Machine Learning Applied to Indoor Hydroponic Lettuce Optimization

Miguel Afonso Beckers - 57403

Thesis presented to the School of Technology and Management in the scope of the Master in Informatics.

Supervisors:
Prof. Dr. Arnaldo Candido Junior
Prof. Dr. Pedro João Soares Rodrigues
Prof. Dr. Pedro Luiz de Paula Filho

This document does not include the suggestions made by the board.

Bragança
2023-2024

Machine Learning Applied to Indoor Hydroponic Lettuce Optimization

Miguel Afonso Beckers

Dissertation presented to the School of Technology and Management of Bragança to obtain the master's degree in Informatics within the scope of the double degree program with the Federal University of Technology – Paraná

Supervisors:

Prof. Dr. Arnaldo Candido Junior

Prof. Dr. Pedro João Soares Rodrigues

Prof. Dr. Pedro Luiz de Paula Filho

Bragança

June 2024

Acknowledgment

I want to express my deepest gratitude to my supervisors, Prof. Dr. Arnaldo Candido Junior, Prof. Dr. Pedro João Soares Rodrigues, and Prof. Dr. Pedro Luiz de Paula Filho, for their invaluable guidance, patience, and support. I am also grateful to Prof. Dr. Glauco Vieira Miranda and his team, whose collaboration in carrying out the lettuce cultivation, data collection, and advice were exceptional. My appreciation extends to the *Universidade Tecnológica Federal do Paraná* (UTFPR) and the *Instituto Politécnico de Bragança* (IPB) for allowing me to participate in the double degree program in which this work was developed. I must also thank my family, especially my parents, Lidio Beckers and Rosalha Beckers, and my brother Carlos Alberto Beckers, for their support and encouragement during my education. My friends who accompanied me on this journey deserve special mention, in particular my friend Daniel Augusto Rodrigues Farina, whose collaboration was significant for the development of this work. Thank you all.

Abstract

Optimizing indoor hydroponic lettuce production is important for enhancing its economic viability. This study examined machine learning approaches for its optimization. Among the options analyzed, an artificial neural network (ANN) combined with a brute-force algorithm was chosen. This choice proved compatible with the available time and materials. The gain in fresh matter mass was defined as the optimization goal. And, the irrigation time off, the electric conductivity, and the pH of the nutrient solution were defined as the optimization parameters. The employed method involved four steps: conducting exploratory plantings to generate a database; training an ANN to create a regression model; using the brute-force algorithm to find the optimal combination in the generated model; and conducting validation plantings to verify if the found combination produces the predicted value. The exploratory plantings generated results with variations. The variations were normalized and additional lighting data were collected. The optimization result predicted that the found combinations would generate an average normalized production of 114.22g per individual. Two validation plantings were conducted: the first exceeded the prediction by 34.43%, while the second was 17.20% below the expected. The first result surpassed the values of the exploratory plantings, but the second did not. Due to this divergence, further studies will be necessary to reinforce the validation of the results.

Keywords: Modified hydroponic shipping container, controlled-environment agriculture, artificial neural networks, brute-force algorithm.

Resumo

A otimização da produção de alface hidropónica em ambiente fechado é importante para ampliar a sua viabilidade económica. Este trabalho analisou as abordagens de aprendizagem automática para a realização da sua otimização. Entre as opções analisadas, optou-se pelo uso de uma rede neuronal artificial (RNA) combinada com um algoritmo de força-bruta. Esta opção revelou-se compatível com a disponibilidade de tempo e de materiais. O ganho de massa da matéria fresca foi definido como sendo o objetivo da otimização. E, o tempo de irrigação desligada, a condutividade elétrica, e o pH da solução nutritiva foram definidos como sendo os parâmetros da otimização. O método empregado envolveu quatro etapas: a realização de plantios exploratórios para a geração de uma base de dados; o treino de uma RNA para a geração de um modelo de regressão; o uso do algoritmo de força-bruta para encontrar a combinação ótima no modelo gerado; e a realização de plantios de validação para verificar se a combinação encontrada produz o valor previsto. Os plantios exploratórios geraram resultados com variações. As variações foram normalizadas e dados adicionais de iluminação foram coletados. O resultado da otimização previu que as combinações encontradas gerariam uma produção média normalizada de 114,22g por indivíduo. Dois plantios de validação foram realizados: o primeiro superou a previsão em 34,43%, enquanto o segundo ficou 17,20% abaixo do esperado. O primeiro resultado superou os valores dos plantios exploratórios, mas o segundo não. Devido a essa divergência, mais estudos serão necessários para reforçar a validação dos resultados.

Palavras-chave: Contêiner hidropónico modificado, agricultura em ambiente controlado, redes neuronais artificiais, algoritmo de força-bruta.

Contents

List of Tables

List of Figures

Acronyms

ACO Ant Colony Optimization.

AF Air Flow.

AIS Artificial Immunological System.

ANN Artificial Neural Network.

AT Ambient Temperature.

B Blue.

Ca Calcium.

CEA Controlled-Environment Agriculture.

CO_2 Carbon dioxide.

DMM Dry Matter Mass.

DO Dissolved Oxygen.

EC Electric Conductivity.

FMM Fresh Matter Mass.

G Green.

GA Genetic Algorithm.

GELU Gaussian Error Linear Unit.

GPU Graphics Processing Unit.

H^+ Hydrogen ions.

IF Indoor Farming.

IPB Polytechnic Institute of Bragança.

IT-Off Irrigation Time Off.

IT-On Irrigation Time On.

K Potassium.

LED Light-Emitting Diode.

MAE Mean Absolute Error.

Mg Magnesium.

MHSC Modified Hydroponic Shipping Container.

ML Machine Learning.

MSE Mean Square Error.

N Nitrogen.

NFT Nutrient Film Technique.

O Oxygen.

OH^- Hydroxide ions.

P Phosphor.

PAR Photosynthetic Active Radiation.

pH Hydrogen potential.

PPFD Photosynthetic Photon Flux Density.

ppm parts per million.

PSO Particle Swarm Optimization.

R Red.

RB Red-Blue.

RGB Red-Green-Blue.

RH Relative Humidity.

RMSE Root Mean Square Error.

S Sulfur.

SA Simulated Annealing.

SD Standard Deviation.

ST Solution Temperature.

TLL/SD Total Leaf Length to Stem Diameter.

TPU Tensor Processing Unit.

UTFPR *Federal University of Technology - Paraná.*

UV Ultraviolet.

Chapter 1

Introduction

Lettuce is a widely consumed food[1], however, its soil production presents some difficulties such as dependence on suitable climate, pest attacks, and distance from consumption sites[2]. To address these issues, some techniques like hydroponics and indoor cultivation have been developed[3]. The combination of these two methods offers numerous advantages, including residue reduction, protection against pests, and the possibility of cultivation in locations close to consumption points. However, there are drawbacks, with the main one being the higher production cost due to high light consumption[3], [4]. It is within this context that the need arises to optimize production, to maximize resource utilization and ensure viability.

Optimizing a problem like this requires identifying the combination of parameters that maximizes production. One approach to find an optimal result is to perform a grid planting, where the quantity of each parameter is gradually varied to find the best combination among them. Thus, the smaller the variation, the closer to the real optimal result it is possible to get, but more plants will be needed to be planted to perform the process. Another approach is to use Machine Learning (ML) techniques. These techniques can achieve a similar result in a shorter time with fewer plantings[5]. In this sense, it is possible to optimize using evolutionary algorithms[5], or by combining Artificial Neural Network (ANN) and an optimization algorithm[6], [7].

There are several evolutionary algorithms, such as Genetic Algorithm (GA) and Artificial Immunological System (AIS)[5]. These have the characteristic of evolving through generations, and in each one, the algorithm evolves until the result converges to an optimum value. Although they evolve quickly, it may still require several generations for convergence[5]. In this case, one question arises: which one would perform better in a specific scenario? The other strategy consists in collecting data to build a database, training an ANN to generate a regression model, and finding the optimal value of the model using an optimization algorithm. This technique has the advantage of not being iterative, but it also may require a significant amount of data for the network to generate a precise model[6], [7].

1.1 Objectives

The primary goal of this study is: to apply a ML technique for optimizing indoor hydroponic lettuce production. In order to accomplish this, it were set several specific objectives:

- Establish the optimization goal and parameters;
- Choose the most suitable ML technique to find the optimal parameters;
- Implement a computational system for the chosen ML technique;
- Define the planting strategy and procedures;
- Perform exploratory and validation plantings;

1.2 Document Organization

This document is organized as follows: chapter 2 presents the theoretical background and the related works; chapter 3 describes the used materials, the decisions made, and the used methods; chapter 4 presents discuss the obtained results; and chapter 5 presents the conclusions and future works.

Chapter 2

Literature Review

This chapter aims to provide the reader with the necessary context to adequately understand the concepts involved in this work. When studying the use of ML in optimizing indoor hydroponic lettuce, it is necessary to establish what each part of this study entails. In this regard, the first and second sections explain lettuce and its cultivation in an indoor hydroponic system, presenting the characteristics of the plant and techniques associated with its cultivation in this system. The third and fourth sections address issues involving ML, initially discussing optimization and the main techniques used in this context, and subsequently discussing concepts related to ANN, as the work involves their use. Finally, the fifth section presents related works to situate the reader regarding what has already been done in this context.

2.1 Controlled-Environment Agriculture

The Controlled-Environment Agriculture (CEA) is a concept that is becoming highly popular for cultivating plants in urban centers, involving the cultivation of crops grown indoors in controlled spaces that often utilize hydroponic growth methods. CEA methods involve five crucial[8] factors of plant development: light, temperature, water, nutrients, and atmosphere. CEA has been promoted as a sustainable alternative to conventional agriculture system, as it can produce food with significantly lower water requirements[3].

Although initially indoor crop production did not seem commercially feasible, CEA started as a benefit for researchers. Through the use of various techniques, they can systematically investigate the effects of specific factors on crop growth isolated from uncontrolled external conditions that would confound or change experimental findings. However, these studies enabled the manipulation of plants in ways that were never seen before[4].

2.1.1 Hydroponics

The term hydroponics derives from the Greek words: *hidro*, which means water; and *ponia*, which means labor. Also known as soilless cultivation, this technique involves cultivation in which soil is replaced by an aqueous solution. This solution contains only the essential mineral elements for plants[9].

Although current hydroponic systems are sophisticated, their development was not rapid. There are records that its first use may have been in the cultivation of cucumbers out of season for the Roman Emperor Tiberius during the 1st century, and it is believed that the technology was little used, if at all, in the following 1500 years[10].

The major change came with Hoagland and Broyer (1936) who formulated the nutrient solution that, with modifications, is widely used today. Hoagland and Broyer (1983)[11] are considered major pioneers of the modern period of plant nutrition, and their book "Principles of Plant Nutrition" (1994) is considered a classic[12].

In the 1960s, the Nutrient Film Technique (NFT) was developed in Littlehampton, England. As can be seen in Figure 2.1, this technique involves a system composed of pipes that support the plants, a hydraulic system consisting of a reservoir of nutrient solution, distribution piping, a water pump, and a timer. The solution irrigates the roots continuously or intermittently at a predefined time according to the needs of the crop[9].

It is important to note that the pipes are commonly arranged side by side in a horizontal orientation, as can be seen in Figure 2.1. This arrangement has the advantage of allowing greater utilization of sunlight by the plants[9].

Another possibility for the use of the NFT system is the installation in a vertical

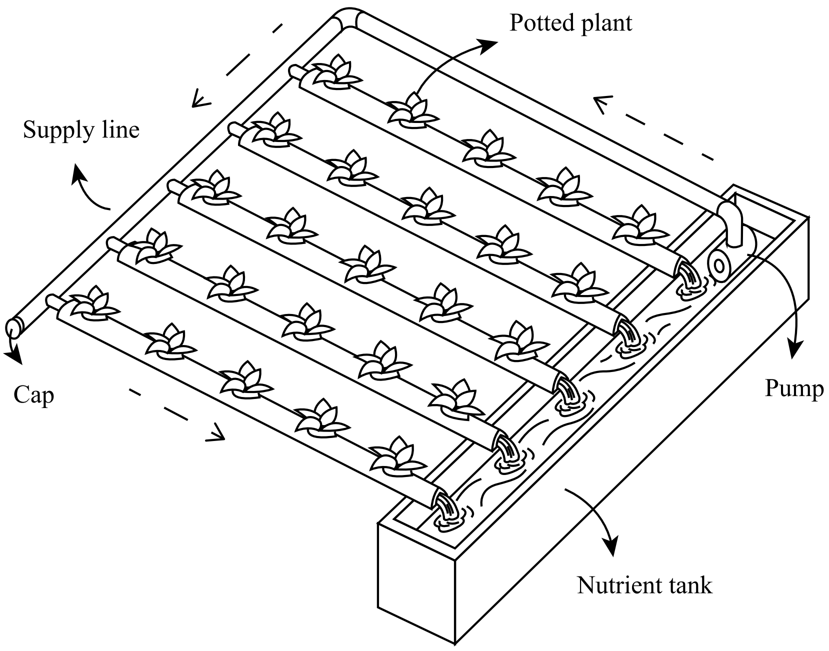


Figure 2.1: Nutrient film technique. Source: adapted from Resh (2013)[9]

orientation. In certain cases where space for cultivation is limited, the system has been adapted to function vertically in order to enable greater cultivation in this condition. In this case, the solution is pumped into the top pipe and then sequentially flows through all the tubes until it returns to the reservoir[9]. Although, it is also possible to stack horizontal systems, in this case, artificial lights are used between the layers.

2.1.2 Indoor Farming

The technique of Indoor Farming (IF) involves growing vegetables in a controlled environment using a structure to shield the crops, in combination with hydroponics. This activity is highly technological, productive, and economical, with environmental control being one of the main concerns in agricultural work. The production is generally carried out inside a space designed to control air and root temperature, light, water, and nutrition, providing plants with the ideal conditions for growth and maximizing yield potential[10].

Favorable climatic conditions are necessary to obtain a high-quality crop. In conventional farming systems, producers typically rely on selecting the most suitable region or season for crop cultivation, leading to localized production or seasonality in production, impacting supply. Therefore, the adoption of protected cultivation becomes necessary as it allows for cultivation in diverse locations and over extended periods of time[2].

The concern for the physical protection of crops against climatic adversities is ancient, with the earliest records dating back to the Roman Empire period. However, it was only after the Second World War that there was a significant expansion in the area covered by glass greenhouses, especially in the Netherlands. In the 1950s, the United States of America and some countries in Europe became the dividing line for a subsequent period of significant expansion in protected cultivation[13]. Today, indoor cultivation practices have extended beyond traditional rural settings and are increasingly adopted in urban spaces as well, marking a new chapter in the evolution of agriculture[3].

Embedded within urban architecture, one particular form among the various methods of IF is constituted by hydroponic production inside the shipping containers. These units, known as Modified Hydroponic Shipping Container (MHSC), can produce a large volume of plants and can be easily transported and operated. When installed near consumption points, MHSC units reduce the need for packaging, storage, and food transportation, significantly reducing residue and energy consumption[3].

2.2 Hydroponic Lettuce (*Lactuca Sativa L.*)

The *Lactuca sativa*, commonly known as lettuce, is a member of the *Asteraceae* family. It is one of the most consumed vegetables[1]. In Brazil, lettuce is a prominent vegetable in terms of production, commercialization, and nutritional value. The increasing consumption demands producers to maintain a consistent quality and regularity in production[12]. In Portugal, the hydroponic lettuce has proven to be a viable option for producing a higher quality product, and it has been well-received by consumers, especially by the younger people[14].

Lettuce production in a hydroponic system requires attention to its nutrition as well as control of other variables. The characteristics of the plant, the care to be taken for its cultivation, and the way to evaluate the quality of the product are defined below.

2.2.1 Parameters

Nutrients are divided into two groups according to the amount absorbed by the plant. Those absorbed in abundance are macronutrients: Nitrogen (N), Phosphor (P), Potassium (K), Calcium (Ca), Magnesium (Mg), and Sulfur (S). And those absorbed in small quantities are micronutrients: Boron, Chlorine, Copper, Iron, Manganese, Molybdenum, and Zinc[12].

N is the most influential nutrient in vegetative growth. However, its excess or imbalance with other macros or micros nutrients can be detrimental[15]. P is important for metabolism, thus related to nutrient absorption, photosynthesis, and respiration processes, and also contributes to root system growth. K influences leaf quality, but its excess can hinder Mg absorption. In turn, Mg is a constituent of chlorophyll molecules and is therefore necessary for photosynthesis. Finally, Ca is essential for the structure and functioning of cell membranes. Rapid plant growth without sufficient absorption of this nutrient can lead to a physiological disorder called tip burn[12].

It is important to note that these nutrients should be diluted separately to avoid the precipitation of insoluble Ca sulfate and Ca phosphate. According to Brechner and A. J. Both (2013)[16], the quantities of each macronutrient indicated for the composition of the nutrient solution are shown in Table 2.1.

Table 2.1: Indicated nutrient quantities. Source: Brechner (2013)[16]

Nutrient	**Quantity**
N	4.50%
P	0.76%
K	5.40%
Ca	1.39%
Mg	0.31%

Temperature controls plant growth by influencing chemical processes. Most chemical processes in a plant are regulated by enzymes that perform optimally within certain temperature ranges. Below and above these temperature ranges enzyme activity begins to deteriorate, leading to a chemical process result where the activity may decrease or even stop. The Ambient Temperature (AT) should be kept between 19 ℃ (night) and 24 ℃ (day), while Solution Temperature (ST) should be kept between 24 ℃ and 26 ℃[16].

The Relative Humidity (RH) of the air influences the rate of transpiration in plants. High RH leads to lower transpiration and reduces nutrient transport from roots to leaves, as well as less cooling of leaf surfaces. High RH can also cause problems with diseases in some cases, such as the growth of fungi and mold. For healthy development, it is recommended to maintain RH between 50% and 70%[16].

The concentration of Carbon dioxide (CO_2) directly influences plant photosynthesis and, consequently, its growth. In the external environment, the concentration of CO_2 is around 390 parts per million (ppm). Plants in a closed environment on a sunny day can deplete the concentration of CO_2 by 100 ppm, severely reducing photosynthesis. In closed environments, increasing the CO_2 concentration to 1000 to 1500 ppm accelerates growth[16].

Light is essential for plants, as they depend on it to carry out photosynthesis. Its measurement should be done using a quantum sensor that measures Photosynthetic Active Radiation (PAR) in mol/m^2/d. PAR is the light that is useful for plants to perform photosynthesis. Measuring PAR provides an indicator of the potential amount of plant growth through photosynthesis. An optimal PAR is indicated between 10 mol/m^2/d and 20 mol/m^2/d, with a value of 17 mol/m^2/d considered optimal[16]. Furthermore, the color of the light can influence the development of the plant, with a 3:1 ratio of Red-Blue (RB) commonly used[17].

The Air Flow (AF) is used to aid in plant transpiration, increasing nutrient transport, especially Ca from the roots for rapid growth of new leaves. This accelerated transport helps prevent tip burn[16]. The suggested maximum and minimum values are 0.1 m^3/min/m^2 and 0.7 m^3/min/m^2 respectively.

