISIC datasets, where it was used 3000 generations. Moreover, the fitness function used in those experiments was the F-score macro.

Table 9 – NEAT Hyper-Parameters.

Parameters	Value
Compatibility Threshold	6.0
Mutate Add Node Probability	0.01
Mutate Add Link Probability	0.1
Mutation Probability	0.2
Mutate Link Weights Probability	0.8
Mutate Gene Re-enable Probability	0.05
Mutate Toggle Enable Probability	0.1
Cross-over Probability	0.25
Interspecies Mate Probability	0.001
Disjoint Coefficient	1.0
Excess Coefficient	1.0
Population	50

6 Results

6.1 Image Retrieval in Medical Applications Dataset

As mentioned earlier, the IRMA dataset is classified into three classes: normal, benign lesion and malignant lesion. Initially, we used the IRMA dataset with all three classes. With the intent of improving the results, the dataset was then investigated using two classes. The binary version of the dataset was tested classifying between normal (without lesion) and with lesion (by taking benign and malignant lesions and count as one single class).

Here we present the results using IRMA as a multi-class dataset. We compare NEAT's performance to a Standard MLP. The MLP has been optimized using Grid-Search and Randomized Search. As mentioned in the Chapter 5, the parameter for the maximum number of hidden nodes used on the grid-search and the randomized search was taken from the highest number of nodes NEAT used on each dataset. Afterward, we compare NEAT's performance against an SVM and Random Forest algorithms also optimized by grid-search.

The Table 10 shows the results from NEAT, grid-search optimized MLP and randomized search optimized MLP. NEAT scored lower results than both MLPs when using the GLCM and LBP descriptor. On average NEAT scored 7.325% and 14.725% lower macro F-scores on GLCM and LBP respectively. NEAT showed standard deviations much lower than both MLPs. On Zernike's descriptor, NEAT achieved higher scores than both classifiers. Due to NEAT's nature, as it optimizes both topology and weights, in some instances, NEAT was able to overcome the optimal solution found by the grid-search algorithm as it was searching only through the number of hidden nodes. NEAT scored an F-score of 47.77% with 3.153 of standard deviation, 11.13% higher than the grid-search optimized MLP. Even though NEAT obtained similar accuracy and recall to the MLPs, it got a precision of 48.89% with 3.103 of standard deviation. This is 13.23% and 12.89% higher than grid-search MLP and randomized search MLP respectively.

The Table 11 presents the results comparing NEAT to SVM and Random Forest on multi-class IRMA. On this experiment, NEAT also scored lower on most metrics against both classifiers when on the GLCM and LBP descriptors. NEAT scored, on average, an F-score 7.06% and 11.735% lower than the other classifiers on GLCM and LBP respectively. On the other hand, NEAT overcame both classifiers on all metrics on Zernike's descriptor, with an F-score 6.24% higher than the SVM and 4.2% higher than the Random Forest.

Table 10 – Macro metrics from NEAT and MLP on IRMA multi-class

Descriptor	Classifier	F-Score Macro	Accuracy	Precision Macro	Recall Macro
GLCM	NEAT	59.39% (± 3.42)	60% (± 3.49)	59.81% (± 3.37)	60% (± 3.48)
	MLP GS	**66.79% (± 11.11)**	**67.67% (± 9.24)**	**68.67% (± 9.08)**	**67.67% (± 9.24)**
	MLP RS	66.64% (± 11.30)	67.55% (± 9.38)	68.55% (± 9.24)	67.55% (± 9.38)
LBP	NEAT	46.41% (± 3.10)	46.75% (± 2.92)	47.28% (± 3.43)	46.75% (± 2.92)
	MLP GS	**61.56% (± 8.28)**	**62.03% (± 6.74)**	**63.30% (± 7.22)**	**62.03% (± 6.74)**
	MLP RS	61.11% (± 8.58)	61.67% (± 6.94)	62.82% (± 7.59)	61.67% (± 6.94)
Zernike's	NEAT	**47.77% (± 3.15)**	**48.18% (± 3.46)**	**48.89% (± 3.10)**	**48.18% (± 3.45)**
	MLP GS	36.64% (± 8.76)	45.40% (± 8.21)	35.66% (± 9.61)	45.40% (± 8.21)
	MLP RS	36.43% (± 8.58)	45.60% (± 8.27)	36.00% (± 10.24)	45.60% (± 8.27)

