DEEP LEARNING AND FEATURE ENGINEERING TECHNIQUES APPLIED TO THE MYOELECTRIC SIGNAL FOR ACCURATE PREDICTION OF MOVEMENTS

GABRIEL CIRAC MENDES SOUZA

DOCTORAL THESIS IN ELECTRICAL ENGINEERING

INSTITUTE OF SYSTEMS ENGINEERING AND INFORMATION TECHNOLOGY

FEDERAL UNIVERSITY OF ITAJUBÁ

FEDERAL UNIVERSITY OF ITAJUBÁ
INSTITUTE OF SYSTEMS ENGINEERING AND INFORMATION TECHNOLOGY

DEEP LEARNING AND FEATURE ENGINEERING TECHNIQUES APPLIED TO THE MYOELECTRIC SIGNAL FOR ACCURATE PREDICTION OF MOVEMENTS

GABRIEL CIRAC MENDES SOUZA

ADVISOR: ROBSON LUIZ MORENO
CO-ADVISOR: TALES CLEBER PIMENTA

DOCTORAL THESIS IN ELECTRICAL ENGINEERING

ITAJUBÁ/MG: JULY - 2021

FEDERAL UNIVERSITY OF ITAJUBÁ

INSTITUTE OF SYSTEMS ENGINEERING AND INFORMATION TECHNOLOGY

DEEP LEARNING AND FEATURE ENGINEERING TECHNIQUES APPLIED TO THE MYOELECTRIC SIGNAL FOR ACCURATE PREDICTION OF MOVEMENTS

GABRIEL CIRAC MENDES SOUZA

DOCTORAL THESIS SUBMITTED TO THE INSTITUTE OF SYSTEMS ENGINEERING AND INFORMATION TECHNOLOGY OF THE FEDERAL UNIVERSITY OF ITAJUBÁ AS PART OF THE REQUIREMENTS TO OBTAIN THE GRADE OF PHD.

APPROVED BY:

____

PhD. Robson Luiz Moreno – UNIFEI/IESTI
Advisor

____

PhD. Tales Cleber Pimenta – UNIFEI/IESTI
Co-advisor

____

PhD. Joel José Puga Coelho Rodrigues – UFPI
External Member

____

Alexandre Mello Ferreira – UNICAMP
External Member

____

PhD. Gabriel Antonio Fanelli De Souza – UNIFEI/IESTI
Internal Member

____

PhD. Carlos Henrique Valerio De Moraes – UNIFEI/IESTI
Internal Member

ITAJUBÁ JULY 13, 2021.

CATALOG FORM

SOUZA, G.
Deep Learning and Feature Engineering techniques applied to the Myoelectric Signal for accurate prediction of movements [Itajubá/MG] 2021.
xvi, 145p., 210 x 297 mm (EEL/IESTI/UNIFEI, PhD., Electrical Engineering, 2021).
Doctoral thesis – FEDERAL UNIVERSITY OF ITAJUBÁ, Institute of Systems Engineering and Information Technology.

1. Deep Learning
2. Mioelétrico
3. CNN
4. LSTM

I. EEL/IESTI/UNIFEI
II. Título (série)

BIBLIOGRAPHIC REFERENCE

(2021). Deep Learning and Feature Engineering techniques applied to the Myoelectric Signal for accurate prediction of movements Doctoral thesis in Electrical Engineering, Publication , INSTITUTE OF SYSTEMS ENGINEERING AND INFORMATION TECHNOLOGY, FEDERAL UNIVERSITY OF ITAJUBÁ, Itajubá, MG, 145p.

ASSIGNMENT OF RIGHTS

Gabriel Cirac Mendes Souza

INSTITUTE OF SYSTEMS ENGINEERING AND INFORMATION TECHNOLOGY - IESTI
FEDERAL UNIVERSITY OF ITAJUBÁ
Campus Itajubá
37500 903 - Itajubá - MG - Brasil

Dedication: In memory of my grandparents, doctors in the science of life. To my parents, true pillars. To my love jú, bold happiness.