The measurement of Dissolved Oxygen (DO) indicates the amount of Oxygen (O) available in the nutrient solution for root respiration. Lettuce will grow satisfactorily at a DO level of 4 ppm. If O is not added to the nutrient solution, DO levels will drop to 0 ppm. O deprivation will halt the respiration process, causing serious damage and killing the plant. Pure O is added to the system directly to the tanks. Typically, the level to be maintained is 7 to 10 ppm[16].

The acidity or Hydrogen potential (pH) of a solution is a measure of the concentration of Hydrogen ions (H^+). The pH of a solution can range from 0 to 14. A neutral solution has a pH of 7, meaning there is an equal number of H^+ and Hydroxide ions (OH^-) ions. Solutions with pH levels between 0 and 6.9 are considered acidic and have a higher degree of H^+ ions. Solutions with pH levels between 7.1 and 14 are basic or alkaline with a higher concentration of OH^- ions. The pH is important because it controls the availability of fertilizer salts. A pH of 5.8 is considered optimal for lettuce growth systems; however, a range of 5.6 to 6 is completely acceptable[16].

The Electric Conductivity (EC) is a measure of the salts dissolved in a solution. As nutrients are absorbed by a plant, the EC level decreases as there are fewer salts in the solution. Conversely, EC increases when water is removed from the solution through the process of evaporation or transpiration. If EC increases, it can be decreased by adding water. If EC decreases, it can be increased by adding a small amount of nutrients. Ideally, EC should be maintained between 1.15 mS/cm and 1.25 mS/cm[16].

Irrigation time is another crucial factor for plant development. The irrigation can be done in a non-continuous or intermittent manner, so that production may vary according to the Irrigation Time On (IT-On) and Irrigation Time Off (IT-Off). Some studies seek to analyze lettuce behavior under different combinations of IT-On and IT-Off. Soares (2020)[18] tested combinations of 15 minutes of IT-On to 15, 30, and 45 minutes of IT-Off, showing that a IT-Off of 60 minutes can reduce the electricity costs by 53.57% without compromising productivity. On the other hand, Zanella *et al.* (2008)[19] tested combinations of 15 minutes of IT-On to 5, 15, and 30 minutes of IT-Off, showing that the shortest IT-Off time resulted in the greatest mass gain for the tested crops.

The Table 2.2 presents a review of the parameters that must be controlled for hydroponic lettuce cultivation (except nutrients which were reviewed in Table 2.1).

Table 2.2: Parameters review.

Parameter	Minimum	Maximum
AT	19 ℃	24 ℃
ST	24 ℃	26 ℃
RH	5%	7%
CO_2	1000 ppm	1500 ppm
PAR	10 $mol/m^2/d$	20 $mol/m^2/d$
AF	0.1 $m^3/min/m^2$	0.7 $m^3/min/m^2$
DO	7 ppm	10 ppm
pH	5.6	6
IT-Off	5 min	45 min

2.2.2 Cultivation Process

The hydroponic lettuce cultivation consists of two phases. The first phase begins with the seeding of lettuce in trays and ends with the transplanting of germinated seedlings into the hydroponic system, lasting 11 days. The second phase occurs between transplanting and harvesting, lasting 24 days. Thus, the complete cycle of lettuce lasts for 35 days[16].

The planting process begins with the placement of substrate and cotton in each tray slot, followed by seed insertion. The trays are then situated on a pool-type bench, where a nutrient solution, diluted to 50%, circulates in a thin layer, ensuring the substrate remains moist. When seedlings are produced in substrate, it is crucial to wash the roots in clean water before their introduction into the hydroponic system, to eliminate any residual matter. The removal, washing, and positioning of the seedling should be executed with care to prevent damage to the leaves, stems, and roots[2].

Plants should be monitored, before and after transplanting, at least once a day to detect potential issues such as reduced growth, chlorosis, necrosis, wilting, insect or pest incidence, phytopathogens, or other anomalies. Detecting and identifying problems in the early stages of occurrence prevents or reduces proliferation[2]. The grower should be familiar with the appropriate climatic conditions for hydroponic lettuce production and ensure that they are maintained inside the environment through regulatory mechanisms[2]. Similarly to climatic conditions, other parameters must be maintained, either automatically or through periodic manual adjustment.

The harvest should be done at the end of its cycle when it reaches its maximum development. Harvesting beyond the deadline may result in plants with altered flavor and texture. The harvesting and packaging process requires care to keep the product clean and avoid mechanical damage[2]. After harvesting, the following characteristics can be checked: Fresh Matter Mass (FMM), Dry Matter Mass (DMM), leaf area, height, circumference. This information can be useful for analyzing the qualitative characteristics of the plants.

2.3 Optimization Methods

Real-world problems are represented computationally through models ranging from mathematical equations to complex structures with multiple pointers. After the representation of these problems, many tasks can be performed such as simulations, visualizations, integrations with other models, etc. One of these tasks involves optimizing the model, giving rise to an area that deals with optimization problems. Due to various studies, many optimization methods have been developed. These methods are basically divided into three classes: calculus-based; enumerative; and, random[5].

Calculus-based methods are subdivided into indirect and direct approaches. Indirect methods seek local extremes by solving sets of nonlinear equations, adjusting the objective function to zero[20]. In contrast, direct methods pursue local optima by iteratively adjusting the parameters of the function to move towards the optimal point, a process known as the hill-climbing method. These methods are not robust for several simple reasons. Firstly, they seek local points in the vicinity of the current point, which means that the maximum may not be reached depending on the initial point[5]. Moreover, both methods assume that functions are continuous in the optimization intervals and admit derivatives. However, in real-world problems, functions are often discontinuous, making their application possible only in very restricted domains[21].

Enumerative methods are quite simple and resemble the natural form of optimization used by humans. In a finite space, values of the objective function are sought one at a time. Despite its simplicity, this method proves to be extremely inefficient when applied to larger search spaces[5].

Finally, random techniques rely on random choices to guide highly exploratory methods. The essential point to note about algorithms of this type is that they do not necessarily imply aimless searches. Algorithms categorized as random methods have been widely employed in complex search spaces where other techniques are not feasible. Their use is attributed to the fact that there is no need to model mathematical functions representing the problem or calculate derivatives[5].

It is worth noting that algorithms classified as random methods are often inspired by natural phenomena. These phenomena, in general, prove to be highly effective when applied in nature, and it is believed that such effectiveness is similar when applied to computationally modeled problems[22].

2.3.1 Brute-Force Algorithm

An algorithm from the enumerative class of optimization methods is the brute-force algorithm. This algorithm has the characteristic of being able to evaluate all combinatorial possibilities of a problem according to a range of variation for parameter values.

An example could involve testing the maximum weight of a plant as a function of the quantity of three nutrients, N_1, N_2, N_3, each of which can take a value of 0 or 1. In this scenario, the result of each combination could be calculated to identify the optimal one among them, as presented in Table 2.3.

The brute-force algorithm uses the concept of graphs to generate a tree and, from this tree, obtain all possible combinations. A graph is a data structure where each data is

Table 2.3: Example of parameter combinations.

N_1	N_2	N_3
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

inserted into a structure called a node, and each node may or may not be linked to one or more nodes. The tree is a specific type of graph where the nodes are interconnected hierarchically. Thus, the tree can have three types of nodes: root, which may or may not have children but does not have a parent; intermediate, which have a parent and one or more children; and leaf, which have a parent but do not have children[23].

The Figure 2.2 illustrates the tree created by the algorithm to generate all possible combinations of the problem. Note that each combination is obtained through the sequence of values of the nodes in each branch of the tree, that is, each path from each child of the root to a leaf. It is also worth noting that the number of children of each node is related to the number of possibilities for the values of each variable, which in this case are two: 0 and 1. And, the number of levels is related to the number of variables excluding the root, which in this case are three: N_1, N_2, N_3.

Thus, the number of combinations can be calculated by Equation 2.1, where C is the number of combinations, V is the number of possible values for each parameter, and Q is the quantity of variables to be combined in each subset[24].

$$C = V^Q \tag{2.1}$$

It is important to note that the larger the desired precision for the optimal value to be found, the smaller the range of variation of the variable values should be, that is, the larger the number V. Consequently, although it is effective, depending on the problem it

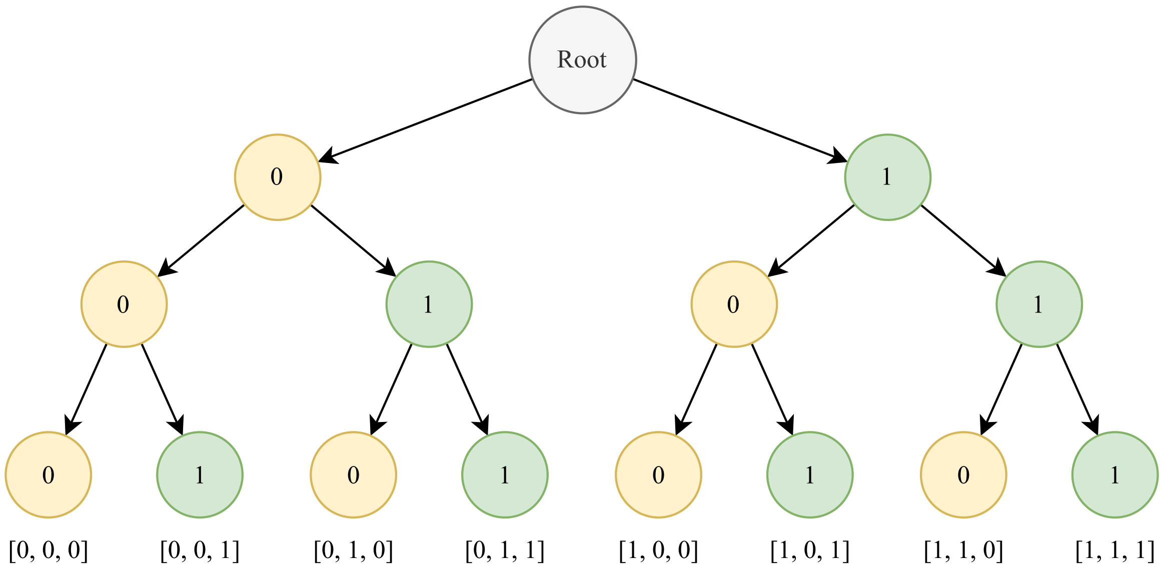


Figure 2.2: Combination tree. Source: adapted from Rosen (2012)[24]

can be computationally very costly and even unfeasible[24].

Operationally, to extract a combination, the algorithm traverses all nodes, descending each level, adding their values from each node to a list until reaching a leaf. Upon reaching it, the combination is concluded, and a new combination begins, removing the last added value, and stepping back one level in the tree, proceeding to form a new combination if there are more possibilities, or removing the last value from the vector again and stepping back levels until finding a node with an unexplored branch[25].

The implementation of the algorithm can be divided into different stages, as shown in Figure 2.3. Initially, combinations are generated from a tree of possibilities, then the combinations are evaluated, and finally, the best value is selected.

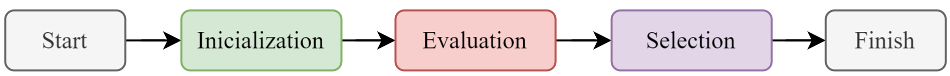


Figure 2.3: Brute-force algorithm. Source: adapted from Floyd (1967)[25]

2.3.2 Evolutionary Algorithms

Evolutionary algorithms use iterations to find results in the search space. These algorithms employ random techniques to guide highly exploratory searches. Examples of this type of algorithm include GA, Simulated Annealing (SA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) and AIS[5].

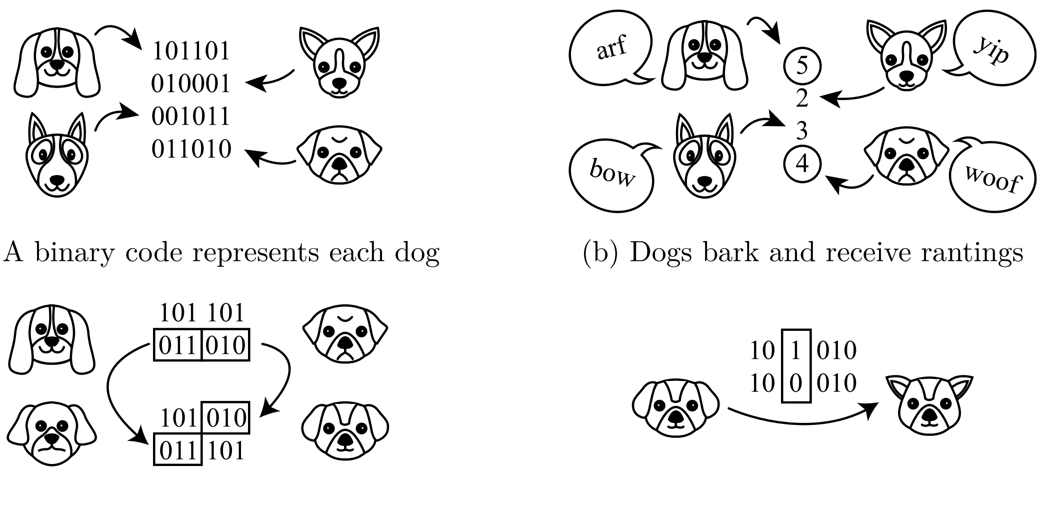
Although they have much in common, the different inspirations for their implementations lead to distinct behaviors that may be more suitable in specific situations. Two of these algorithms, GA and AIS, are presented below. While GA stand out for their ability to be relatively faster in finding a good solution, AIS excels in its ability to always find better results, despite being slower.

Genetic Algorithm

In his theory of evolution, Darwin (1859)[26] asserts that all individuals within a population are different, and due to this variation, some are better adapted to the environment than others. Those that are better adapted have a greater chance of surviving and reproducing, passing on their characteristics to future generations. Thus, the term evolution can be defined as the changes in a species over generations due to its adaptations to the environment in which it is situated. This evolutionary mechanism is called natural selection[5].

This algorithm emerged in the 1960s from John Holland and his research group at the University of Michigan, with their publication “Adaptation in Natural and Artificial Systems”[27], earning him the title of creator of genetic algorithms. However, much of the credit for this work is attributed to one of his students, David Goldberg, and his publication titled “Genetic Algorithms in Search, Optimization, and Machine Learning”[20], explains further[5].

The Figure 2.4 illustrates an analogy of natural evolution with GA. The process begins with a random population of individuals, in this case, a group of dogs. Each sequence of binary values represents a chromosome, which contains the characteristics of a dog. The ability to bark loudly is associated with this sequence of binary values. If it is desired to create a dog with the loudest bark, then the rule is to select only a few dogs with loud barks to be kept for reproduction. It is presumed that there is a way to measure how loud a dog barks, and these dogs then bark and receive a score based on their barks. Two of them are chosen to generate two offspring by crossing their chromosomes. The result of this crossing is the offspring, which usually replace the dogs that did not perform well, keeping the population the same. Finally, mutation occurs, altering some gene of the chromosome, modifying the offspring to some extent. Each repetition results in a new generation until the goal is achieved[21].



(a) A binary code represents each dog

(b) Dogs bark and receive rantings

(c) The result replace poor performers

(d) Introduce mutations

Figure 2.4: Genetic biology analogy. Source: adapted from Haupt (2004)[21]

The GA begins like any other optimization algorithm, setting up the variables and the cost function. And, it ends like any other optimization algorithm, testing for convergence. In the middle of the process, however, this algorithm is different[21]. The steps of the GA are shown in Figure 2.5.

A cost function generates a result from a series of input variables (chromosome). The objective is to modify the output into something desirable by finding appropriate values for the input variables. For example, this is done without thinking when filling a bathtub with water. The cost is the difference between the desired temperature and the current

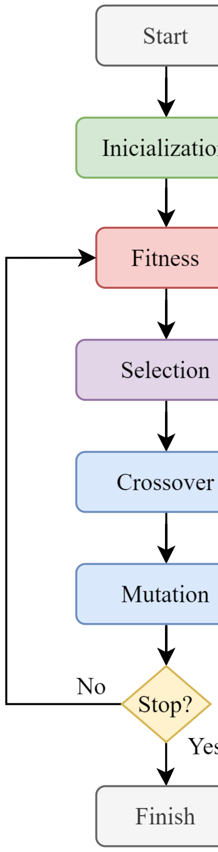


Figure 2.5: Genetic algorithm. Source: adapted from Haupt (2004)[21]

temperature of the water. The input variables are how much the hot and cold taps are open. In this case, the cost function is the experimental result of putting your hand in the water[21]. It is worth mentioning that the term “cost” is often addressed by fitness.

A chromosome is a vector of values of the variables to be optimized. Thus, considering that p is a parameter, and that a chromosome can have n variables, then a chromosome can be expressed as shown in Equation 2.2. The cost, which is a result of applying the input variables to a cost function, can be expressed as shown in Equation 2.3.

$$chromosome = [p_1, p_2, p_3, \ldots, p_n] \tag{2.2}$$

$$cost = f(chromosome) = f(p_1, p_2, p_3, \ldots, p_n) \tag{2.3}$$

Most optimization problems require limits for the variables. Allowing the weight of a car to reach zero or its width to be equal to 10 meters is impractical. Unrestricted variables can take on any value, while constrained variables adhere to three conditions: first, strict limits in the form of $>$, $<$, $\geq$, $\leq$ can be imposed on the variables; second, the variables can be transformed into new variables that include the constraints; finally, there may be a finite set of variable values to choose from, and all values are within the region of interest[21].

Since the GA has variables with values that goes from 0 to 1, when working with continuous values of restricted variables, it is necessary to normalize the real interval to a range between 0 and 1. To do this, the Equation 2.4 is used to normalize a variable, where p is the value of the variable, $p_{\min}$ is the minimum value of the interval, $p_{\max}$ is the maximum value of the interval, and p_{normal} is the normalized value[28].

$$p_{normal} = \frac{p - p_{min}}{p_{max} - p_{min}} \tag{2.4}$$

The initialization of the population in the GA occurs by generating a group of chromosomes where each chromosome represents an individual. The population size is defined, and the values of each gene of the chromosome are generated randomly[28].

Selection is the stage where two chromosomes from the population are selected to cross, generating two new chromosomes. However, there are several ways to select the chromosomes for crossing. The pairing techniques include: top-down; roulette wheel; biased roulette; tournament selection[21].

The top-down pairing technique is the simplest of all. It consists of pairing chromosomes in pairs ordered by cost, so all those in even positions are combined with those in odd positions in order from the first to the last[21].

In the biased roulette technique, chromosomes that obtained the best result will have a higher chance of being combined, however, those that had a not-so-good result can still be chosen. For this, the probability of chromosome n is calculated using the Equation 2.5, where N is the number of chromosomes, and P_n is the probability of n.

$$P_n = \frac{N - n + 1}{\sum_{n=1}^{N} n} \tag{2.5}$$

Finally, the tournament technique mimics the mating competition that occurs in nature, where two chromosomes are randomly chosen, and the one with the best result will be chosen to be the parent. Each selected pair results in a different combination[21].