GS grid search, *RS* randomized search, (±) standard deviation

Table 11 – Macro metrics from NEAT, SVM and Random Forest on IRMA multi-class

Descriptor	Classifier	F-Score Macro	Accuracy	Precision Macro	Recall Macro
GLCM	NEAT	59.39% (± 3.42)	60% (± 3.49)	59.81% (± 3.37)	60% (± 3.48)
	SVM	66.16% (± 10.50)	66.93% (± 8.71)	68.10% (± 8.59)	66.93% (± 8.71)
	RF	**66.74% (± 12.72)**	**68.06% (± 10.53)**	**69.22% (± 9.77)**	**68.06% (± 10.53)**
LBP	NEAT	46.41% (± 3.10)	46.75% (± 2.92)	47.28% (± 3.43)	46.75% (± 2.92)
	SVM	56.00% (± 6.23)	56.36% (± 5.17)	56.89% (± 5.94)	56.36% (± 5.17)
	RF	**60.29% (± 8.90)**	**61.09% (± 7.23)**	**62.22% (± 7.73)**	**61.09% (± 7.23)**
Zernike's	NEAT	**47.77% (± 3.15)**	**48.18% (± 3.46)**	**48.89% (± 3.10)**	**48.18% (± 3.45)**
	SVM	41.53% (± 4.18)	45.35% (± 5.12)	44.01% (± 5.19)	45.35% (± 5.12)
	RF	43.57% (± 4.67)	44.53% (± 4.00)	47.39% (± 5.94)	44.53% (± 4.00)

RF random forest, (±) standard deviation

In general, NEAT obtained poor scores when using multi-class IRMA using GLCM and LBP descriptors and good results using Zernike's descriptor compared to the other classifiers. All the classifiers obtained low accuracies on multi-class IRMA using any descriptor, with the highest accuracy being 68.06% with a standard deviation of 10.535 from the Random Forest tested using GLCM descriptor.

The first analysis using IRMA as a binary problem counted the instances labeled as benign and malignant as one (with lesion) and the remainder of the dataset as normal (without lesion) as mentioned in the Chapter 5. The class with lesion was used as the positive to calculate F-scores. Firstly, NEAT is compared against an MLP optimized by grid-search and randomized search, and then against an SVM an Random Forest. The same descriptors used in the previous analysis was used on binary IRMA.

The Table 12 present the results of the comparison between NEAT and the MLPs optimized by grid-search and random search. In general, NEAT performed slightly better than the MLP, except when using the LBP descriptor where NEAT got an F-score of

79.70% against 90.13% from the grid-search MLP. It is worth noticing that NEAT's scores when using the GLCM descriptor where very satisfactory, with an accuracy of 89.75% and F-score of 92.15%. NEAT achieved a precision of 94.07% and a recall of 90.40% using this descriptor, which means that it classified almost all instances with lesions correctly and almost all instances classified as positive were positive. Moreover, the general scores from all classifiers were better than when classifying IRMA into three classes, which shows that in this dataset it is easier to differentiate normal tissue from a lesioned tissue.