ACKNOWLEDGMENT

*First, was my **Mother**, she bought books, taught me, and told me stories. Teacher **Lúcia** also liked plants and exercised her creativity with cutouts or architectural projects. She taught me how to read the world. Then came my **Father**, calm, with his head down and a lot of curiosity. He had no titles, but he was wise and persistent, sometimes stubborn, sometimes empirical. **Cezar** didn't mind making the food. I looked at it and absorbed it. Then came some women who left their homes to teach other people's children. Aunt **Cida**, **Márcia**, **Marlene**, **Mara**, **Zélia** and many others, who left noble impressions. There were also the teachers, **Amadeu** played the guitar, and **Bonatto** put music to play. The music impressed me, and there is no creativity without art, the hypotenuse of the equilateral triangle. There were, yes, literate and elegant teachers, **Enzo** walked into the room and broke the impersonality, **Thaty** calmly explained, **Zambroni** said there is no shame in thinking. **Pimenta** was methodical, seemed resilient and inspired admiration, **Moreno** trusted, and that motivated me. **Maurílio** had planes and spoke of such intelligence that he flew. All of them, I remember them all, with affection and kindness. Many others with their thoughts and importance. Each one an impression, a lesson, some stay, and others go, but all are. Some **Friends** told me that it was necessary to take a deep breath sometimes that **Coqueiro** doesn't shade in Lisbon, drink good beer and talk casually. Others said that what you like to do makes you. I think of all of our dear **Itajubá**, whose impressions are still impressive. The **Family** inspired confidence, and that was how, afraid that it would come to an end, every minute of this journey was precise, coming to an end. Then **Ju** came and taught me the love, calmed me down with warmth, and showed me the sea of yesteryear. The **Faith**, the science of instinct, taught me hermetic things. Ahh, the **Gratitude**... this magnetic force.*

RESUMO

Técnicas de reconhecimento de padrões no Sinal Mioelétrico (EMG) são empregadas no desenvolvimento de próteses robóticas, e para isso, adotam diversas abordagens de Inteligência Artificial (IA). Esta Tese se propõe a resolver o problema de reconhecimento de padrões EMG através da adoção de técnicas de aprendizado profundo de forma otimizada. Para isso, desenvolveu uma abordagem que realiza a extração da característica a priori, para alimentar os classificadores que supostamente não necessitam dessa etapa. O estudo integrou a plataforma BioPatRec (estudo e desenvolvimento avançado de próteses) a dois algoritmos de classificação (*Convolutional Neural Network e Long Short-Term Memory*) de forma híbrida, onde a entrada fornecida à rede já possui características que descrevem o movimento (nível de ativação muscular, magnitude, amplitude, potência e outros). Assim, o sinal é rastreado como uma série temporal ao invés de uma imagem, o que nos permite eliminar um conjunto de pontos irrelevantes para o classificador, tornando a informação expressivas. Na sequência, a metodologia desenvolveu um software que implementa o conceito introduzido utilizando uma Unidade de Processamento Gráfico (GPU) de modo paralelo, esse incremento permitiu que o modelo de classificação aliasse alta precisão com um tempo de treinamento inferior a 1 segundo. O modelo paralelizado foi chamado de BioPatRec-Py e empregou algumas técnicas de Engenharia de *Features* que conseguiram tornar a entrada da rede mais homogênea, reduzindo a variabilidade, o ruído e uniformizando a distribuição. A pesquisa obteve resultados satisfatórios e superou os demais algoritmos de classificação na maioria dos experimentos avaliados. O trabalho também realizou uma análise estatística dos resultados e fez o ajuste fino dos hiper-parâmetros de cada uma das redes. Em última instancia, o BioPatRec-Py forneceu um modelo genérico. A rede foi treinada globalmente entre os indivíduos, permitindo a criação de uma abordagem global, com uma precisão média de 97,83%.

Palavras-chave: Bio-sinais, BioPatRec, CNN, Engenharia de Feature, Engenharia de Reabilitação, LSTM.

ABSTRACT

Pattern recognition techniques in the Myoelectric Signal (EMG) are employed in the development of robotic prostheses, and for that, they adopt several approaches of Artificial Intelligence (AI). This Thesis proposes to solve the problem of recognition of EMG standards through the adoption of profound learning techniques in an optimized way. The research developed an approach that extracts the characteristic a priori to feed the classifiers that supposedly do not need this step. The study integrated the BioPatRec platform (advanced prosthesis study and development) to two classification algorithms (Convolutional Neural Network and Long Short-Term Memory) in a hybrid way, where the input provided to the network already has characteristics that describe the movement (level of muscle activation, magnitude, amplitude, power, and others). Thus, the signal is tracked as a time series instead of an image, which allows us to eliminate a set of points irrelevant to the classifier, making the information expressive. In the sequence, the methodology developed software that implements the concept introduced using a Graphical Processing Unit (GPU) in parallel this increment allowed the classification model to combine high precision with a training time of less than 1 second. The parallel model was called BioPatRec-Py and employed some Engineering techniques of Features that managed to make the network entry more homogeneous, reducing variability, noise, and standardizing distribution. The research obtained satisfactory results and surpassed the other classification algorithms in most of the evaluated experiments. The work performed a statistical analysis of the outcomes and fine-tuned the hyperparameters of each of the networks. Ultimately, BioPatRec-Py provided a generic model. The network was trained globally between individuals, allowing the creation of a standardized approach, with an average accuracy of 97.83%.

Keywords: BioPatRec, Bio-signals, CNN, Feature Engineering, LSTM, Rehabilitation Engineering.