Crossover is the process where the selected chromosomes are crossed to create two new chromosomes. The most common way to cross is by randomly choosing a point within the chromosome and making the exchange. The first child gains the first part of the first parent and the second part of the second parent, and the second child gains the first part of the second parent and the second part of the first parent[21].

Mutation is the step that performs a mutation on the chromosome based on a certain percentage. Mutation is one of the ways that the GA explores the space in search of the optimal value[21].

The number of generations evolved depends on whether an acceptable solution is reached or if a certain number of generations is exceeded. After a certain time, all chromosomes and their associated costs become the same if there is no mutation. At this point, the algorithm should be stopped[21].

Artificial Immunological System

Belonging to the class of random optimization methods, the AIS is based on the natural immune system. The natural immune system is a complex of cells, molecules, and organs that aim to limit the damage caused by pathogens in the body through an immune response. One way to respond is by secreting antibody molecules from B cells. Antibodies are Y-shaped receptor molecules attached to the surface of a B cell, designed to recognize and bind to the epitope of an antigen. The interaction between the antigen and the antibody is measured through affinity, as illustrated in Figure 2.6[29].

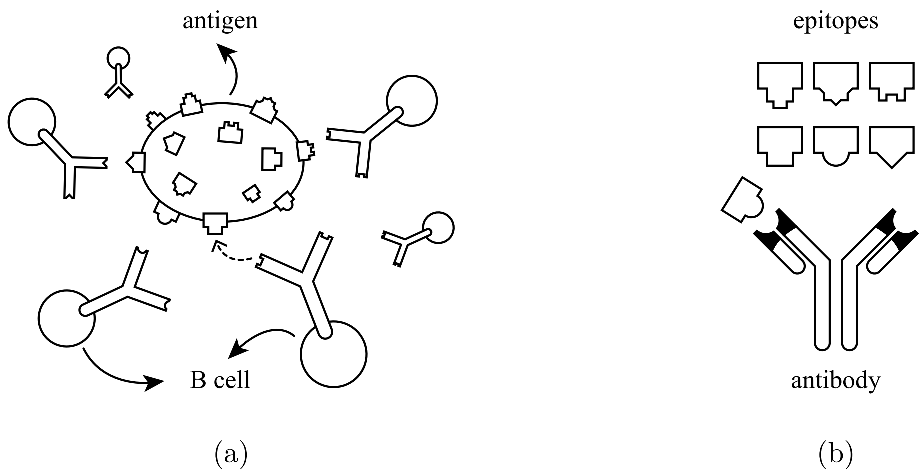


Figure 2.6: Immunological system. Source: adapted from De Castro (2000)[29]

The implementation of AIS explores the theory of clonal selection, establishing the idea that only antibodies that recognize the antigen proliferate, generating copies (clones). Then genetic mutation occurs in the clones, generating antibodies with high affinity for the antigen, which, due to the process of natural selection, become memory cells. This implies better adaptation of individuals and faster responses to future antigens[30].

Thus, the objective of AIS is to develop a group of antibodies, where each one represents a solution within the search space. Initially, the algorithm provides a local search through the affinity maturation (hypermutation) of cloned antibodies. Therefore, the scope of the local search is inversely proportional to their ranking. Then, a second mechanism involves the random insertion of antibodies into the population to increase diversity, leading to a broader exploration of the space in search of the optimal result[30].

As depicted in Figure 2.7, the AIS algorithm initiates by generating a random population of antibodies. Each antibody's affinity to the antigen is then calculated. Antibodies with the highest affinity are selected for cloning. Each clone undergoes a hypermutation, potentially altering its characteristics based on its affinity. New antibodies are introduced to maintain the initial population size. The termination criterion is assessed. If unmet, the process repeats from the evaluation step. Upon meeting the termination criterion, the process ends, and the antibody with the highest affinity is presented as the solution[29].

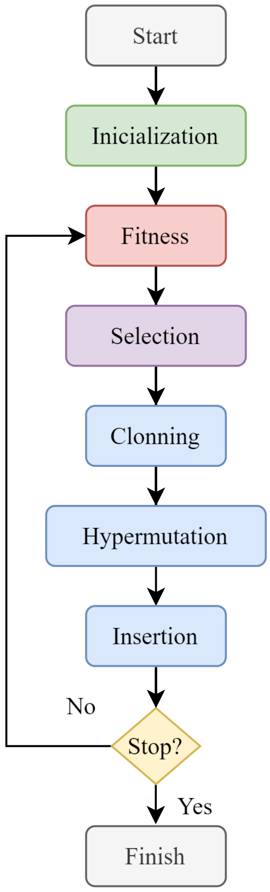


Figure 2.7: Artificial immunological system. Source: adapted from De Castro (2000)[29]

The quantity of clones is determined by Equation 2.6, where c represents the number of clones, $round()$ is a rounding function, β is the clonal multiplication factor, A is the total number of antibodies, and i is the affinity value of the current antibody. The hypermutation follows Equation 2.7, where m is the probability of modification, ρ defines the mutation rate, and f is the normalized affinity of the current antibody[30].

$$c = round(\frac{\beta A}{i}) \tag{2.6}$$

$$m = (\frac{1}{\rho})exp(-f) \tag{2.7}$$

2.4 Artificial Neural Networks

Inspired by the hypothesis that mental activity primarily consists of biological activity in neural networks into the brain, some early works focused on creating artificial neural networks. The Figure 2.8 shows a simple mathematical model of a neuron developed by McCulloch and Pitts (1943). An ANN is just a collection of these connected neuron units. The properties of the network are determined by the topology and properties of the neurons[31].

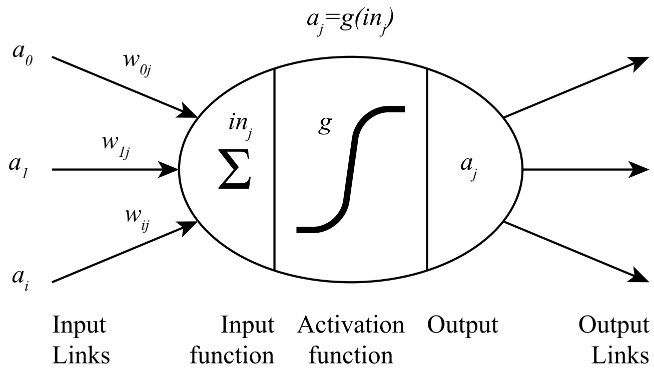


Figure 2.8: Neuron model. Source: adapted from Russel and Norvig (2010)[31]

The Equation 2.8 represents the mathematical model of a neuron in an ANN. Here, a_j represents the output of the neuron, determined by applying an activation function $g()$ to the result of the weighted sum of inputs (a_i) multiplied by their corresponding weights ($w_{i,j}$), summed with the bias term b. The inputs (a_i) represent the input signals to the neuron, while the weights ($w_{i,j}$) indicate the importance of these inputs to the neuron's output. The bias term (b) is an additional parameter that controls the neuron's activation threshold, affecting its propensity to fire. Together, these elements form the basis of information processing in artificial neural networks, allowing neurons to respond in complex ways to varied stimuli[31].

$$a_j = g\left(\sum_{i=0}^{n}(w_{i,j} \times a_i) + b\right) \tag{2.8}$$

The activation function $g()$ is hard, as represented in Figure 2.9 (a); in this case, the term "perceptron" is used. Alternatively, the activation function can be smooth, as represented in Figure 2.9 (b); in this case, the term used is "sigmoid perceptron". Both forms of activation ensure the important property of the neural network representing a non-linear function[31].

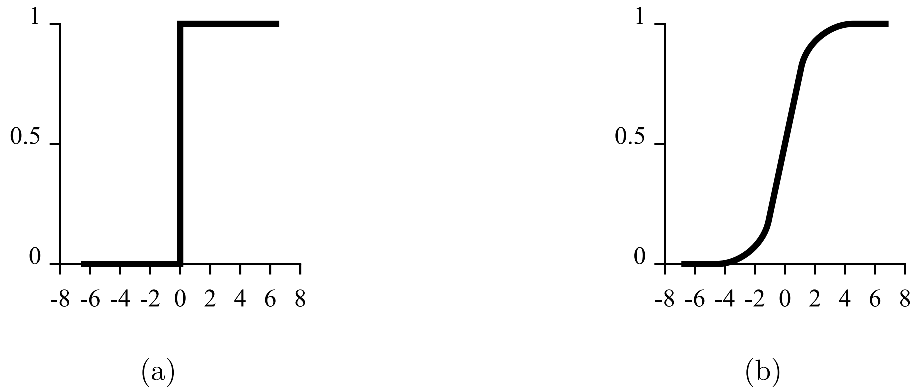


Figure 2.9: Activation functions. Source: adapted from Russel and Norvig (2010)[31]

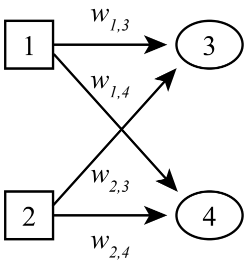
After defining the mathematical model for neurons, the next step is to define how to connect them in a network. There are two fundamentally different ways to do this. A feed-forward network has connections in only one direction, that is, it forms an acyclic graph. Each neuron receives an input and delivers an output; there are no cycles. This type of network represents a function over the input and has no state other than its weights. A recurrent network, on the other hand, uses its output as input. This means that the activation levels of the network form a dynamic system that can reach a stable, oscillatory, or chaotic state. Additionally, the network's response depends on the input from its initial state and possible previous states[31].

2.4.1 Feed-forward Neural Networks

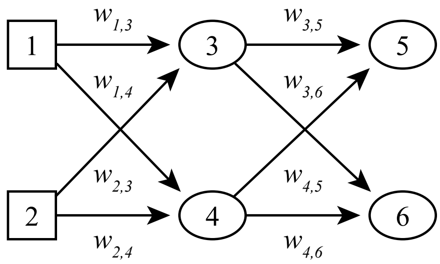
Feed-forward neural networks are typically arranged in layers, such that each neuron receives inputs generally from the immediately preceding layer. The next two subsections explain this type of network, with only one output layer connecting to the input, and with one or more hidden layers connected between the input and output. Additionally, the examples presented focus on regression problems, but neural networks can also be used to solve classification problems, where there are two or more outputs[31].

Single-layer feed-forward neural networks

A network where all inputs are connected directly to the output is called a single-layer neural network, or perceptron network. The Figure 2.10 shows a simple network with two inputs and two outputs. With a network like this, one can learn the sum of two bits, for example. All the necessary data for training is provided in Table 2.4[31].



(a)



(b)

Figure 2.10: A perceptron network. Source: adapted Russel and Norvig (2010)[31]

Table 2.4: Training data example. Source: Russel and Norvig (2010)[31]

x_1	x_2	y_3 (carry)	y_4 (sum)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

The first thing to note is that a perceptron network with m outputs is actually m separate networks, because each weight affects only one of the outputs. Hence, there is an m separate training. Additionally, depending on the type of activation function used, the training process will follow either the perceptron learning rule or the logistic regression[31].

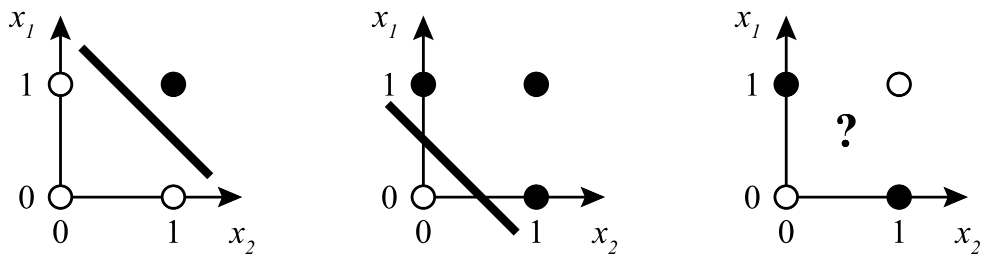
The perceptron learning rule is shown in Equation 2.9, where w_i represents the weight associated with the i-th input of the model. During training, the weights are updated iteratively to minimize the error between the desired output (y) and the current output of the model ($h_w(x)$). The learning rate α controls the magnitude of weight updates. The difference between the desired output and the current output, multiplied by the i-th input (x_i), determines the direction and magnitude of the weight update w_i[31].

$$w_i \leftarrow w_i + \alpha\,(y - h_w(x)) \times x_i \tag{2.9}$$

The update of weights through logistic regression can be represented by Equation 2.10. In this equation, w_i represents the weight associated with the i-th input of the logistic regression model. During the training process, the weights are updated iteratively to minimize the error between the desired output (y) and the current output of the model ($h_w(x)$). The learning rate α controls the magnitude of weight updates. The expression $y - h_w(x)$ calculates the error between the desired output and the current output of the model. Multiplying this error by the product of $h_w(x)$ and $(1 - h_w(x))$ takes into account the slope of the logistic (sigmoid) function, as shown in Figure 2.9 (b), which is the commonly used activation function in logistic regression. This term adjusts the magnitude of the weight update according to the model's confidence in its own prediction. Finally, x_i represents the i-th input of the model, and its multiplication completes the weight update, adjusting it according to the importance of the input to the model[31].

$$w_i \leftarrow w_i + \alpha\,(y - h_w(x)) \cdot h_w(x) \cdot (1 - h_w(x)) \cdot x_i \tag{2.10}$$

If either of these methods is tried on the bit adder data in a single-layer network, something interesting happens. Unit 3 learns the CARRY function easily, but unit 4 completely fails to learn the sum function. This doesn't mean there's a problem with unit 4; the problem lies with the sum function itself. Linear classifiers can represent dividing lines in the input data. This works for AND and OR functions; however, the sum function is an XOR (exclusive or) function of two inputs. The Figure 2.11 illustrates that this function is not linearly separable, so the perceptron can't learn it[31].



(a) x_1 **and** x_2 (b) x_1 **or** x_2 (c) x_1 **xor** x_2

Figure 2.11: Function types. Source: adapted Russel and Norvig (2010)[31]

Multilayer feed-forward neural network

McCulloch and Pitts (1943) knew that a single neuron would be unable to solve all problems. Their publication proves that this unit can represent basic binary functions AND, OR, and NOT, and discusses that the desired functionality can be obtained by combining a large number of units in networks (possibly recurrent) at different depths. The problem was that they did not know how to train these networks[31].

This becomes easy when it is considered that a network is seen as a function $h_w(x)$ parameterized by the weights w. Considering a simple network like the one shown in Figure 2.10 (b), which has two input units, two hidden units, and two output units. Given an input vector $x = (x_1, x_2)$, the activation of the input units is set to $(a_1, a_2) = (x_1, x_2)$, and the output of unit 5 is given by:

$$a_5 = g(w_{0,5} + w_{3,5}\, a_3 + w_{4,5}\, a_4)$$
$$= g(w_{0,5} + w_{3,5}\, g(w_{0,3} + w_{1,3}\, a_1 + w_{2,3}\, a_2) + w_{4,5}\, g(w_{0,4} + w_{1,4}\, a_1 + w_{2,4}\, a_2))$$
$$= g(w_{0,5} + w_{3,5}\, g(w_{0,3} + w_{1,3}\, x_1 + w_{2,3}\, x_2) + w_{4,5}\, g(w_{0,4} + w_{1,4}\, x_1 + w_{2,4}\, x_2))$$

Derivatives of such expressions with respect to the weights can be calculated, enabling the use of the method of loss minimization by gradient descent for network training. Given the potential for the function represented by a network to be nonlinear, neural networks can be viewed as a tool for performing nonlinear regression[31].

2.4.2 Learning in Multilayer Networks

In a network with multiple outputs, it is crucial to treat the network as a vector function h_w rather than a scalar function. This implies that the network's output should be a vector, such as $[a_5, a_6]$ in the case of the network in Figure 2.10 (b), rather than a single value a. The target output y should also be a vector. While a perceptron network allows for the decomposition of the problem into separate problems for each output, this approach is not applicable in a multi-layer network. However, this interdependency can be managed by employing additive loss functions with respect to the components of the error vector $y - h_w(x)$. For the L_2 loss, known as the square of the difference, the following is observed:

$$\frac{\partial}{\partial w}\mathrm{Loss}(w) = \frac{\partial}{\partial w}|y - h_w(x)|^2 = \frac{\partial}{\partial w}\sum_k (y_k(y_k - a_k)^2) = \sum_k \left(\frac{\partial}{\partial w}(y_k - a_k)^2\right) \tag{2.11}$$

Here, the index k varies over the nodes in the output layer. Each term in the final sum represents the gradient of the loss for the k-th output, calculated as if the other outputs did not exist. Therefore, one can decompose a learning problem with m outputs into m learning problems, provided that the gradient contributions of each are summed when updating the weights[31].

The addition of hidden layers into a network introduces complexity. The error at the output layer, $y - h_w$, is straightforward, but the error at the hidden layers is less clear as training data does not specify the hidden nodes' values. However, it has been found possible to backpropagate the error from the output layer to the hidden layers. This backpropagation process originates directly from a derivation of the general error gradient. The weight update rule at the output layer aligns with Equation 2.10. Let Err_k denote the k-th error from the vector $y - h_w$. The modified error can be defined as $\Delta_k = \mathrm{Err}_k \times g'(in_k)$. Consequently, the weight update rule is executed according to Equation 2.12[31].

$$w_{j,k} \leftarrow w_{j,k} + \alpha \times a_j \times \Delta_k \tag{2.12}$$

The error term for outputs, used to update connections between input and hidden units, is defined similarly. Backpropagation is then performed. Each unit j contributes to a fraction of the error δ_k in each connected output. δ_k is divided based on the connection length between the hidden unit and output, creating δ_j. The backpropagation rule is outlined in Equation 2.13 and the algorithm in Algorithm 1[31].

$$\Delta_j = g'(in_j) \sum_k (w_{j,k} \Delta_k) \tag{2.13}$$

Algorithm 1 Backpropagation algorithm. Source: Russel and Norvig (2010)[31]

Δ, a vector of errors, indexed by network node
repeat
for each weight $w_{i,j}$ in network **do**
$w_{i,j} \leftarrow$ a small random number
end for
for each example (x, y) in examples **do**
for each node i in the input layer **do**
$a_i \leftarrow x_i$
end for
for $l = 2$ to L **do**
for each node j in layer l **do**
$in_j \leftarrow \sum_i w_{i,j} a_i$
$a_j \leftarrow g(in_j)$
end for
end for
for each node j in the output layer **do**
$\Delta[j] \leftarrow g'(in_j) \times (y_j - a_j)$
end for
for $l = L - 1$ to 1 **do**
for each node i in layer l **do**
$\Delta[i] \leftarrow g'(in_i) \times \sum_j w_{i,j} \Delta[j]$
end for
end for
for each weight $w_{i,j}$ in network **do**
$w_{i,j} \leftarrow w_{i,j} + \alpha \times a_i \times \Delta[j]$
end for
end for
until some stopping criterion is satisfied

The backpropagation process can be summarized as:

- computes the Δ values for the output units, using the observed error;
- starting from the output layer, iterates the following for each layer of the network until the first hidden layer is reached:
  - propagates the Δ value back to the previous layer,
  - updates the weights between the two layers.

Upon training, one can observe the evolution of training and validation errors over epochs. The Figure 2.12 depicts the relationship between these two types of errors. When both are decreasing, it indicates that the model is learning effectively and can generalize the seen data. However, when the validation curve starts to rise after a certain point, it suggests that the model might be overfitting the data[31].