Table 12 – Metrics from NEAT and MLP on binary IRMA

Descriptor	Classifier	F-Score	Accuracy	Precision	Recall
GLCM	NEAT	**92.15% (± 1.76)**	**89.75% (± 2.27)**	94.07% (± 2.36)	**90.40% (± 2.88)**
	MLP GS	90.85% (± 10.59)	89.47% (± 11.27)	95.05% (± 1.121)	88.78% (± 17.48)
	MLP RS	90.78% (± 10.69)	89.43% (± 11.32)	**95.12% (± 1.07)**	88.66% (± 17.68)
LBP	NEAT	79.70% (± 3.30)	73.66% (± 3.46)	81.77% (± 3.152)	78.16% (± 6.45)
	MLP GS	**90.13% (± 8.87)**	**87.75% (± 9.83)**	**90.89% (± 2.75)**	**90.53% (± 14.82)**
	MLP RS	89.95% (± 9.11)	87.57% (± 10.07)	90.86% (± 2.98)	90.26% (± 15.14)
Zernike's	NEAT	**82.50% (± 2.70)**	**76.81% (± 3.16)**	**82.99% (± 3.14)**	82.36% (± 5.26)
	MLP GS	81.82% (± 3.26)	73.38% (± 3.92)	74.83% (± 1.36)	**90.40% (± 6.16)**
	MLP RS	81.68% (± 3.37)	73.21% (± 4.03)	74.77% (± 1.41)	90.16% (± 6.36)

GS grid search, *RS* randomized search, (±) standard deviation

The Table 13 shows the results from NEAT, SVM and Random Forest on the binary IRMA dataset. On this dataset, NEAT also performed slightly better than both classifiers using IRMA with GLCM and Zernike's descriptors. On GLCM, NEAT got an F-score 1.45% higher than the SVM and 1.78% higher than the Random Forest. Using this descriptor, the classifier SVM scored an F-score of 90.70% with 10.18 of standard deviation and the Random Forest scored an F-score of 90.37% with 12.59 of standard deviation. On the other hand, NEAT shows more consistent results with an F-score of 92.15% and only 1.76 of standard deviation. NEAT worst results were on LBP descriptor, where it scored an F-score 10.43% lower than SVM.

In general, NEAT's performance using multi-class IRMA was worse than the optimized MLPs, SVM, and Random Forest, except when using Zernike's descriptor. NEAT overcame all classifiers on binary IRMA, except when using LBP descriptor. NEAT's worst scores were when the features were extracted using LBP descriptor. The other classifiers worst scores were when using Zernike's descriptor. All classifiers (NEAT included) best scores were using GLCM descriptor.

Table 13 – Metrics from NEAT, SVM and Random Forest on binary IRMA

Descriptor	Classifier	F-Score	Accuracy	Precision	Recall
GLCM	NEAT	**92.15% (± 1.76)**	**89.75% (± 2.27)**	94.07% (± 2.36)	**90.40% (± 2.88)**
	SVM	90.70% (± 10.18)	89.16% (± 10.93)	94.43% (± 1.42)	88.95% (± 17.07)
	RF	90.37% (± 12.59)	89.56% (± 12.85)	**96.96% (± 0.84)**	87.12% (± 20.05)
LBP	NEAT	79.70% (± 3.30)	73.66% (± 3.46)	81.77% (± 3.15)	78.16% (± 6.45)
	SVM	87.08% (± 5.43)	82.37% (± 6.38)	83.70% (± 2.23)	**91.24% (± 9.89)**
	RF	**89.74% (± 10.64)**	**87.73% (± 11.41)**	**91.82% (± 2.57)**	89.27% (± 17.06)
Zernike's	NEAT	**82.50% (± 2.707)**	**76.81% (± 3.16)**	**82.99% (± 3.14)**	**82.36% (± 5.26)**
	SVM	78.71% (± 7.14)	71.32% (± 7.48)	76.65% (± 3.10)	81.49% (± 11.66)
	RF	78.19% (± 8.13)	71.75% (± 8.08)	78.74% (± 3.16)	78.43% (± 12.81)

RF random forest, (±) standard deviation

6.2 International Skin Imaging Collaboration Dataset

In this section, we analyze the performance of NEAT against all the other classifiers using ISIC dataset. As mentioned earlier, the ISIC dataset is labeled as a benign lesion or malignant lesion. We took the malignant class as the positive one, as it is more important to prevent false negatives on malignant lesions because it could prevent the patient from starting treatment on early stages of the disease. The first analysis was made comparing NEAT's scores against an MLP optimized using Grid-Search and Randomized Search. On the second analysis, NEAT's performance was compared against optimized SVM and Random forest classifiers.