SUMMARY

LIST OF FIGURES

LIST OF SYMBOLS

Greek Symbols

N	Sample size
Λw_{jk}	Weight w_{jk} adjusted according to the descending gradient method
μ	Average
ϕ	Scale parameter
$\rho_{X,Y}$	Pearsons Correlation
σ	Standard deviation
θ_k	Adjustment value *bias*
θ_n	Natural parameter

Latin Symbols

$E(W_{jk})$	Error Function
$E(y)$	Mean fo variable of interest
$T_{j,I(x)}$	Gaussian neighborhood in the neuron j
$V(y)$	Variance fo variable of interest
$W(x)$	Weight vector that resembles the input
X	Input matrix
X_{Norm}	Normalized input
Z	Linear transformation
Z_2	Sigmoid function
Σ_k	Covariance matrix
η	Learning Rate
$\frac{\partial A_1}{\partial Z_1}$	Derivative of the activation function
$\frac{\partial E}{\partial O}$	Error between the input and the output
$\frac{\partial E}{\partial Z_2}$	Derivative of the sigmoid function
$\frac{\partial E}{\partial f}$	Gradient
$\frac{\partial Z_1}{\partial f}$	Derivation with respect to filter
$\tilde{C}_t$	Vector that defines the cell activation candidate
ε	The error in the output j
$\vec{u}$	Averege vector
a, b, c	Função de distribuição
b_i	Bias i
c_t	State vector of the cell
$e_j^2(n)$	Instantaneous error energy in neuron j
$f(y\|\theta, \phi)$	Distribution density
f_t	Forgotten

$g(u_k)$	Activation function
h_t	Hidden State
h_t	Hidden State
i_t	Current state update
o_t	Activation vector of the exit gate
$p(\vec{x}\|y)$	Predictor for the class y given an observation of $\vec{x}$
w_n	Neural network weight n
x_n	Neural network input n
x_t	Input vector
y	Neural network output
yd	Target variable
z	Z-score

LIST OF ACRONYMS

AI Artificial Intelligence.
ANN Artificial Neural Network.

CNN Convolutional Neural Network.

DL Deep Learning.
DoF Degree of Freedom.

EMG Myoelectric Signal.

FD Fractal Dimension.
FFT Fast Fourier Transform.
FPGA Field Programmable Gate Array.

GPU Graphic Processing Units.
GRU Gated Recurrent Units.

HDL Hardware Description Language.
HFD Higuchi Fractal Dimension.

LSTM Long Short-Term Memory.

MABS Mean Absolute Value.
MAD Mean Absolute Difference.
MFL Maximum Fractal Length.
MLP Multilayer Perceptron.
MUAP Motor Unit Action Potential.

PR Pattern Recognition.
PWR Average Power.

REN Rough Entropy.
RMS Effective Value or Root Mean Square.
RNNs Recurrent Neural Networks.

SSC Signal Slope Change.

TMR Targeted Muscle Reinnervation.

VAR Variance.

WHO World Health Organization.
WL Waveform Length.

ZC Zero Crossing.

1 INTRODUCTION

> The intelligence is the educated insolence.
>
> Aristóteles

1.1 MOTIVATION

Individuals who were born with some congenital malformation or who suffered an accident that resulted in the amputation of a limb have a natural decline in their quality of life. In an attempt to improve the daily lives of these people, Rehabilitation Engineering develops computational methods capable of classifying the movements according to the myoelectric signal collected in the person's residual limb. The researchers embed these methods in a mechanical prosthesis capable of reproducing movement and assisting in the daily tasks of the wearer. The researchers implement computational techniques to distinguish movements through Artificial Intelligence algorithms in a pattern recognition system. Due to the natural variability of the system during everyday use, classification results differ significantly from laboratory testing. Furthermore, the literature lacks methods sophisticated enough to operate under complex conditions. In this sense, the motivation of this thesis would be the technical application of Deep Learning for the accurate prediction of movements according to the electrical signal collected on the individual's skin (EMG).

1.1.1 Social Motivation

Engineering is at the service of society, and designers must embark value in their creations so that the purpose of the use is aligned with their actual use [Souza, Pimenta e Moreno 2020]. This Thesis has a social appeal as its contribution is linked to the life quality of people who do not have one of its limbs. The product derived from the research is a robust candidate to be shipped in an intelligent prosthesis. The concepts introduced can help improve the quality of life of people who suffer from this injury, as it can improve the methods currently available. The principles that guided the development of the research are ethical and directed towards social well-being. The author analyzed the practical aspects of the product under development and then proceeded to create a solution capable of bringing some addition to society.