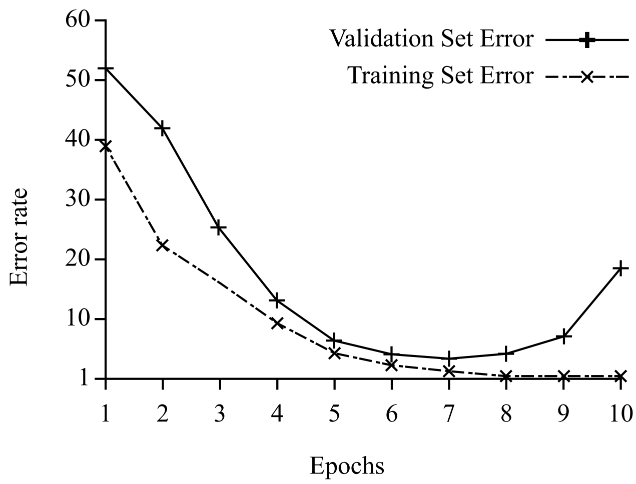


Figure 2.12: Training graph. Source: adapted from Russel and Norvig (2010)[31]

In the evaluation of the model obtained from the training of the ANN, tests are conducted. For a regression problem, the metrics employed aim to compare the error between the actual and predicted values. The Mean Absolute Error (MAE) considers the average of the absolute values of the differences between the predictions and the actual values. The Mean Square Error (MSE) considers the average of the squares of the differences between the predictions and the actual values. The Root Mean Square Error (RMSE) is the square root of the MSE. The calculations for the MAE, MSE, and RMSE are performed respectively through Equation 2.14, Equation 2.15, and Equation 2.16, where: n is the total number of observations or data points; y_i is the actual value of the i-th observation; and, $\hat{y}_i$ is the predicted value of the i-th observation[32]. There are advantages in analyzing the error in each of these metrics. The MSE is more sensitive to larger errors compared to the MAE, however, it is presented on a different scale. The RMSE, in addition to being more sensitive, like the MSE, is presented on the same scale as the MAE.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.14}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.15}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.16}$$

It is also important to determine the best structure for the network. Choosing a network that is too large may lead to memorizing examples and failing to generalize input data. Therefore, a common approach is to test different configurations and retain the one that yields the best results[31]. Finally, one must be careful with the quality of the data provided to the neural network. It is also important to consider that preprocessing steps should be taken to ensure proper learning during training. There are many techniques that can be applied for data conversion and normalization, as well as handling outliers, missing data, and imbalanced data[31].

2.5 Related Works

Understanding related works is important to define the most suitable approaches for the research, and also to avoid or explain potential issues. This section presents works that involved different techniques for hydroponic lettuce optimization, possible techniques not yet applied, and details about cultivation in closed environments.

2.5.1 Lettuce Optimization Using Evolutionary Algorithms

Valenzuela *et al.* (2018)[33] developed an experiment to evaluate the use of GA to find the best conditions for lettuce production in terms of light intensity, temperature, and CO_2. Using the leaf photosynthesis model as the basis for fitness function calculation, the system evolved 50 individuals over 100 generations, and concluded that the optimal values obtained were a light intensity of 175.22 μmol/m^2/s, a temperature of 19.36 ℃, and a CO_2 level of 803.02 ppm.

Tessele (2018)[34] conducted a study optimizing lettuce grown in soil using AIS. It was performed on the parameters N, P, and K, and the fitness function was calculated considering the profit obtained from the market price and the fresh mass of the product. The system evolved 80 individuals over 7 generations, and as a result, it was found that the combination of 20.85g of N, 19.30g of P, and 17.64g of K generated the highest profit, surpassing the previously considered recommendation values. However, in their conclusions, the author mentions the need to perform more generations to evaluate the significance of the results obtained.

2.5.2 Optimization Problems Using Artificial Neural Networks

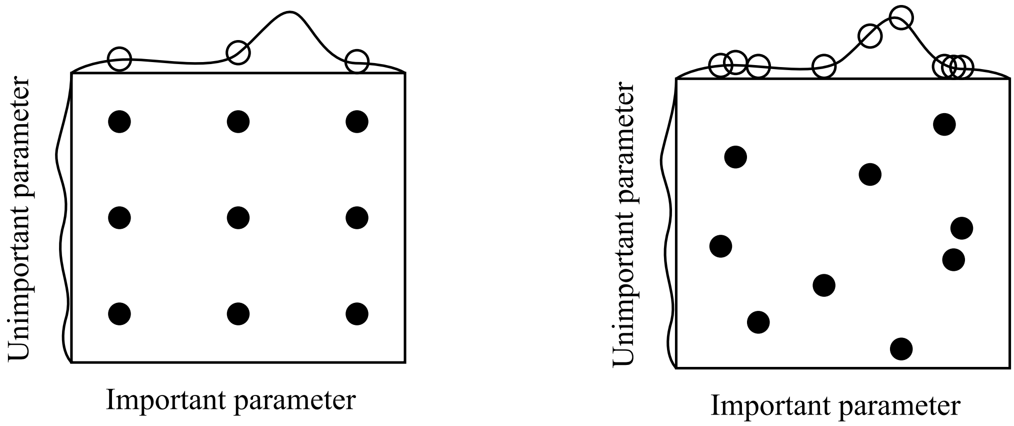
Villarrubia *et al.* (2018)[6] conducted a study on the use of ANN in optimization problems. Problems involving optimization are commonly solved using heuristics and metaheuristics. Exact mathematical methods may be infeasible for solving certain problems; however, heuristics such as GA and AIS are techniques that abstract certain characteristics of the problem in order to find an acceptable, though not necessarily exact, solution.

Metaheuristics, on the other hand, are higher-level procedures designed to find, generate, or select a heuristic that may provide a sufficiently good solution to an optimization problem, especially with incomplete or imperfect information or limited computation capacity. The work developed by Villarrubia *et al.* involved generating a dataset with data obtained from an objective function, training an ANN to create a new objective function, and then solving the optimization using PSO in the new objective function.

An important aspect to consider is the architecture of the ANN. The Kolmogorov theorem states that a multilayer perceptron with 3 layers allows for the precise definition of any continuous function. However, a crucial point is to define the activation function. The ArcTan function was chosen because its derivation is simple and allows for the simplification of functions[6].

A similar approach was utilized to optimize tomato production by Morimoto *et al.* (1993)[7]. The authors employed a ANN in conjunction with a GA to optimize the growth of tomatoes *Lycopersicon esculentum* in a hydroponic system. The growth process was divided into four stages, with the objective being to identify the optimal combination of nutrient concentration and light intensity in the final stage. The ANN was designed with nutrient concentration and light intensity as inputs, and the Total Leaf Length to Stem Diameter (TLL/SD) as the output. The architecture of the ANN consisted of an input layer with five neurons, a hidden layer with five neurons, and an output layer with a single neuron. The size of the database used for the training was not specified. The predictions accurately represented the TLL/SD. The GA was utilized to find the optimal combination of inputs for the generated model.

For implementing an approach using ANN, it is essential to have a dataset that covers correctly the searching space. In this sense, there are some techniques that can be applied to create a database. A straightforward method is to employ a grid search technique, which systematically explores the search space by altering each parameter in a predetermined step, thereby forming a grid. However, as Bergstra *et al.* (2012)[35] pointed out, this technique has a potential drawback: it may excessively explore parameters of lesser relevance while not sufficiently exploring the more crucial ones. A potentially more efficient alternative is to utilize random combinations of parameters. This method ensures a thorough exploration of the important parameters, even if some parameters are not particularly relevant, as illustrated in Figure 2.13.



(a) Grid layout

(b) Random layout

Figure 2.13: Grid versus random search. Source: adapted from Bergstra *et al.* (2012)[35]

2.5.3 Indoor Farming Experiments

Sparks (2018)[3] conducted a study that not only developed a container project for hydroponic production but also included a practical experiment. The experiment involved three production cycles and demonstrated a significant variation in lettuce productivity. The lack of experience of the team in managing the complexities involved in hydroponic production was mentioned as a factor that negatively affected the production. This is evidenced by the improvement in results over the cycles, indicating that as experience was gained, better results were achieved. Another important factor to note is that the variation in plant development is related to the intensity of lighting. The study showed a significantly positive influence of PAR values on the results obtained. Therefore, it is pointed out that adequate lighting and uniform coverage are crucial factors that strongly affect the success of production in indoor environments[3].

Bafort *et al.* (2022)[36] developed a study to evaluate the agronomic and economic viability of cultivating the medicinal plant *Euphorbia peplus* in a MHSC. The impacts of three hydroponic substrates, three light intensities, three plant locations, and two surface areas were tested to assess production and cost. It was evaluated height, FMM, and DMM of the plants. Although it was not commented, significant variations in harvested values can be observed. The economic viability analysis compared with the hydroponic production of Romaine lettuce. The study showed that with the choice of the most favorable conditions it was possible to increase the value of fresh biomass by 200%.

Chapter 3

Methodology

In this chapter it will be explained why a combination of an ANN and the brute-force algorithm was selected as the ML technique for this study. This approach necessitates an initial step of conducting exploratory plantings to generate a dataset. Following the application of the technique, validation plantings are carried out to confirm the found values. This chapter presents a detailed description of the available materials, the methodological decisions, and the methods used in the process. Taking these factors and the characteristics observed in chapter 2 into account, this information aids in understanding the factors that determined the choice of ML technique and how it was executed.

3.1 Materials

This section presents the available materials and its characteristics. The MHSC where the plantings are conducted, and its systems used to control the crops parameters. The monitoring equipments used to check the MHSC conditions during the cycle and to evaluate the result of the production. The lettuce seedlings, which are bought under a specific condition. And, the computer technologies used to implement the ML technique.

3.1.1 Modified Hydroponic Shipping Container

The MHSC used in the experiment is provided by *Federal University of Technology - Paraná* (UTFPR). This MHSC is a standard 40 feet (12.19 m) shipping container, typically used for product transportation, that has been adapted for plant cultivation. As Figure 3.1 shows, inside the MHSC, there is a hydroponic system that consists of 6 channels, each with 43 positions, allowing for the cultivation of up to 258 plants per cycle. The MHSC is installed at the campus of Santa Helena (coordinates: 24°50'49.7"S 54°20'40.1"W), and it is operated by the agronomy department staff.

Figure 3.1: Hydroponic system.

As shown in Figure 3.1, the hydroponic system is a NFT type, installed in a horizontal layout. As explained in subsection 2.1.1, the plants are placed in the spots along the channel, and its roots receive the water with nutrients from the reservoirs through an irrigation system. Also, as explained in subsection 2.1.2, the cultivation in the closed space of the MHSC allows for the control of the internal environment parameters.

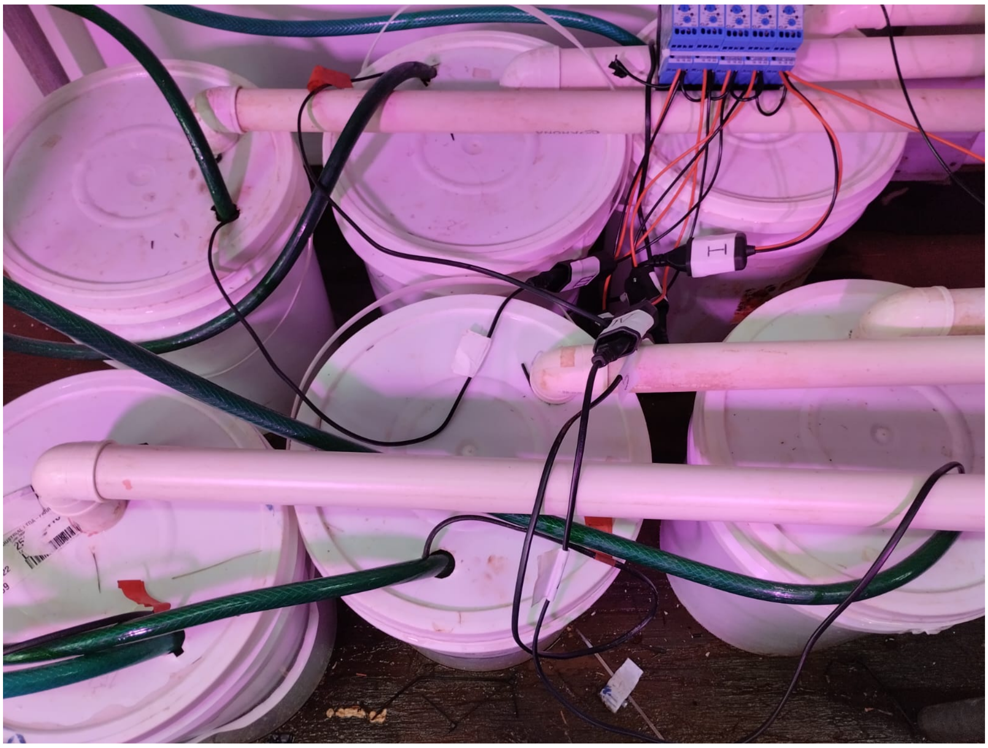


Figure 3.2: Irrigation system.

As depicted in Figure 3.2, each channel of the hydroponic system has its own reservoir and pump, allowing for the delivery of a specific aqueous solution individually. This solution is pumped from the reservoir to the channels via an automated system, which enables the configuration of the IT-On and IT-Off. Although there is an automated system to control the ST, this control only allows for cooling, making it less precise. Lastly, the composition of the aqueous solution is manually maintained by adding nutrients and adjusting the EC, pH, and DO.

As an indoor system, the MHSC is equipped with an artificial lighting system. As shown in Figure 3.3, three lighting lines are installed 30 cm above the channels. Each lighting line consists of 8 Light-Emitting Diode (LED) type lamps, each with a power of 100 W. These LED lamps are Red-Green-Blue (RGB), which allows adjusting the color to suit the specific needs of the plants being grown. The intensity of the lamps can also be adjusted, allowing for precise control over the amount of light the plants receive.

Figure 3.3: Lighting system.

In terms of internal atmosphere control, the AT can be controlled automatically through an air conditioning system. Although it is possible to maintain a certain temperature, it is important to note that this temperature is global, that is, it acts simultaneously on all channels. It is possible to generate an AF by activating ceiling fans, however, these also act globally. It is also important to note that, although AT and AF are global, as the conditioning system and the ceiling fans are installed at specific points, it is not possible to ensure uniform values throughout the MHSC. Finally, the current setup does not include equipment for controlling the RH and CO_2.

In summary, the Table 3.1 draws a parallel between the lettuce parameters that can be controlled, pointed out in section 2.2, and the capabilities of the MHSC. The mechanism can be automatic or manual. The actuation is total when controlling system allows increasing and decreasing the value, otherwise, it is partial. And the effect shows how the control affect the crops, it can be individual (by channel) or general (all channels).

Table 3.1: Parameters controlling review.

Parameter	Mechanism	Actuation	Effect
Nutrients	manual	total	individual
AT	automatic	total	general
ST	automatic	partial	individual
RH	none	none	none
CO_2	none	none	none
PAR	manual	total	general
RGB	manual	total	general
AF	manual	total	general
DO	manual	partial	individual
pH	manual	total	individual
EC	manual	total	individual
IT-On	automatic	total	individual
IT-Off	automatic	total	individual

There are portable equipment in the MHSC that can be used for measurements. One of the available equipment (Akso - AK88) that combines the functions of an EC-meter, pH-meter and DO-meter, which is essential for adjusting and monitoring the nutrient solution. To adjust the lighting, there is a photosynthetic light meter (Hopocolor Technology - OHSP350P), capable of measuring the values of PAR, Photosynthetic Photon Flux Density (PPFD), Red (R), Green (G), Blue (B), and Ultraviolet (UV) light. Finally, to measure the weight gain of the plants, there is a digital scale with milligram precision.

3.1.2 Lettuce Seedlings

There is a wide variety of lettuces of the species (*Lactuca Sativa L.*). Due to availability, this work chose to cultivate the variety called Valentina. This variety was developed and is marketed by Sakata Seed. Its characteristics are: large plants, long leaves, bright green color and moderate crispness; thick stem and vigorous root system; average total cycle: 55 days. As for the strong points, they stand out: high rusticity and wide adaptability; high tolerance to Tip Burn and early bolting. It can be cultivated in hydroponic systems or open fields throughout the year[37].

As mentioned in subsection 2.2.2, the seedlings are transplanted to the hydroponic system in the 11th day after seeding. In this experiment they are not produced by the staff, they are bought from a local company called Viveiro Agrodomus instead, with exact 10 days after being planted. The Figure 3.4 shows a tray with seedlings ready to be transplanted.

Figure 3.4: Lettuce seedlings tray.

Also, the lettuce seedlings are transplanted into the hydroponic system to complete their growth cycle. The Figure 3.1 illustrates the seedlings post-transplantation. It's noteworthy that each seedling is placed in a specific spot along the channels. The placement can be sequential or with intervening empty spots.

3.1.3 Computer Technologies

To implement the system, the chosen programming language was Python[38]. It is a high-level programming language known for its simplicity and readability, making it accessible to beginners while also powerful for experienced developers. Python has extensive standard library and large ecosystem of third-party libraries, which make it suitable for a wide range of applications.

As a development platform, it was chosen Google Colab[39]. Short for Google Colaboratory, is a cloud-based platform provided by Google that allows users to write and execute Python code directly in the browser. It offers free access to computational resources, including Graphics Processing Unit (GPU) and Tensor Processing Unit (TPU), what makes it an attractive option for ML projects. It supports collaborative editing, enabling multiple users to work on the same notebook simultaneously.

And, to implement the ANN it was used the PyTorch library[40]. It is an open-source ML library for Python developed by Meta AI. It provides tensor computation with strong GPU acceleration and is primarily used for deep learning applications. It provides an extensive support for neural network layers and architectures, and integration with popular libraries for data processing and visualization.

3.2 Methodological Decisions

This section outlines the key methodological decisions that guided the research. The optimization objective and the parameters to be optimized are defined first. This is followed by a discussion on the planting schedule, which provides the timeline for the experiment. The chosen ML approach is then explained, detailing the reasons for opting for a combination of an ANN and a brute-force algorithm.

3.2.1 Optimization Objective and Parameters

One important aspect in an optimization project is defining the optimization objective. This project is about optimizing the hydroponic lettuce production, this means that it is expected to have a greater production for the hydroponic lettuce. A greater production can be achieved by improving the product characteristics. As mentioned in subsection 2.2.2, this characteristics can be the FMM, DMM, leaf area, height, circumference. Also, to optimize the production it can be considered increasing the profit. Although, to simplify the process, the FMM maximization was chosen to be the optimization objective in this project.

As summarized in Table 3.1, not all the parameters can be totally and individually controlled. This condition allows the optimization of the following parameters: nutrients, pH, EC, IT-On, and IT-Off. Although, to simplify the problem, only three of them were chosen.

As seen in subsection 2.2.1 there is more than one suggestion for the values of IT-Off, for this reason, it was decided to include it to be optimized. Also, it was noticed that the nutrients have already being deeply studied, because of that, it was decided to not add them as a set of parameters to be optimized. Finally, it was defined to optimize the following parameters: IT-Off, EC, and pH. It's important to note that the other parameters should be set as constants.