The Table 14 presents the scores achieved by NEAT and a standard MLP. NEAT obtained similar accuracy and F-score than MLP when using the GLCM descriptor, with an F-score of 61.63% and standard deviation of 11.74 against grid search MLP's F-score of 61.67% and standard deviation of 23.27. NEAT also a similar F-score compared to randomized search MLP, NEAT's F-score was only 3.41% higher. When using LBP descriptor NEAT's F-score and accuracy were 6.29% and 7.05% inferior respectively against grid-search MLP. The observed metrics displayed high variance across the folds, possibly due to a very small number of sample per fold.

Table 14 – Metrics from NEAT and MLP on ISIC

Descriptor	Classifier	F-Score	Accuracy	Precision	Recall
GLCM	NEAT	61.63% (± 11.73)	61.50% (± 10.38)	61.91% (± 11.89)	63.70% (± 16.81)
	MLP GS	**61.67% (± 23.26)**	**63.8% (± 16.39)**	**62.93% (± 19.77)**	**67.5% (± 31.92)**
	MLP RS	58.22% (± 23.29)	61.20% (± 15.60)	59.78% (± 20.56)	62.70% (± 31.85)
LBP	NEAT	58.18% (± 13.44)	59.60% (± 10.93)	60.49% (± 12.540)	58.60% (± 18.40)
	MLP GS	**64.47% (± 19.5)**	**66.65% (± 15.48)**	**66.9% (± 16.93)**	**64.9% (± 23.79)**
	MLP RS	63.45% (± 20.29)	65.80% (± 15.92)	65.19% (± 17.34)	63.90% (± 24.47)

GS grid search, *RS* randomized search, (±) standard deviation

The Table 15 shows the performance obtained by each NEAT, SVM and Random Forest. NEAT's performance was better than the other classifiers on the GLCM descriptor. NEAT F-score was 5.67% higher than SVM's F-score and 4.77% higher than Random Forest's F-score. It can be noted that, even though Random Forest classifier achieved higher accuracy than NEAT's, NEAT got better precision and recall. This means that NEAT is better at identifying when there is a malignant lesion, than Random Forest. On LBP descriptor, NEAT performed worse than both classifiers. NEAT reached an F-score of 58.18% with standard deviation of 13.44 and accuracy of 59.60% with standard deviation of 10.93. The best classifier for this dataset using LBP descriptor was SVM, with an F-score of 63.53% and standard deviation of 16.15 and accuracy of 65.15% with standard deviation of 12.69. SVM achieved an F-score 5.35% higher than NEAT's F-score.

Table 15 – Metrics from NEAT, SVM and Random Forest on ISIC

Descriptor	Classifier	F-Score	Accuracy	Precision	Recall
GLCM	NEAT	**61.63% (± 11.73)**	61.5% (± 10.38)	**61.91% (± 11.89)**	**63.7% (± 16.81)**
	SVM	55.96% (± 21.26)	60.30% (± 12.51)	59.74% (± 17.76)	58.40% (± 29.23)
	RF	56.86% (± 24.65)	**62.00% (± 15.03)**	59.56% (± 17.47)	61.00% (± 34.48)
LBP	NEAT	58.18% (± 13.44)	59.60% (± 10.93)	60.49% (± 12.54)	58.60% (± 18.40)
	SVM	**63.53% (± 16.15)**	**65.15% (± 12.69)**	64.94% (± 13.00)	**64.3% (± 20.43)**
	RF	62.62% (± 14.18)	64.00% (± 11.13)	**64.95% (± 11.34)**	64.00% (± 20.59)

RF random forest, (±) standard deviation

The best descriptor for this dataset was the grid-search optimized MLP, using LBP descriptor. This classifier achieved 64.47% of F-score with 19.5 of standard deviation. In general, all classifiers performed poorly on this dataset.

6.3 Wisconsin Diagnosis Breast Cancer Dataset

In this section, it is presented NEAT's accuracy compared to other NeuroEvolution algorithms found in the literature using the Wisconsin Diagnosis Breast Cancer dataset. As mentioned earlier, the WDBC has 569 instances, and it is labeled as a benign lesion and malignant lesion.

The Table 16 presents the results from this experiment. The comparison between the algorithms is going to be regarding accuracy, as this is the only metric in common between this papers. As shown in the Table 16, NEAT achieved slightly higher accuracies than all the algorithms, except (MANNING; WALSH, 2013)'s CGPANN-RBF, which achieved an accuracy 0.95% higher than NEAT's. NEAT reached an accuracy of 96.24% with a standard deviation of 2.393%, which was 2.66% higher than (BELCIUG; GORUNESCU, 2013)'s Hybrid MLP/GA and 0.24% higher than (AHMAD et al., 2012)'s CGPANN. Although all

the results were very close from one another, it should be noted that NEAT converged much faster. While (AHMAD et al., 2012)'s CGPANN and (MANNING; WALSH, 2013)'s CGPANN-RBF executed 100,000 generations, NEAT executed only 1,000 generations. On the other hand, (BELCIUG; GORUNESCU, 2013)'s Hybrid MLP/GA only executed 100 generations with a population size of 100 organisms.

Table 16 – Comparing results from NEAT with related work using the WDBC dataset

Algorithm	Accuracy	Reference
NEAT	96.24% (± 2.393)	-
Hybrid MLP/GA	93.58%	(BELCIUG; GORUNESCU, 2013)
CGPANN-RBF *crossover*	**97.19%**	(MANNING; WALSH, 2013)
CGPANN	96%	(AHMAD et al., 2012)

6.4 Wisconsin Breast Cancer Dataset

This section presents the results regarding the Wisconsin Breast Cancer Dataset. This dataset was chosen to compare NEAT with other algorithms found in the literature. In this study, NEAT was compared against three neuroevolution algorithms: Genetically Optimized Neural Network (GONN) algorithm by (BHARDWAJ; TIWARI, 2015); Genetic Algorithm Artificial Neural Network Resilient Back-Propagation (GAANN RP) by (AHMAD et al., 2015); and Intelligent Multi-Objective classifier (IMOC) by (IBRAHIM et al., 2015). The comparations were made regarding accuracy due to the fact that this is the only metric in common between all papers. The Table 17 presents the results of each algorithm. NEAT achieved an accuracy very similar to all the other algorithms, with 97.44% and standard deviation of 1.785. The algorithm with the highest accuracy was GONN by (BHARDWAJ; TIWARI, 2015), which had an accuracy 1.82% higher than NEAT's accuracy. GONN reached an accuracy of 99.26% with 0.602 of standard deviation. The algorithm proposed by (BHARDWAJ; TIWARI, 2015) uses genetic algorithm to select features to pass to the neural network. It is possible that NEAT could have achieved better results using the same technique.

Table 17 – Comparing results from NEAT with related work using the WBCD dataset

Algorithm	Accuracy	Reference
NEAT	97.44% (± 1.785)	-
GONN	**99.26 % (± 0.602)**	(BHARDWAJ; TIWARI, 2015)
GAANN RP	98.29 (± 0.8)	(AHMAD et al., 2015)
IMOC	97.51 (± 1.69)	(IBRAHIM et al., 2015)

7 Conclusion and Future Work

This work has presented an analysis of the performance of the NeuroEvolution of Augmenting Topologies applied in the context of detecting tumors in medical images. In this study, as NEAT evolves Artificial Neural Networks (ANN), it was compared to a Multilayer-Perceptron optimized by a Grid-Search and a Randomized Search to assess NEAT's ability to optimize an ANN when compared to the optimal solution and a solution found by a randomized search. Another study was made comparing NEAT to popular learning algorithms in classification problems. Those algorithms were the Support Vector Machines (SVM), and the Random Forest. Those algorithms were also optimized for each dataset tested in this study using the grid-search technique. Moreover, NEAT was compared against six neuroevolution algorithms found in the literature, being them: MLP/GA Hibride, proposed by (BELCIUG; GORUNESCU, 2013), CGPANN-RBF crossover proposed by (MANNING; WALSH, 2013), CGPANN, proposed by (AHMAD et al., 2012), GONN, proposed by (BHARDWAJ; TIWARI, 2015), GAANN RP, proposed by (AHMAD et al., 2015), and Intelligent Multi-Objective classifier (IMOC), proposed by (IBRAHIM et al., 2015).