Based on the recommendations of subsection 2.2.1 and the experience of the staff operating the MHSC, it was defined the values of the constant parameters. The values of the nutrients are presented in Table 3.2, and values of the other parameters in Table 3.3. Finally, the range of values of the optimizing parameters are presented in Table 3.4.

Table 3.2: Nutrientes.

Substance	**Quantity (g/150L)**
Calcium Nitrate	128.00
Potassium Nitrate	062.00
Monoammonium Phosphate	26.00
Magnesium Sulfate	65.00
Chelated Iron	6.00
Boric Acid	0.90
Copper Sulfate	0.30
Manganese Sulfate	0.24
Zinc Sulfate	0.15
Sodium Molybdate	0.04

Once the parameters values are defined, it is important to define how they are maintained. The parameters that are automatically controlled are set in the system and remain constant. However, the EC, pH and DO parameters, which are manually controlled, require a defined protocol to ensure their stability. For EC, pH and DO, a check is performed twice a day. If the pH is low, the procedure involves dripping 85% phosphoric

Table 3.3: Other parameters.

Parameter	**Quantity**
DO	> 5.5 ppm
ST	22 ºC
AT	25 ºC
IT-On	1 min
PPFD	200 µmol/m^2/s
RB	3:1

Table 3.4: Optimizing parameters panges.

Parameter	**Minimum**	**Maximum**
IT-Off	10 min	50 min
EC	1.0 mS/cm	1.8 mS/cm
pH	5.5	6.5

acid (H3PO4), then measuring again and repeating the process until the desired value is reached. If the pH is high, Potassium Hydroxide (KHO) is dripped, followed by measurement and repetition of the process until the desired value is achieved. In the case of EC, which can only decrease, the procedure involves calculating the amount to be added using proportional reasoning, and then adding the nutrients to the solution. Finally, DO is increased by manually activating the O pump for a short period, followed by measurement and repetition of the process until the desired value is achieved.

3.2.2 Planting Schedule

Considering that there was one year available to design, execute, and evaluate the experiments this project, it was decided to use the three firsts months to the experiment design, and the lasts two months to the experiment evaluation. Then, to know how many cycles is possible to carry on the available time, it is necessary to define the duration of each cycle and the interval between them.

As outlined in subsection 2.2.2, the seedlings are planted 10 days prior to being transplanted into the hydroponic system. Following transplantation, they remain in the hydroponic system for 24 days to complete their growth cycle. Once a cycle is completed, a seven-day interval is required to prepare the system for the next cycle and process the data. For organizational purposes, transplantations are always carried out on the same day of the week. As a result, the hydroponic duration was set to 21 days instead of 24. The schedule was established based on these durations as presented in Table 3.5, allowing carrying out 8 cycles. It is important to note that due to a technical problem with the irrigation system, there was a greater interval between the cycles 5 and 6.

Table 3.5: Planting schedule.

Cycle	**Planting**	**Transplanting**	**Harvesting**
1	25/04/2023	05/05/2023	26/05/2023
2	23/05/2023	02/06/2023	23/06/2023
3	20/06/2023	30/06/2023	21/07/2023
4	18/07/2023	28/07/2023	18/08/2023
5	15/08/2023	25/08/2023	15/09/2023
6	26/09/2023	06/10/2023	27/10/2023
7	24/10/2023	03/11/2023	24/11/2023
8	21/11/2023	01/12/2023	22/12/2023

3.2.3 Machine Learning Approach

As it was observed in subsection 2.5.1, in the study of Valenzuela *et al.* (2018)[33], it was used a GA with 50 individuals over 100 generations. Also, in the study of Tessele (2018)[34], it was used AIS with 80 individuals over 7 generations. Considering that in the MHSC there are 6 channels, it is possible to have a maximum of 6 individuals per cycle, and, the schedule allows over 8 of them, it became clear that this condition would not be great enough to evolve an evolutionary algorithm.

Therefore, the chosen ML approach was to use an ANN combined with a searching algorithm, as mentioned in subsection 2.5.2. Even though the study by Villarrubia *et al.* (1993)[7] did not specify the size of the database, it was expected that the data generated from the current scenario would be sufficient to train the ANN. As the schedule allowed for conducting 8 cycles, 6 cycles were dedicated to exploratory plantings to create a database. The remaining 2 cycles were used for validation plantings to verify the effectiveness of the process.

Although the studies by Villarrubia *et al.* (2018)[6] and Morimoto *et al.* (1993)[7] used PSO and GA algorithms respectively, in this work it was decided to use a brute-force algorithm for the optimization problem in the new objective function generated by the ANN. This decision was influenced by the MHSC constraints, which limit parameter control to a maximum of 2 decimal places. As a result, the optimization step size didn't need to be extremely small to find the optimum value. The use of brute-force was feasible as it didn't require excessive time to process all possible combinations.

3.3 Methods

This section outlines the methodologies employed in this research. The process begins with exploration plantings, involving initial plantings to generate a dataset for training the ANN. The ANN training subsection provides details on the setup and training of the ANN, which generates the regression model that serves as the cost function for the optimization process. Upon training the ANN, a brute-Force optimization algorithm is utilized to pinpoint the optimal parameters that enhance lettuce production. Lastly, the validation plantings subsection describes the verification of the predicted optimal parameters through additional plantings. The whole process is summarized in Figure 3.5.

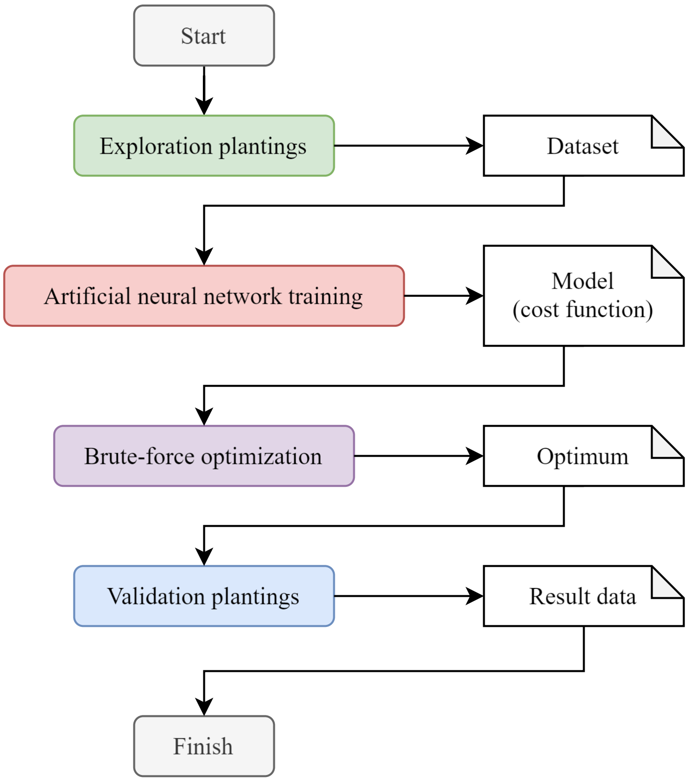


Figure 3.5: Methodology diagram.

3.3.1 Exploration Plantings

As discussed in subsection 2.5.2, according to Bergstra *et al.* (2012)[35], conducting plantings by exploring the search space in a random manner provides a better description of the space through the collected data. Based on this, random values were pre-established for the parameters IT-Off, EC, and pH.

The random values were generated in a normal scale, from 0 to 1, and denormalized to the real value by applying the Equation 3.1, where: p is the real value; p_{max} is the maximum value; p_{min} is the minimum value; and p_{normal} is the normal value. It is important to mention that the ranges of the parameters were established in Table 3.4.

$$p = (p_{max} - p_{min}) * p_{normal} + p_{min} \tag{3.1}$$

Another important consideration concerns the standard sample. To enable comparison of production across cycles, the values of one of the channels were fixed. If the parameters are standard, production should remain constant throughout the cycles. Any changes can be detected by examining the results from this channel. For this reason, the first channel was chosen as the standard sample, and it consistently received the same combination of parameter values: an IT-Off of 19 minutes, an EC of 1.6 mS/cm, and a pH of 6.

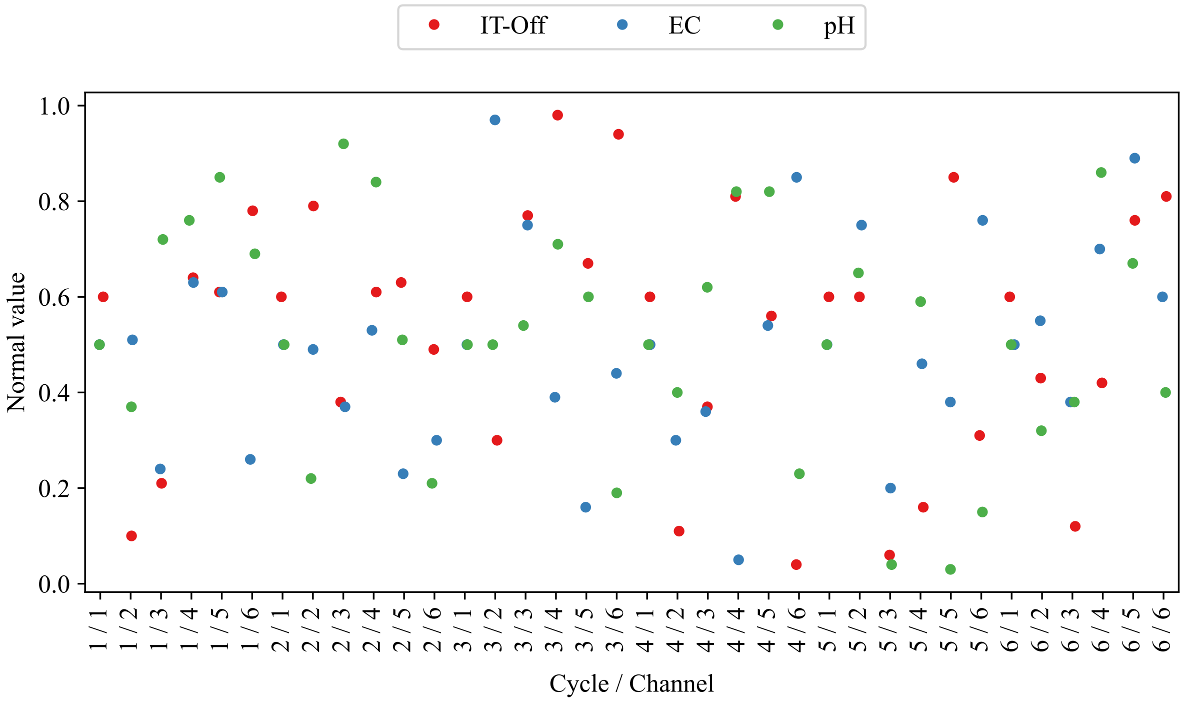


Figure 3.6: Random values chart.

Based on those considerations, the exploration planting values were defined. The Figure 3.6 shows the dispersion of the normal values of the parameters, meaning that the searching space was well explored. And, the Table 3.6 indicates the real values obtained from the denormalization of the random values. Also, it is possible to note that the values of the first channel are repeating every cycle as they are standard samples. Those combinations were planted according to Table 3.5.

Table 3.6: Exploratory planting parameters values.

Cy. / Ch.	IT-Off (min)	EC (mS/cm)	pH
1 / 1	34	1.6	6.0
1 / 2	14	1.6	5.9
1 / 3	18	1.3	6.2
1 / 4	36	1.8	6.3
1 / 5	34	1.7	6.4
1 / 6	41	1.3	6.2
2 / 1	34	1.6	6.0
2 / 2	42	1.6	5.7
2 / 3	25	1.4	6.4
2 / 4	34	1.6	6.3
2 / 5	35	1.3	6.0
2 / 6	30	1.4	5.7
3 / 1	34	1.6	6.0
3 / 2	22	2.2	6.0
3 / 3	41	1.9	6.0
3 / 4	49	1.5	6.2
3 / 5	37	1.2	6.1
3 / 6	48	1.5	5.7
4 / 1	34	1.6	6.0
4 / 2	14	1.4	5.9
4 / 3	25	1.4	6.1
4 / 4	42	1.1	6.3
4 / 5	32	1.6	6.3
4 / 6	12	2.0	5.7
5 / 1	34	1.6	6.0
5 / 2	34	1.9	6.2
5 / 3	12	1.2	5.5
5 / 4	16	1.6	6.1
5 / 5	44	1.5	5.5
5 / 6	22	1.9	5.7
6 / 1	34	1.6	6.0
6 / 2	27	1.7	5.8
6 / 3	15	1.5	5.9
6 / 4	27	1.8	6.4
6 / 5	40	2.1	6.2
6 / 6	42	1.7	5.9

3.3.2 Artificial Neural Network Training

The training of the ANN necessitates the establishment of its architecture, achieved through the application of the hyperparameter tuning technique. This technique facilitates the identification of the optimal configuration. However, the training of the ANN is contingent on the data procured from the exploratory planting. Consequently, it is necessary to address any issues that arise during this phase. This necessitates the application of the feature selection technique, to choose the relevant information, and data engineering, to adjust the values as desired.

As will be detailed in chapter 4, a significant standard deviation in the FMM values of the plants along the channel was observed shortly after the first harvest. Given that the plants in the same channel are exposed to identical values of IT-Off, EC, and pH, this variation was unexpected. The hypothesis was that uneven lighting was the primary cause of this variation. In addition, a significant variation between the values of one cycle to another was observed through the standard channel. Although the exact cause of this variation is uncertain, the main hypothesis is that some uncontrolled variable may have interfered with the results. Despite these occurrences, the applied techniques allowed for better learning in the training of the ANN.

Network Configuration

The ANN was implemented with Pytorch on Google Colab. Based on the studies analyzed in subsection 2.5.2, it was determined that a single hidden layer is suitable for solving regression problems in this kind of complexity. Consequently, the network was designed to include an input layer, an intermediate layer, and an output layer. The output layer consists of a single output, representing the FMM value. The following techniques were applied to define the hyperparameters values, such as: the input parameters, the number of neurons in the intermediate layer, the number of training epochs, the learning rate value, and the weight decay.

The Hyperparameter tuning technique consists of optimizing the values of the hyperparameters in order to find the combination that results in the best learning based on the values of MAE, MSE, and RMSE. A grid search was implemented using the GridSearchCV function from the Scikit-Learn library. Thus, a sequence of values was defined for each hyperparameter as shown in Table 3.7, and training was carried out for each possible combination. In the end, the combination that generated the best learning was: 128 neurons; 500 epochs; a learning rate of 0.01; and a weight decay of 0.0001.

Table 3.7: Hyperparameter values.

hyperparameter	**1st value**	**2nd value**	**3rd value**	**4th value**
neurons	64	128	256	512
learning rate	0.0001	0.001	0.01	0.1
weight decay	0.0001	0.001	0.01	0.1
epochs	250	500	1000	1500

Upon collecting new parameters, it was applied the feature selection technique using a brute-force search to identify the optimal combination of parameters. This led to the training of the network with the discovered hyperparameter combinations. The optimal feature combination was determined based on the best average learning, considering the values of MAE, MSE, and RMSE. Consequently, the input features of the network were defined as: PPFD, channel, position, IT-Off, EC, and pH. The final structure of the ANN was then established as depicted in Figure 3.7.

Also, for the activation function, the Gaussian Error Linear Unit (GELU) was chosen for the hidden layer. The activation function GELU $x\Phi(x)$, proposed by Hendrycks and Gimpel (2016)[41], is the standard Gaussian cumulative distribution function. It has the advantage of weighting the inputs by their value, rather than just by signaling, which results in improvements in its performance. The second layer does not have an explicit activation function applied to it. Thus, the output is linear, which is suitable for predicting continuous values[42].

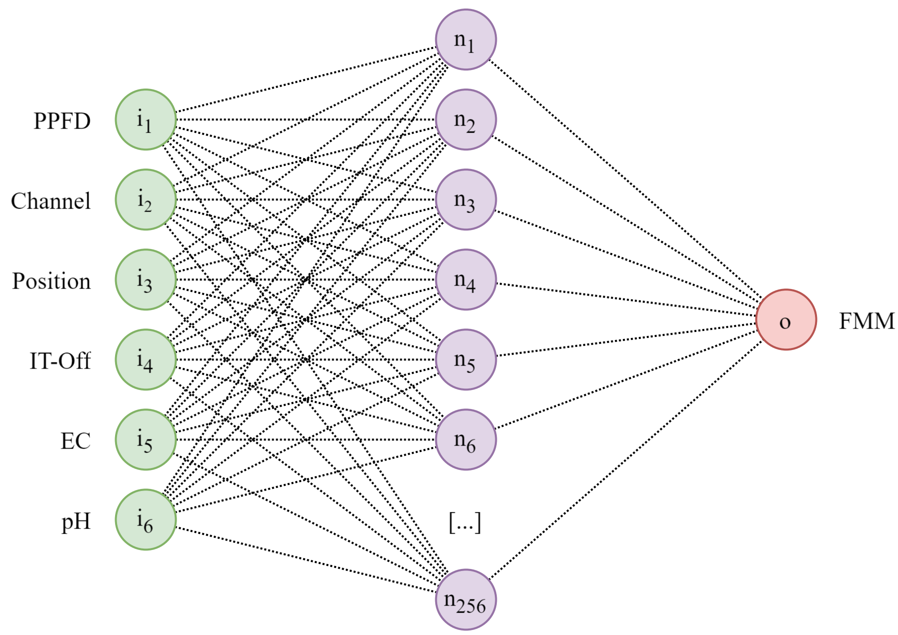


Figure 3.7: Artificial neural network structure.

Data engineering

To ensure that one parameter does not have more relevance than another, it is important that they are normalized on a certain scale. As discussed in subsection 3.3.1, the values of IT-Off, EC, and pH were generated on a normalized scale between 0 and 1, therefore, this scale was maintained. The parameters of PPFD, channel, and position, which had their real values, were normalized to be on the same scale between 0 and 1. For this, the normalization formula Equation 2.4 was used, and their respective maximum and minimum values.

The values obtained from the harvest, however, did not need to be normalized to the same scale as the inputs because they are not an input parameter, but a label. Despite this, due to the variation in these values from one cycle to another, as observed through the average of the standard sample, they also had to be normalized. For this reason, they were normalized through the percentage adjustment in relation to the highest average obtained in the standard sample. Therefore, 104.75g was considered as the highest value for the average of the standard, and the values were normalized according to this value.

Data separation

Before training the ANN, it was needed to address the dataset separation. As discussed in subsection 2.4.2, the training process requires three sets of data: training, validation, and testing data. Initially, it was allocated 20% of the data for validation and testing, with the remainder used for training. However, this approach could introduce bias as the training set contained examples of IT-Off, EC, and pH combinations from plants sharing the same channel.

To mitigate this, it was decided to separate the dataset by cycles. Thus, the designated cycles 1, 2, 3, and 4 for training, cycle 5 for testing, and cycle 6 for validation. This change necessitated a redo of the hyperparameter tuning process. Thus, the new combination that generated the best learning was: 256 neurons; 500 epochs; a learning rate of 0.01; and a weight decay of 0.01. After this adjustment, the network was ready to be trained for the experiment.