In general terms, NEAT presented promising results, when compared to the algorithms state of the art. NEAT overcame two of those classifiers Hybrid MLP/GA and CGPANN while reaching results very close to the others. NEAT also achieved good results when using the binary IRMA with all descriptors, overcoming all classifiers, except when using LBP descriptor. NEAT achieved lowest scores were using the LBP descriptor, and its highest scores were using the GLCM descriptor. The GLCM descriptor produced the best results for all classifiers combined with all image datasets.

7.1 Contributions

The main contributions of this work the experimental study of the algorithm NEAT, which have not been thoroughly in the context of tumor detection in medical images. The study compared NEAT against commonly used classifiers and neuroevolution algorithm state of the art found in the literature. In contrast to other works in the area, this project investigates NEAT using a variety of datasets and larger databases. Another contribution is the paper in (FRANÇA; MIRANDA; CORDEIRO, 2017) presenting preliminary results for this work published on the XIV Encontro Nacional de Inteligência Artificial e Computacional.

7.2 Future Works

During the experiments, new questions arrived that could be assessed in further studies. Although some preliminary results showed that F-score macro produced good results with NEAT, one analysis would be regarding other possible fitness function, such as accuracy, mean squared error, among others. It would be interesting to make a hypothesis test to analyze the statistical difference between the results.

In several studies, the feature selection was performed aiming to improve the performance of the classifiers. The feature selection could be experimented alongside with the combination of descriptors.

Also, as deep learning has shown great results in many areas in the literature, it would be interesting to compare NEAT with a Deep learning algorithm, and also to compare Deepneat (a version of neat that evolves deep learning algorithms) against a regular deep learning.

Another work would be to analyze NEAT to check how fast it converges, the history of the species generated, the topologies generated, compared with the optimal topology found by the grid-search on an MLP, and the decrease of error.

Finally, a study could be made regarding the optimization of NEATs hyperparameters. As shown in the Chapter 5, NEAT has many hyperparameters, that can influence directly on its performance. With these parameters optimized, NEAT could achieve even greater performances than those shown in this work.

Bibliography

AHMAD, A. M. et al. Breast cancer detection using cartesian genetic programming evolved artificial neural networks. p. 1031–1038, 2012. 24, 25, 26, 32, 45, 46, 47

AHMAD, F. et al. A ga-based feature selection and parameter optimization of an ann in diagnosing breast cancer. *Pattern Analysis and Applications*, Springer, v. 18, n. 4, p. 861–870, 2015. 24, 26, 32, 46, 47

AWAD, A. I.; HASSABALLAH, M. Image feature detectors and descriptors. Springer, v. 630, 2016. 22

BELCIUG, S.; GORUNESCU, F. A hybrid neural network/genetic algorithm applied to breast cancer detection and recurrence. *Expert Systems*, Wiley Online Library, v. 30, n. 3, p. 243–254, 2013. 14, 24, 26, 32, 45, 46, 47

BELL, J. *Machine learning: hands-on for developers and technical professionals*. [S.l.]: John Wiley & Sons, 2014. ISBN 978-1-118-88906-0. 19

BENNETT, K. P.; MANGASARIAN, O. L. Robust linear programming discrimination of two linearly inseparable sets. *Optimization methods and software*, Taylor & Francis, v. 1, n. 1, p. 23–34, 1992. 36

BHARDWAJ, A.; TIWARI, A. Breast cancer diagnosis using genetically optimized neural network model. *Expert Systems with Applications*, Elsevier, v. 42, n. 10, p. 4611–4620, 2015. 24, 26, 32, 46, 47

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: ACM. *Proceedings of the fifth annual workshop on Computational learning theory*. [S.l.], 1992. p. 144–152. 19