3.3.3 Brute-Force Optimization

The objective of this optimization is to identify the combination of IT-Off, EC, and pH that maximizes lettuce production in terms of FMM. The brute-force optimization algorithm is applied to each channel and involves several steps: generating possible combinations for the inputs; testing the combinations to obtain predictions; grouping by parameter combination; calculating the average of the predictions for each group; and selecting the highest average.

The generation of combinations is represented in Figure 3.8. First, it is necessary to get the combination of IT-Off, EC, and pH varying the normalized values using the combination tree as specified in Figure 2.2. To generate those combinations it was used a step of 0.125. This step size is chosen as it is small enough to thoroughly explore the model without necessitating an excessive number of tests. Second, it is necessary to get all the combinations that are fixed, and that is the case of channel, position and PPFD. Finally, the two sets are combined.

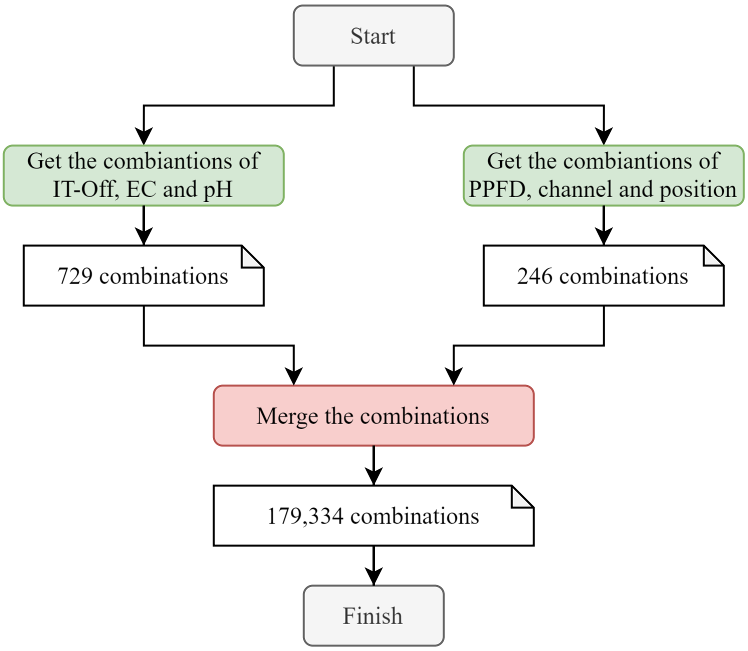


Figure 3.8: Brute-force inputs generation process.

These combinations are then tested in the model generated by the ANN to obtain the respective predictions. After that, the data is separated by channel and grouped by IT-Off, EC, and pH. Then, each group has individuals of a channel. With that, it is calculated the average production of the channel. Finally, by sorting these averages, the highest one and its corresponding combination are identified.

3.3.4 Validation Plantings

After identifying the parameters that generate the highest production in each channel, plantings of these combinations were carried out to verify if what was predicted could be confirmed. Cycles 7 and 8 were carried out according to Table 3.5, where the optimal parameter combinations found for each channel were applied, except for channel 1, which remained as a standard sample, receiving the same combinations applied in the exploratory plantings. In both validation plantings, the same combinations were applied to observe the consistency of the results.

It is worth remembering that the optimal values found by the optimization process, on the model generated by the ANN, are normalized values according to the highest average production achieved observed in the standard sample. Therefore, the real value to be harvested must also be normalized to be compared, and thus know if the real value confirms the expected value.

Chapter 4

Results and Discussion

This chapter delineates the outcomes of each stage of the research as specified in section 3.3. The research process was structured such that the exploration plantings phase yielded a dataset for training the ANN; the ANN training phase generated a regression model which is the cost function for the brute-force optimization; the brute-force optimization phase produced a set of optimum combinations for the validation plantings; and, the validation plantings phase provided the final result which is the key to answer the question: Can indoor hydroponic lettuce production be optimized using the chosen approach? The subsequent sections present the results of each phase.

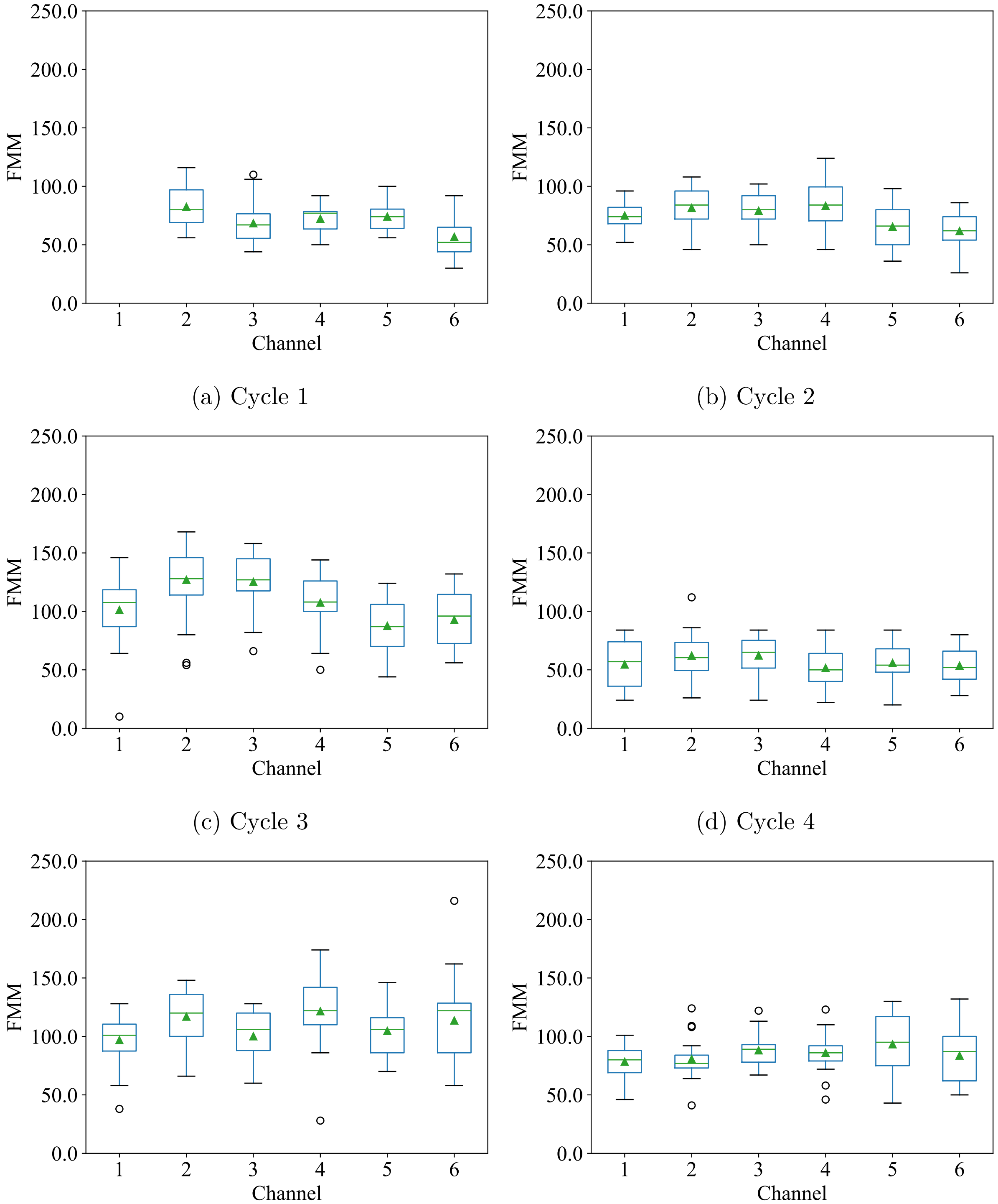
4.1 Exploration Plantings

The exploration plantings were carried out according to the values established in Table 3.6 following the schedule defined in subsection 3.2.2. Despite some problems being identified, in the end it was possible to obtain the desired dataset. The results obtained can be verified in Table 4.1 and in Figure 4.1.

During the first cycle, there was a failure in the irrigation system of channel 1, resulting in the loss of all plants in that channel, which was of particular importance, as it contained the standard samples. In addition, of the remaining 110 plants, 19 died for unknown reasons. And, the FMM values of the harvest presented a high Standard Deviation (SD).

Table 4.1: Exploratory plantings results.

Cy. / Ch.	Count	Min. (g)	Max. (g)	Mean (g)	SD (g)
1 / 1	-	-	-	-	-
1 / 2	19	56	116	82.42	18.29
1 / 3	16	44	110	68.38	18.88
1 / 4	16	50	92	72.25	11.26
1 / 5	20	56	100	74.10	12.90
1 / 6	19	30	92	56.74	16.92
2 / 1	21	12	96	74.95	12.32
2 / 2	21	16	108	81.52	15.75
2 / 3	21	16	102	79.05	16.13
2 / 4	21	21	124	84.86	20.76
2 / 5	21	19	98	65.43	19.01
2 / 6	21	16	86	61.71	16.08
3 / 1	20	10	146	101.15	30.72
3 / 2	21	54	168	126.95	31.70
3 / 3	20	66	158	125.20	24.98
3 / 4	21	50	144	107.43	24.24
3 / 5	18	44	124	87.56	23.71
3 / 6	20	56	132	92.60	24.19
4 / 1	20	24	84	54.50	18.96
4 / 2	20	26	112	62.10	19.91
4 / 3	20	24	84	62.25	17.07
4 / 4	20	22	84	51.60	18.73
4 / 5	21	20	84	55.81	19.31
4 / 6	18	28	80	53.56	15.53
5 / 1	20	38	128	96.80	21.87
5 / 2	20	66	148	116.90	23.68
5 / 3	21	60	128	100.10	22.50
5 / 4	21	28	174	121.52	30.81
5 / 5	21	70	146	104.67	24.35
5 / 6	20	58	216	113.60	37.61
6 / 1	21	46	101	78.33	14.92
6 / 2	21	41	124	80.57	17.32
6 / 3	21	67	122	88.10	13.88
6 / 4	21	46	123	85.95	16.31
6 / 5	21	43	130	93.24	25.72
6 / 6	21	50	132	83.57	23.28



(a) Cycle 1 (b) Cycle 2

(c) Cycle 3 (d) Cycle 4

(e) Cycle 5 (f) Cycle 6

Figure 4.1: Exploratory plantings boxplots.

After the first cycle, the irrigation was reestablished and worked correctly until the end of the fifth cycle, when it was necessary to take a two-week break to carry out maintenance procedures, having also worked correctly after that. From the second cycle, the plants were transplanted in an interleaved manner along the channels, providing a greater spacing between them. This greater spacing may have contributed to the decrease in the occurrence of deaths.

The variation of FMM continued to be observed in the subsequent cycles. The hypothesis is that proximity to light points favors some plants to the detriment of others. This is because there is not a light point for each planting position. And, the plants that are closest to the light points tend to receive a greater incidence of light, as illustrated in Figure 4.2. The subsection 2.2.1 presents the fact that plants need a certain value of light to properly carry out the photosynthesis process. Therefore, such variations may have contributed to the variation of FMM observed.

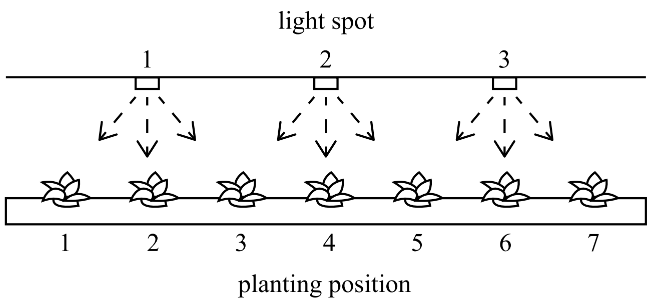
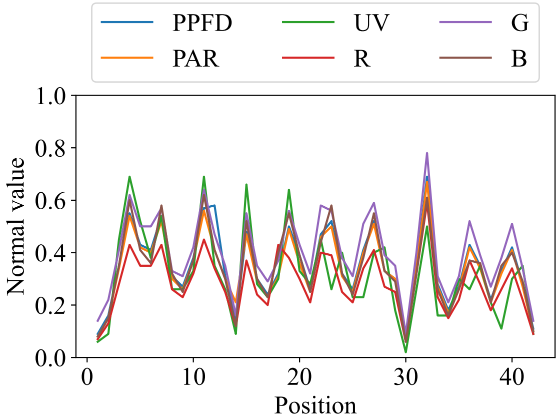
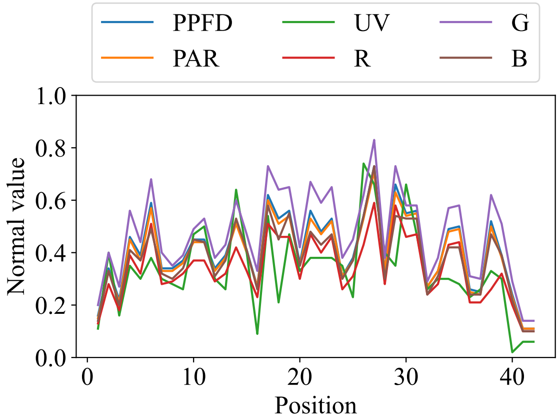


Figure 4.2: Light coverage.

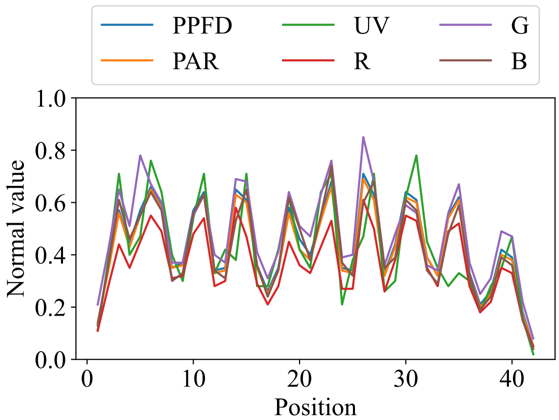
Because of this, measurements of the parameters related to lighting were carried out to be considered in the training of the ANN. The measurement was carried out by recording the values of PAR, PPFD, R, G, B, and UV for each planting point, at a height of 3 cm in relation to the channel. In general, the measured values were very close after being normalized, however, some disparities can be noticed, as shown in Figure 4.3. Also, it is noticed that there is less light intensity in the last positions of the gutters, and Figure 4.4 shows this about the PAR values.



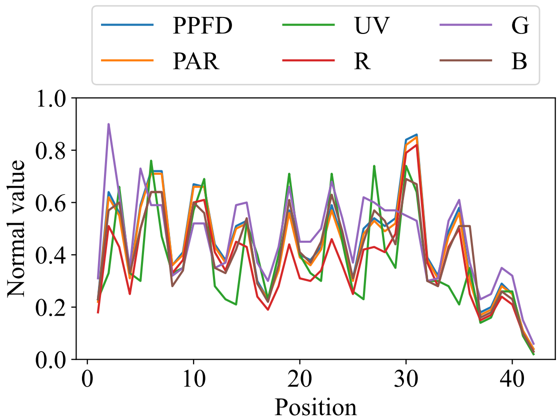
(a) Channel 1



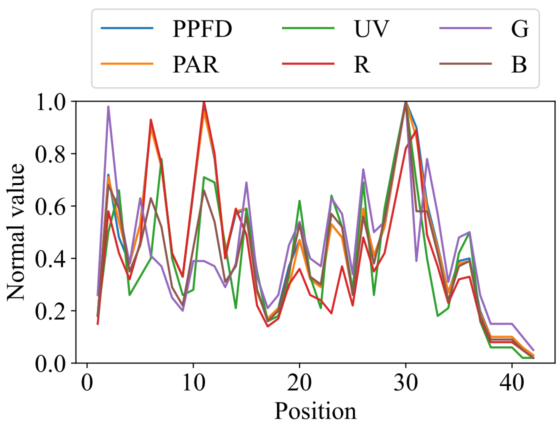
(b) Channel 2



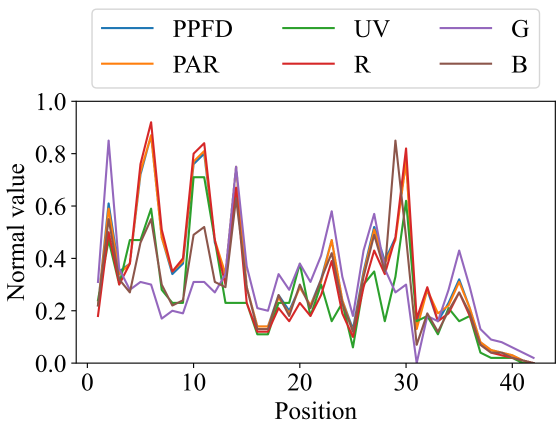
(c) Channel 3



(d) Cannel 4



(e) Cannel 5



(f) Cannel 6

Figure 4.3: Normalized light parameters by channel.

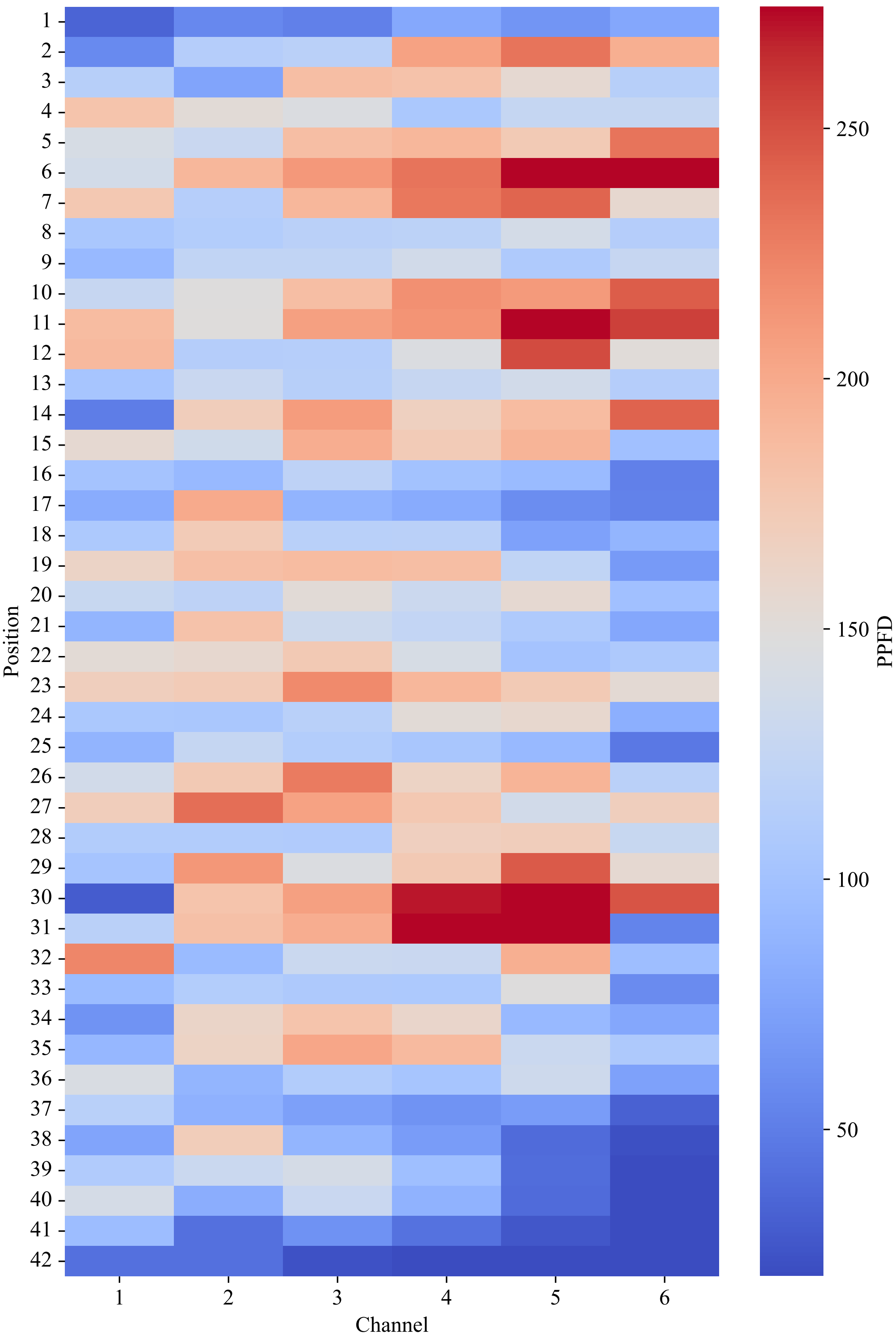


Figure 4.4: Light coverage per channel and position.

Another important point to observe is the variation in the results of the averages of the standard samples over the cycles. As discussed in subsection 3.2.2, it was defined that the first channel of each cycle would receive the same parameters to check for any possible variation between one cycle and another. Although it was not expected, as can be briefly observed in Figure 4.5, such variations occurred. And, although the real reason for this occurrence is not certain, it is suspected that the uncontrolled parameters such as RH and CO_2 may have influenced. Despite the obtained values having undergone unexpected variations, this stage resulted in a dataset composed of 704 rows.

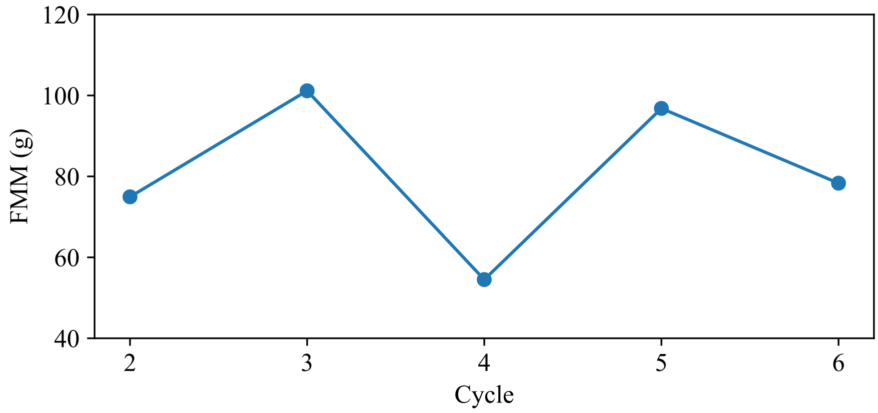


Figure 4.5: Standard samples results.

4.2 Artificial Neural Network Training

With the data obtained from the 6 exploratory cycles, as established in subsection 3.3.2, the hyperparameter tuning process was carried out to define the network configuration, in addition, the feature selection process was carried out to define which features would be used. At the end of the process, it is recapitulated that the network was configured with 5 input features (PPFD, channel, position, IT-Off, EC, and pH), 256 neurons in an intermediate layer, and a single output. And, 500 epochs for training, learning rate of 0.01, and weight decay of 0.01.

The generated dataset underwent the data engineering process, and the description of the data can be observed in Table 4.2. Through the mean, it is possible to observe that the values tend to be positioned in the middle of the normal range from 0 to 1, with the exception of the result, which was normalized between 0 and the highest average production obtained in the control channel, 101.15 g. The minimum, maximum and quartile values also show a balanced distribution of values. The Figure 4.6 shows the correlation between the dataset features.

Table 4.2: Data description.

Feature	**Count**	**Mean**	**SD**	**Min.**	**25%**	**50%**	**75%**	**Max.**
PPFD	704	0.42	0.19	0.00	0.30	0.39	0.56	1.00
Channel	704	0.51	0.33	0.00	0.20	0.60	0.80	1.00
Position	704	0.44	0.28	0.00	0.21	0.43	0.69	0.98
IT-Off	704	0.54	0.28	0.04	0.30	0.60	0.77	0.98
EC	704	0.54	0.23	0.05	0.38	0.50	0.75	0.97
pH	704	0.51	0.22	0.03	0.38	0.50	0.67	0.86
FMM	704	107.92	29.22	10.00	88.00	110.00	127.00	216.00

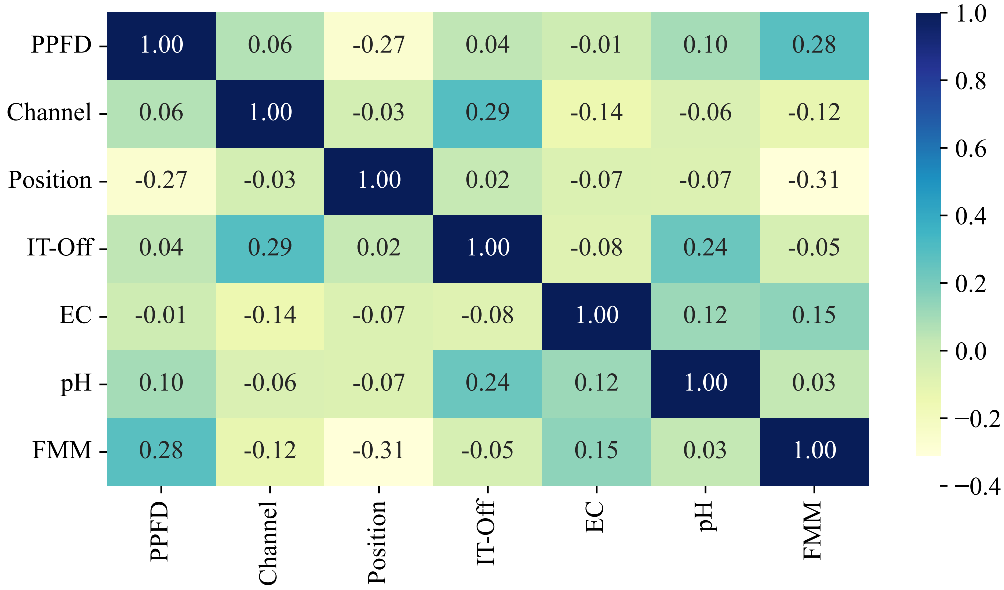


Figure 4.6: Feature correlation.

From Figure 4.6 it is observed that there is a greater correlation between FMM and PPFD values. Furthermore, it is observed that EC has a greater correlation with FMM than IT-Off and pH, with the latter two being quite similar and close to zero. Finally, it is observed that there is a negative correlation between channel and position in relation to FMM.

After the analysis was conducted, the dataset was split as defined in subsection 3.3.2, and the training of the ANN proceeded. As a result, a model was obtained whose MAE, MSE, and RMSE values obtained in the test were, respectively, 21.35g, 815.76g^2, and 28.56g. The training and validation graph can be visualized through Figure 4.7. The graph shows a very rapid reduction of loss values up to epoch 100, decreasing from 5,000 to less than 1,000, maintaining this value stable until the end of the training.

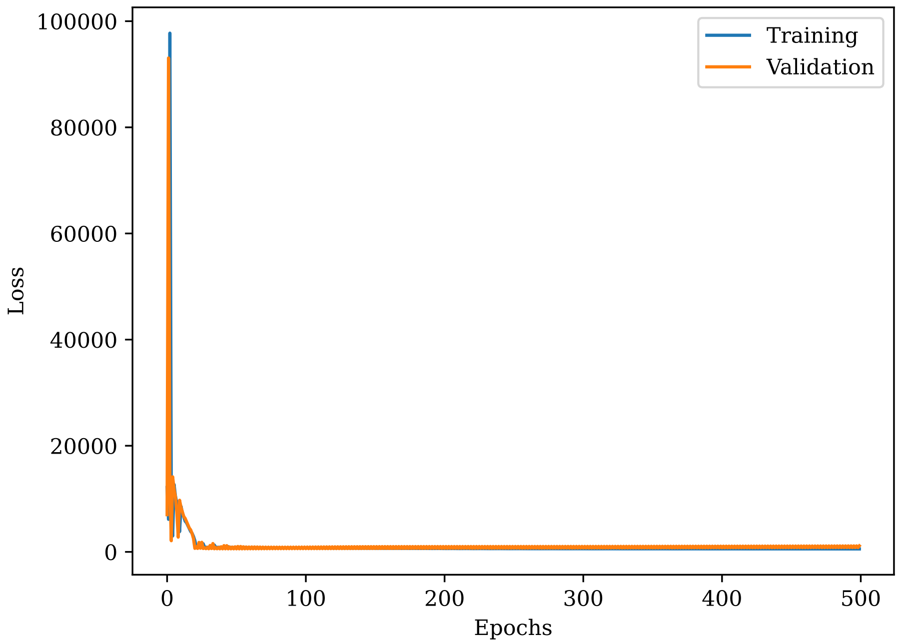


Figure 4.7: Training validation curves.

4.3 Brute-Force Optimization

After the training was completed, the brute-force optimization was carried out using the generated regression model as a cost function, as defined in subsection 3.3.3. The Table 4.3 presents the optimal values found in the normalized scale. Through the real values, it is possible to observe that the optimal combinations are quite different from one channel to another. It is observed that the values of IT-Off were intermediate values, variating from 0.38 to 0.75. And the values of EC and pH varied from one extreme to the other, from 1 to 0. The real values are presented in Table 4.4. The highest value is found in the channel 3, with 122.19g of FMM.

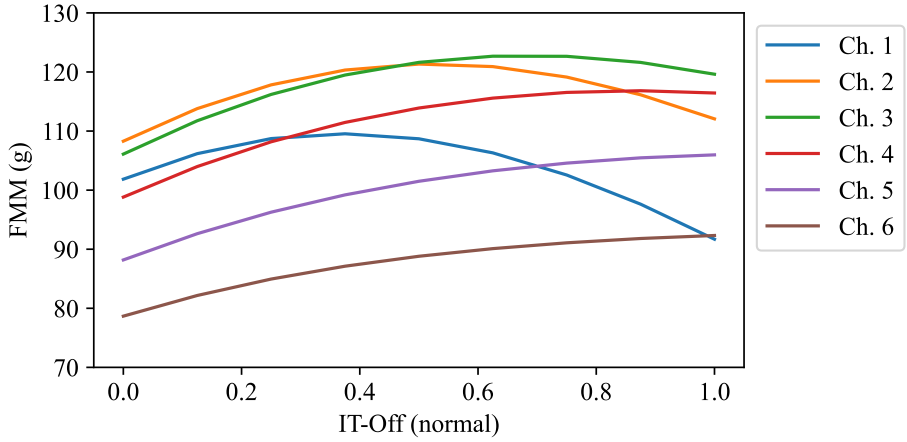
Table 4.3: Optimum normal values.

Ch.	**IT-Off (normal)**	**EC (normal)**	**pH (normal)**	**Prediction (g)**
1	0.50	1.00	0.75	104.75
2	0.38	1.00	0.50	121.95
3	0.38	0.62	0.25	122.19
4	0.38	0.38	0.00	118.98
5	0.62	0.00	0.00	112.64
6	0.75	0.00	0.00	104.83

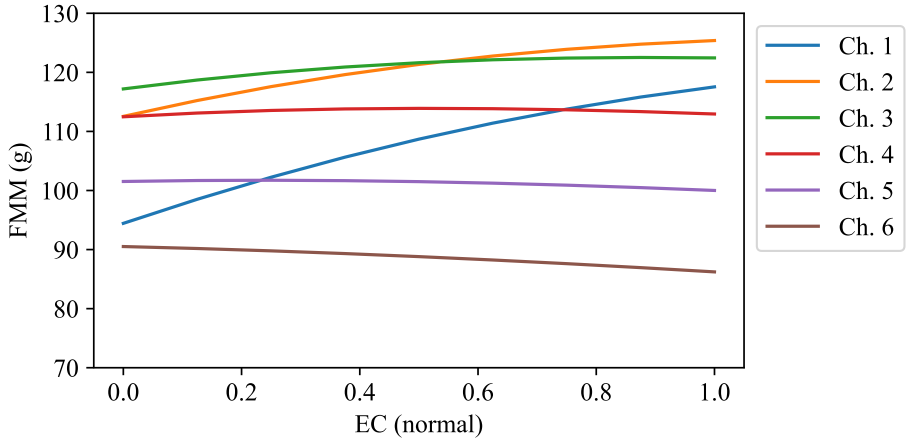
Table 4.4: Optimum real values.

Ch.	**IT-Off (min)**	**EC (mS/cm)**	**pH**	**Prediction (g)**
1	30	2.2	6.3	104.75
2	25	2.2	6.0	121.95
3	25	1.7	5.8	122.19
4	25	1.5	5.5	118.98
5	35	1.0	5.5	112.64
6	40	1.0	5.5	104.83

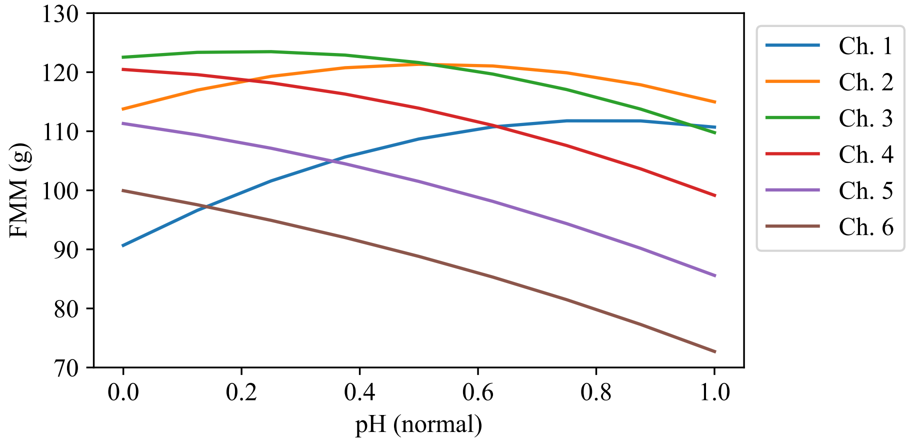
As the brute-force optimization tests a large number of combinations to find the optimal values, it is possible to generate graphs that show the gain of FMM according to the variation of the parameters. The Figure 4.8 shows this progression in each of the channels. To generate these graphs, one parameter varies while the others are fixed at half of the range (normal value equal to 0.5).



(a) Irrigation time off



(b) Eletric conductivity



(c) Hydrogen potential

Figure 4.8: Parameter curves by channel.

It is important to highlight that the gain of FMM occurs under the condition that the constant parameters remain as presented in subsection 3.2.1. In addition, each line represents the average of the plants through a specific channel. Among the channels, it is observed that first one presents a different pattern when analyzing the three parameters. This distinct behavior is probably due to the fact that channel 1 was used for the cultivation of the standard samples in all cycles. Therefore, as it was not properly explored like the others, it is possible that the network did not correctly learn the growth pattern in this channel. Although there is a gap in between the other channels, they show a similar behavior. The greater the IT-Off and the lower the pH, the greater the gain of FMM. The EC does not have much impact on the gain of FMM compared to the others parameters. This information suggest that its range could be extended for both sides.

It is also possible to analyze in general what the productivity of the MHSC would be if the same combination of parameters were applied to all channels. To do this, for each combination the average of the FMM results is calculated. The new set of values allow the generation of the Figure 4.9. This voxel type graph is composed of a three-dimensional grid of points. Each point in the graph corresponds to a unique combination of IT-Off, EC, and pH, which are mapped to the X, Y, and Z positions respectively. The colors represent the gain of FMM. The maximum and the minimum value of FMM are, respectively, 113.89g and 68.3g. The normalized and real values of the combinations of IT-Off, EC, and pH that generated the maximum and minimum values of FMM can be seen, respectively in Table 4.5 and Table 4.6.

The voxel shows that the maximum value is located in one of the edges, where IT-Off, EC, and pH are, respectively, 0.38, 1, 0. When the best value is located on the extremity of a range that means this value could not be the optimal, and a better one could be found if the range is extended. Therefore, the minimum value of pH could be reduced, while the maximum value of EC could be increased. Also, the worst value is located in one of the vertices, where IT-Off, EC, and pH are, respectively, 0, 0, 1. This suggests that the least optimal scenario involves irrigating the lettuces for the shortest time, with a solution that has the lowest nutrient amount and the highest pH.

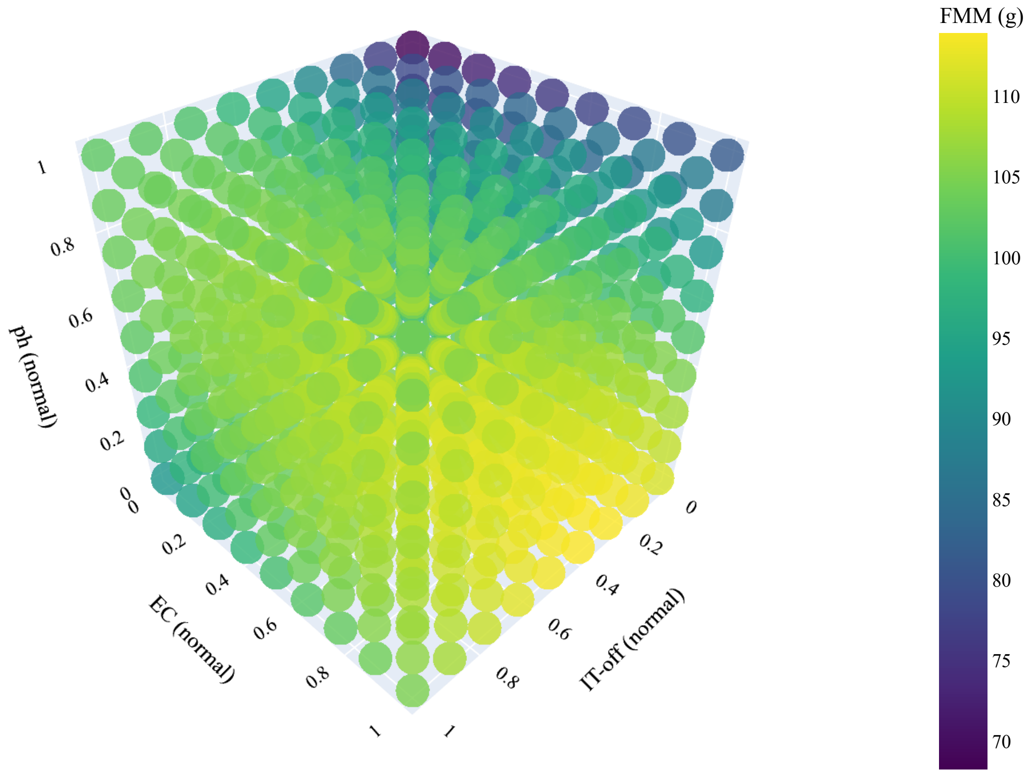


Figure 4.9: Parameters voxel.