DAWSON, C.; WILBY, R. Hydrological modelling using artificial neural networks. *Progress in physical Geography*, Sage Publications Sage CA: Thousand Oaks, CA, v. 25, n. 1, p. 80–108, 2001. 33

DESERNO, T. M. et al. Computer-aided diagnostics of screening mammography using content-based image retrieval. In: *Proceedings of Society of Photo-Optical Instrumentation Engineers (SPIE)*. [S.l.: s.n.], 2012. v. 8315, p. 831527–831527. 34

DHEEBA, J.; SINGH, N. A.; SELVI, S. T. Computer-aided detection of breast cancer on mammograms: A swarm intelligence optimized wavelet neural network approach. *Journal of biomedical informatics*, Elsevier, v. 49, p. 45–52, 2014. 14, 23, 24, 26

EIBEN, A. E.; SCHOENAUER, M. Evolutionary computing. *Information Processing Letters*, Elsevier, v. 82, n. 1, p. 1–6, 2002. 16

EIBEN, A. E.; SMITH, J. E. et al. *Introduction to evolutionary computing*. 2. ed. [S.l.]: Springer-Verlag Berlin Heidelberg, 2015. (Natural Computing Series). ISBN 978-3-662-44873-1. 16

FAN ZHENHUA WANG, F. W. a. B. *Local Image Descriptor: Modern Approaches*. 1. ed. [S.l.]: Springer-Verlag Berlin Heidelberg, 2015. (SpringerBriefs in Computer Science). ISBN 978-3-662-49171-3. 21

FRANÇA, L. D.; MIRANDA, P. B.; CORDEIRO, F. R. Um Estudo Experimental da Aplicação do Algoritmo de Neuroevolução com Crescimento Topológico em Imagens Médicas. *XIV Encontro Nacional de Inteligência Artificial e Computacional*, p. 599–608, 2017. 47

GRAÑA, M. *Advances in knowledge-based and intelligent information and engineering systems*. [S.l.]: IOS press, 2012. v. 1. ISBN 978-1-61499-104-5. 21

GRISCI, B.; DORN, M. Predicting protein structural features with neuroevolution of augmenting topologies. In: IEEE. *Neural Networks (IJCNN), 2016 International Joint Conference on*. [S.l.], 2016. p. 873–880. 14, 26

GRUAU, F. Genetic synthesis of modular neural networks. In: MORGAN KAUFMANN PUBLISHERS INC. *Proceedings of the 5th International Conference on Genetic Algorithms*. [S.l.], 1993. p. 318–325. 18

HARALICK, R. M.; SHANMUGAM, K. et al. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, Ieee, n. 6, p. 610–621, 1973. 21, 36

HE, D.-C.; WANG, L. Texture unit, texture spectrum, and texture analysis. *IEEE transactions on Geoscience and Remote Sensing*, IEEE, v. 28, n. 4, p. 509–512, 1990. 21, 36

HOLLAND, J. H. Adaptation in natural and artificial systems. an introductory analysis with application to biology, control, and artificial intelligence. *Ann Arbor, MI: University of Michigan Press*, p. 439–444, 1975. 16

HSE, H.; NEWTON, A. R. Sketched symbol recognition using zernike moments. In: IEEE. *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. [S.l.], 2004. v. 1, p. 367–370. 22

HSU, C.; CHANG, C.; LIN, C. *Scikit-Learn: A Toolkit of Machine Learning in Python*. Available at: <http://scikit-learn.org>. 39

IBRAHIM, A. O. et al. Intelligent multi-objective classifier for breast cancer diagnosis based on multilayer perceptron neural network and differential evolution. p. 422–427, 2015. 14, 23, 26, 32, 46, 47

JONES, E. et al. *SciPy: Open source scientific tools for Python*. 2001. [Online; accessed <today>]. Disponível em: <http://www.scipy.org/>. 39

KHAN, M. M. et al. Evolving wavelet neural networks for breast cancer classification. 2014. 25, 26