Table 4.5: Maximum and minimum general normal values.

Result	IT-Off (normal)	EC (normal)	pH (normal)	FMM (g)
Maximum	0.38	1.00	0.00	113.89
Minimum	0.00	0.00	1.00	68.27

Table 4.6: Maximum and minimum general real values.

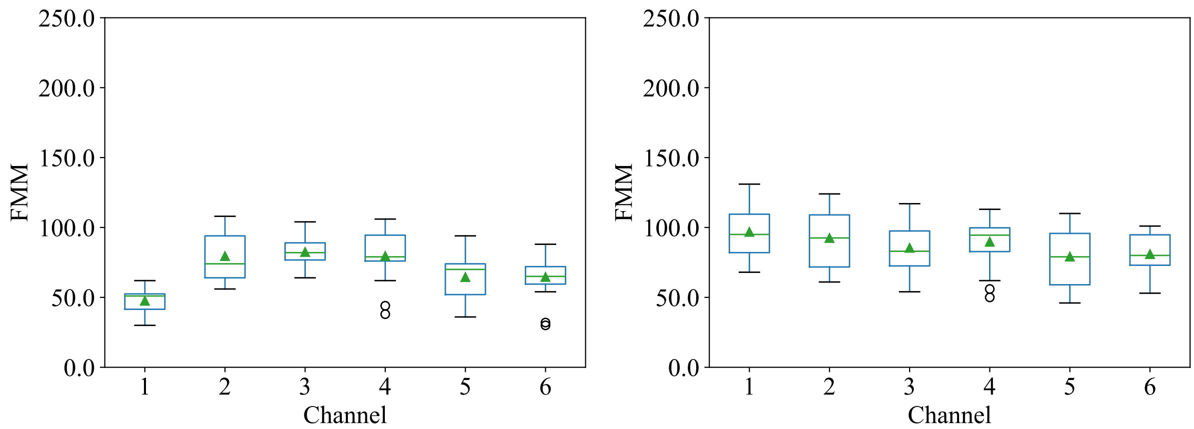
Result	IT-Off (min)	EC (mS/cm)	pH	FMM (g)
Best	25.0	2.2	5.5	113.89
Worst	10.0	1.0	6.5	68.27

4.4 Validation Plantings

The two validation plantings were carried out, the cycle 7 and 8. Those plantings were cultivated using the optimal values found, presented in Table 4.4. The results of the two plantings are presented in Table 4.7 and Figure 4.10. The results show that in cycle 7, the standard samples (channel 1) had a lower average than the other channels. However, in cycle 8, the standard samples had a higher average compared to the other channels.

Table 4.7: Validation plantings results.

Cy. / Ch.	Count	Min. (g)	Max. (g)	Mean (g)	SD (g)
7 / 1	20	30.00	62	47.55	8.65
7 / 2	21	56.00	108	79.52	16.96
7 / 3	20	64.00	104	82.45	10.30
7 / 4	20	38.00	106	79.60	17.68
7 / 5	21	36.00	94	64.48	16.43
7 / 6	20	30.00	88	64.60	14.84
8 / 1	19	68.00	131	96.84	18.29
8 / 2	20	61.00	124	92.45	19.95
8 / 3	19	54.00	117	85.42	17.41
8 / 4	20	50.00	113	89.75	17.86
8 / 5	20	46.00	110	79.15	21.08
8 / 6	18	53.00	101	80.94	15.85



(a) Cycle 7

(b) Cycle 8

Figure 4.10: Validation plantings boxplots.

The Table 4.8 analyzes the normal mean compared to the predicted values and the errors. Despite cycle 7 presenting a relatively low average for the standard samples, these values, when normalized, exceed the predicted value, thereby exceeding expectations. On the other hand, even though cycle 8 has a relatively high average for the standard samples, which was desired, it falls below the expected value when the values are normalized, thus not meeting expectations.

Table 4.8: Error per channel.

Cy. / Ch.	**Normal (g)**	**Predict (g)**	**MAE (g)**	**MSE (g^2)**	**RMSE (g)**
7 / 1	104.74	115.22	21.57	776.93	27.87
7 / 2	175.16	121.95	53.11	4,038.13	63.55
7 / 3	181.61	122.19	58.04	3,949.88	62.85
7 / 4	175.33	118.98	59.58	4,085.53	63.92
7 / 5	142.02	112.64	34.97	1,918.81	43.80
7 / 6	142.29	104.83	38.36	2,119.61	46.04
8 / 1	104.74	115.22	7.90	64.47	8.03
8 / 2	99.99	121.95	7.54	59.35	7.70
8 / 3	92.39	122.19	6.97	50.43	7.10
8 / 4	97.07	118.98	7.32	55.58	7.46
8 / 5	85.60	112.64	6.45	44.47	6.67
8 / 6	93.01	104.83	6.60	45.15	6.72

Another important aspect to be observed is that the model errors for MAE, MSE, and RMSE were 21.35g, 815.76g^2, and 28.56g respectively. As shown in Table 4.9, in general, the error in cycle 7 was more than twice as high as the model's error, while the error in cycle 8 was less than half of the model's error. In other words, despite the first validation planting yielding superior results, the error was larger than expected. Conversely, even though the second validation planting yielded inferior results, the error was within the expected range.

Table 4.9: Error per cycle.

Cy.	**Real (g)**	**Normal (g)**	**Predict (g)**	**MAE (g)**	**MSE (g^2)**	**RMSE (g)**
7	69.70	153.55	114.22	44.27	2,814.81	51.34
8	87.43	94.57	114.22	7.13	53.24	7.28

Finally, the question posed at the beginning of this chapter can be addressed. The Table 4.10 presents the average normal result per cycle. The cycle 7 yielded a superior result (153.55g) compared to all previous plantings, whereas cycle 8 yielded an inferior result (94.57g). Thus, it is indeed possible to optimize indoor hydroponic lettuce production using the chosen technique. However, due to the divergent results of the cycles 7 and 8, further experimentation is necessary to reinforce these findings.

Table 4.10: Mean normal result per cycle.

Cycle	**Mean normal (g)**
1	98.91
2	104.24
3	110.62
4	108.86
5	117.88
6	113.61
7	153.55
8	94.57

Chapter 5

Conclusions

This work aimed to apply a ML technique to optimize indoor hydroponic lettuce production. To achieve this goal, five specific objectives were defined: establish the goal and parameters of optimization; select the ML technique to find the optimal combination of parameters; implement and execute a computational system for the chosen ML technique; define the planting strategy and procedures; conduct exploratory and validation plantings. After theoretical study and analysis of available materials, the optimization goal and parameters, planting schedule, and ML approach to be applied were established. The chosen optimization goal was the maximization of lettuce's FMM, and the selected optimization parameters were IT-Off, EC, and pH. The defined schedule allowed for eight plantings. The chosen ML approach involved the use of a ANN combined with the application of a brute-force optimization algorithm. To apply this approach, a computational system was developed in the Google Colab development environment, using the Python language and the PyTorch library.

To apply the ML approach, the following steps were performed: conducting six exploratory plantings; training a neural network; optimization with a brute-force algorithm; conducting two validation plantings. The exploratory plantings resulted in a dataset of 704 rows with information related to the chosen parameters and additional location and lighting data. The training of the ANN resulted in a regression model, with MAE, MSE, and RMSE metrics of 21.35g, 815.76g^2, and 28.56g, respectively. The application of the brute-force algorithm identified the optimal combinations individually (Table 4.4) and globally (Table 4.6). Two validation plantings were conducted to verify if the predicted individual optima would be confirmed in practice, the first yielded results that were on average 34.43% above the predicted values, and the second yielded results that were on average 17.20% below the predicted values. Despite the first cycle exceeding the predicted values, the error was greater than the model's error.

From these results, it can be concluded that all specific objectives were achieved. Therefore, the general objective of applying a ML technique to optimize indoor hydroponic lettuce production was also achieved. Finally, it is still necessary to determine whether the process optimized production or not. The optimization process predicted the optimal production for each of the channels, with the average of these values being 114.22g. This value is higher than the normalized average presented in the results of 5 of the 6 exploratory plantings, as shown in Table 4.10. It was expected that the validation plantings would present results similar to the predicted average, however, the obtained values were divergent. The first validation planting (cycle 7) resulted in a normalized average of 153.55g, higher than all other plantings. And, the second validation planting (cycle 8) presented a normalized average result of 94.57g, lower than all plantings. Considering the first validation planting, it can be concluded that the process optimized production. However, the divergent result of the second validation planting necessitates further experiments to verify the findings.

Despite not being considered initially, the instability observed throughout the plantings was the main limitation of this work. This limitation was also observed in the study by Sparks (2018)[3], as discussed in subsection 2.5.3. As living organisms, it is natural that they are not entirely identical, however, some measures can be taken to try to make the results more stable. Greater control over the parameters is something to consider. For instance, as can be seen in Table 3.1, due to the limitations of the MHSC, CO_2 was not kept constant. And, as presented in subsection 2.2.1, this parameter significantly influences the photosynthesis process of plants, interfering with their growth. Regardless, it is important to be aware of the degree of instability. This could be obtained by conducting successive plantings with the same configuration to understand the margin of error. Thus, it would be possible to be certain whether the optimization process worked within this margin. Therefore, such measures are crucial for the general objective to be successfully achieved in a similar work.

5.1 Final Considerations

This work demonstrated that the employed ML technique is functional for the given scenario, reinforcing the studies of Villarrubia *et al.* (2018)[6] and Morimoto *et al.* (1993)[7]. Considering the great variability presented by the results between the cycles, it is observed that an evolutionary algorithm would hardly converge in the scenario defined for this study, taking into account the time and materials available. Although the chosen technique was suitable, more studies are necessary to validate the results. If confirmed, some interesting characteristics found can be considered by producers in indoor hydroponic lettuce cultivation.

Observing the graphs of Figure 4.8 and Figure 4.9, and the intervals defined in Table 3.4, it is noticed that median values of IT-Off (25 minutes), low values of pH (5.5), and high values of EC (2.2 mS/cm) contribute to a greater gain of FMM (113.89g). The optimal values found for EC and pH differ from the values suggested by the bibliography, as suggested in subsection 2.2.1. The minimum and maximum suggested values are between 1.15 mS/cm and 1.25 mS/cm for EC, and between 5.6 and 6 for pH. On the other hand, the optimal value for IT-Off is among the values analyzed by the studies of Soares (2020)[18] and Zanella *et al.* (2008)[19].

5.2 Future Works

A need for greater stability in the results between cycles is observed, and this can be achieved through more extensive control over the parameters. The Table 3.1 shows which parameters could not be fully controlled in this study, notably RH and CO_2. Besides these, there may be others that should be kept constant. For example, the value of FMM of the seedlings at the time of transplanting was not considered. Although the nursery time was the same, some interference during this period may have contributed to an irregular final result. And, considering that genetic characteristics can interfere with plant development, even cloning them can be considered.

Another point to consider is the variability between the results of individuals along the channels. It is seen that the distribution of light spots was one of the factors that influenced this variation, therefore, smoothing the lighting can help to standardize the development of the plants in the same cycle. This could be achieved through the installation of a light spot for each planting position, or by the use of light diffusers. Despite the confirmed variability of lighting, the existence of other non-uniform parameters should also be considered.

Understanding the degree of variability between cycles and along the channels is crucial. To achieve this, repeated plantings under the same parameters can be conducted. The differences in the results can then be compared to define the margin of error. Implementing an automated parameter monitoring system would be beneficial for this process. This system could generate a report detailing the behavior of the parameters throughout the cycles and in different areas within the MHSC. This report could provide valuable insights into the causes of the variations, enabling their correction.

As depicted in Figure 4.9, the optimal value generally lies at the extremes of EC and pH. Therefore, it may be beneficial to expand these ranges to check if a better value could be found. Then, the values in Table 3.4 could be adjusted by increasing the maximum value of EC and decreasing the minimum value of pH.

Finally, due to the divergence in the validation plantings, additional validation could be conducted using the configuration outlined in section 4.4. This may further substantiate the conclusions of this study.

Bibliography

[1] E. F. de Lima Barros, "Avaliação de variedades de alface sob cultivo hidropônico", Universidade Federal Fluminense, Ponta Grossa, 2021.

[2] O. A. Carrijo and N. Makishima, "Princípios de hidroponia", *Embrapa*, vol. 22, 2000.

[3] R. E. Sparks, "Design and testing of a modified hydroponic shipping container system for urban food production", *International Journal of Applied Agricultural Sciences*, vol. 4, p. 93, 4 2018.

[4] C. A. Mitchell, "History of controlled environment horticulture: Indoor farming and its key technologies", vol. 57, American Society for Horticultural Science, Feb. 2022, pp. 247–256.

[5] A. D. Pila, "História e terminologia a respeito da computação evolutiva", *Ciências Exatas e Tecnologia*, vol. 1, no. 1, pp. 40–50, 2015.

[6] G. Villarrubia, J. F. D. Paz, P. Chamoso, and F. D. la Prieta, "Artificial neural networks used in optimization problems", *Neurocomputing*, vol. 272, pp. 10–16, Jan. 2018.

[7] T. Morimoto, T. Takeuchi, and Y. Hashimto, *Growth optimization of plant by means of the hybrid system of genfilc algorithm and neural network*, 1993.

[8] C. A. Mitchell, *Plant lighting in controlled environments for space and earth applications*, 2012.

[9] H. M. Resh, *Hydroponic Food Production, A Definitive Guidebook for the Advanced Home Gardener and the Commercial Hydroponic Grower*. New York, USA: CRC Press Taylor & Francis Group, Apr. 2013.

[10] M. H. Jensen, “Hydroponics”, *HortScience*, vol. 32, no. 6, pp. 1018–1021, 1997.

[11] T. C. Hoagland D. R. & Broyer, “General nature of the process of salt accumu-lation by roots with description of experimental methods”, *Plant Physiology*, vol. 11, no. 3, pp. 471–507, 1983.

[12] G. I. O. P. d. S. Torres, “Solução nutritiva: Avaliação da reposição de macronutrientes no cultivo hidropônico de alface”, Universidade Federal Rural da Amazônia, Belém, Feb. 2009.

[13] M. J. de Araujo Faria Junior and R. C. da Hora, “Cultivo protegido”, *Hortaliças-fruto*, pp. 451–487, 2018.

[14] I. C. D. Moura and S. P. D. Carvalho, “Avaliação da qualidade pós-colheita e da perceção do consumidor de alfaces colhidas e comercializadas com raiz”, 2017.

[15] C. B. Shear, “Calcium related disorders of fruits and vegetables”, *Horticultural Science*, vol. 10, pp. 361–365, 1975.

[16] M. Brechner and A. J. Both, *Hydroponic Lettuce Handbook*. Ithaca, US: Cornel University, Jan. 2013.

[17] C. K. Keat and C. Kannan, “Development of a cylindrical hydroponics system for vertical farming”, *Journal of Agricultural Science and Technology B*, vol. 5, pp. 93–100, 2015.

[18] C. S. Soares, A. C. S. de Lima, J. A. da Silva, *et al.*, “Production of lettuce in nft hydroponic system at different planting seasons and irrigation regimes”, *Australian Journal of Crop Science*, vol. 14, pp. 1042–1047, 7 Jul. 2020.

[19] F. Zanella, A. L. da Silva Lima, F. F. da Silva Júnior, and S. P. A. Maciel, *Hydroponic lettuce growth under differents irrigation intervals*, 2008.

[20] J. H. Holland and D. E. Goldberg, “Genetic algorithms and machine learning”, *Machine Learning*, vol. 3, no. 1, pp. 95–99, 1988.

[21] R. L. Haupt, *Practical Genetic Algorithms*. New Jersey, USA: Wiley, Jan. 2004.

[22] M. Mitchell, “Life and evolution in computers”, *Santa Fe Institute Working Paper*, vol. 1, no. 1, pp. 1–21, 2000.

[23] R. C. Read and R. E. Tar, “Bounds on backtrack algorithms for listing cycles, paths, and spanning trees”, *Networks*, vol. 5, pp. 237–252, 1975.

[24] K. H. Rosen, *Discrete mathematics and its applications*. 2012, vol. 7.

[25] R. W. Floyd, “Nondeterministic algorithms”, *Journal of the Association for Computing Machinery*, vol. 14, no. 4, pp. 636–644, 1967.

[26] C. Darwin, *A origem das Espécies*. London, England: Planeta Vivo, Nov. 1859.

[27] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Michigan, USA: The University of Michigan Press, Jan. 1975.

[28] M. Melanie, *An introduction to genetic algorithms*. London, England: Cambridge, Jan. 1996.

[29] L. N. D. Castro and F. J. V. Zuben, “An evolutionary immune network for data clustering”, *Brazilian Symposium on Artificial Neural Networks*, pp. 22–25, 2000.

[30] L. N. D. Castro and F. J. V. Zuben, “Application of bio-inspired metaheuristics in the data clustering problem”, *Clei Eletronic Journal*, vol. 14, no. 3, pp. 1–18, 2011.

[31] S. Russell and P. Norving, *Artificial Intelligence: A Modern Approach*. 2010, vol. 3, p. 1132.

[32] J. Fürnkranz, P. K. Chan, S. Craw, and C. Sammut, *Encyclopedia of machine learning*, 2010.

[33] I. Valenzuela, R. Baldovino, A. Bandala, and E. Dadios, “Pre-harvest factors optimization using genetic algorithm for lettuce”, vol. 10, pp. 159–163, 1-4 2018.

[34] V. Tessele, “Aplicação de algoritmo bio-inspirado para refinamento de recomendação de adubação para área específica”, Universidade Tecnológica Federal do Paraná, Medianeira, 2018.

[35] J. Bergstra, J. B. Ca, and Y. B. Ca, *Random search for hyper-parameter optimization yoshua bengio*, 2012. [Online]. Available: `http://scikit-learn.sourceforge.net.`.

[36] F. Bafort, S. Kohnen, E. Maron, *et al.*, “The agro-economic feasibility of growing the medicinal plant euphorbia peplus in a modified vertical hydroponic shipping container”, *Horticulturae*, vol. 8, 3 Mar. 2022.

[37] Sakata Seed, *Alface valentina*, https://www.sakata.com.br/hortalicas /folhosas/alface/crespa/valentina, Access date: April 6th, 2024.

[38] Python Software Foundation, *Python*, https://www.python.org/, Access date: April 6th, 2024.

[39] Google, *Google colaboratory*, https://colab.google/, Access date: April 6th, 2024.

[40] Meta AI, *Pytorch*, https://pytorch.org/, Access date: April 6th, 2024.

[41] D. Hendrycks and K. Gimpel, “Gaussian error linear units (gelus)”, Jun. 2016.

[42] A. Zhang, Z. C. Lipton, M. U. Li, and A. J. Smola, *Dive into deep learning*, 2012.