KLETTE, R. *Concise computer vision: An Introduction into Theory and Algorithms*. [S.l.]: Springer, 2014. ISBN 978-1-4471-6319-0. 21

MANGASARIAN, O. L. Cancer diagnosis via linear programming. *SIAM news*, v. 23, n. 5, p. 18, 1990. 36

MANGASARIAN, O. L.; SETIONO, R.; WOLBERG, W. Pattern recognition via linear programming: Theory and application to medical diagnosis. *Large-scale numerical optimization*, p. 22–31, 1990. 36

MANNING, T.; WALSH, P. Improving the performance of cgpann for breast cancer diagnosis using crossover and radial basis functions. p. 165–176, 2013. 25, 26, 32, 45, 46, 47

MCKINNEY, W. Data structures for statistical computing in python. In: WALT, S. van der; MILLMAN, J. (Ed.). *Proceedings of the 9th Python in Science Conference*. [S.l.: s.n.], 2010. p. 51 – 56. 39

NICOLAS, P. R. *Scala for machine learning*. [S.l.]: Packt Publishing Ltd, 2015. 19, 37, 38

OLIVEIRAA, J. E. et al. Towards a standard reference database for computer-aided mammography. v. 6915, p. 69151Y, 2008. 34

PAL, M. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, Taylor & Francis, v. 26, n. 1, p. 217–222, 2005. 20

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. 32, 39

RUSSELL, S. J.; NORVIG, P. *Artificial intelligence: a modern approach*. [S.l.]: Prentice hall Upper Saddle River, 2010. v. 3. ISBN 978-0-13-604259-4. 19

STANLEY, K. et al. Neuroevolution of an automobile crash warning system. In: ACM. *Proceedings of the 7th annual conference on Genetic and evolutionary computation*. [S.l.], 2005. p. 1977–1984. 14, 26

STANLEY, K. O.; MIIKKULAINEN, R. Evolving neural networks through augmenting topologies. *Evolutionary computation*, MIT Press, v. 10, n. 2, p. 99–127, 2002. 17, 18, 26, 28, 29, 30, 31, 38, 40

SUTHAHARAN, S. *Machine learning models and algorithms for big data classification*. [S.l.]: Springer, 2016. v. 36. ISBN 978-1-4899-7640-6. 20

SYLENIUS. *Wikimedia Commons. Optimal hyperplane with margins, to illustrate SVM method*. 2016. Last accessed: January 30th, 2018. Available at: <https://commons.wikimedia.org/wiki/File:Separatrice_lineaire_avec_marges.svg>. 20

TAN, M.; PU, J.; ZHENG, B. A new and fast image feature selection method for developing an optimal mammographic mass detection scheme. *Medical physics*, Wiley Online Library, v. 41, n. 8Part1, 2014. 14, 26

TURABIEH, H. Comparison of NEAT and Backpropagation Neural Network on Breast Cancer Diagnosis. *International Journal of Computer Applications*, v. 139, n. 8, p. 40–44, 2016. 14, 25, 26

WALT, S. v. d.; COLBERT, S. C.; VAROQUAUX, G. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, IEEE, v. 13, n. 2, p. 22–30, 2011. 39

WOLBER, D. H.; STREET, W. N.; MANGASARIAN, O. L. *Wisconsin Diagnostic Breast Cancer (WDBC)*. 1995. Data retrieved from University of California Irvine Repositories, <https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)>. 35

WOLBERG, W. H.; MANGASARIAN, O. L. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the national academy of sciences*, National Acad Sciences, v. 87, n. 23, p. 9193–9196, 1990. 36

ZERNIKE, v. F. Beugungstheorie des schneidenver-fahrens und seiner verbesserten form, der phasenkontrastmethode. *Physica*, Elsevier, v. 1, n. 7-12, p. 689–704, 1934. 22

ZOHRA, B. F.; NACÉRA, B. Detection of tumor in mammographic images by rbf neural network and multi population genetic algorithm. *International Journal of Applied Information Systems (IJAIS)*, v. 6, n. 3, 2013. 24, 26