The Role of Engineering: An inattentive look can consider the engineer as an adjunct in this process. Although engineering is not strictly related to rehabilitation issues, it addresses this issue at several points [Souza, Pimenta e Moreno 2020]. Even before trauma occurs, some problems must be taken into account by engineers in various fields. The designer of a car or airplane must always analyze the vehicle safety requirements, just as the electrical engineer must have normative instruction to develop his work. According to Donald Schön, engineers should always think about what they did, how they did it, and what they could have done better, leading to deep thinking. Ibo Poel et al., In his work [Poel e Kroes 2014], quotes a detailed discussion of the subject in philosophical terms, where he presents three different types of values that technology can embark on:

1. The actual purpose of use;
2. Its characteristics indirectly incorporated;
3. Its actual use.

Take, for example, a pacemaker, an instrument used by people who have certain types of heart problems. Project values, embedded values, and use values are closely associated. Thus, the project aligns the design and realization of the use [Souza, Pimenta e Moreno 2020]. On the other hand, a knife whose value is to facilitate food preparation can serve a disastrous purpose. In the development of prostheses, the three values converge on the same point: improving the amputee's life. Looking more closely at this issue, Basart et al. propose a list of 10 characteristics that are complementary to the technical responsibility of the 21st-century engineer. One is the need for a socially responsible conscience [Basart, Farrus e Serra 2019]. Engineers must apply values in line with social progress, ensuring a sustainable development model for humans [Souza, Pimenta e Moreno 2020].

1.2 PROBLEM IDENTIFICATION

Currently, Artificial Intelligence (**AI**) is the engine of several solutions provided by computing [Lin et al. 2019]. Deep learning techniques (**DL**) are being used massively in applications that use classification, regression, and clustering approaches and demand real-time response [Medus et al. 2019]. Autonomous cars and drones, medical applications, and smart devices are some of the examples that have already started to take advantage of this technology [Hengstler et al. 2016].

Currently, there is a growing tendency to imbue decision-making products. Through non-linear computational methods, it is possible to model a wide range of problems similarly, where the machine can create the resolution context. Any situation where there is a relationship between a set of inputs and outputs can adopt such a system. With this approach, it is not necessary to know the dimensioning that correlates the variables to obtain a functional method. This advance frees man from creating the rules of control and allows a proactive system to do a variety of tasks. The most varied procedures are used [Kulkarni et al. 2020] and its applicability became possible thanks to the computational increment. The models are trained offline and are embedded in a dedicated system that meets the application requirements.

The medical field is a science that broadly adopts this concept in its research and develops products with this differential [Kulkarni et al. 2020]. One of the ramifications is Myography, which studies the electrical signal from the brain, whose responsibility is, among other things, muscle control. The signal travels through the nerves and muscles until it reaches the surface of the skin. The Myoelectric Signal (**EMG**) carries a load of information that is used to perform physical diagnosis and develop methods that allow the creation of intelligent prostheses [Li et al. 2018]. At the beginning of such investigations, the researchers believed that this signal was stochastic. That is, a random event devoid of valuable information and difficult to track. However, scientists soon realized that after 200 milliseconds, the electric wave showed intelligible patterns [Oskoei e Hu 2007, Hudgins, Parker e Scott 1993].

This research directs efforts to the problem of classifying movements according to EMG. The idea is to find out which action an individual tries to perform based on the nature of the electrical signal captured on the skin surface. To classify the movements according to the EMG, this work will use pattern recognition techniques, using deep classification algorithms, and feature engineering. Such a prosthesis is a real-time system and can be considered to a certain extent a critical product, where failures must be minimized or extinguished. Failure means the wrong prediction of a movement according to the characteristics (features) captured by the system and presented to the network in one of its iterations with the human-machine interface.

The consortium between myography and AI is not new. However, the approaches that represent the state-of-the-art have not yet solved the problem. Currently, there are accurate and fast solutions that use simple classification methods, but the clinical results differ considerably from those obtained in a controlled environment [Waris et al. 2019, Farina e Aszmann 2014] and no fully functional product is available. The use of trivial techniques is justified by the need to ship software that allows speed and battery savings. Unfortunately, it presents its restrictions when the system is susceptible to variations and cannot be considered robust. DL techniques can create a more elegant abstraction between related inputs, and they create contexts that are sufficiently representative [Esteva et al. 2019] and can work under complex conditions. Thus, these sophisticated neural networks could work more robustly in an authentic environment of use. Moreover, one can think of a system that is not limited to predicting the movement but the strength and speed.

The attraction that allows mixing the approaches is the current computational increment and the parallelization tendency of the devices that perform the necessary calculations to allow more advanced techniques [Li et al. 2016, Madiajagan e Raj 2019]. In addition, the developers realized that it is essential to create equipment with hardware dedicated to the processing of neural networks, capable of accelerating and allowing the reconfiguration of a product. Field Programmable Gate Array(**FPGA**) is an integrated circuit that allows its features to be described by the programmers and not by the manufacturer [Barik e Sinha 2016, Trimberger e Stephen 2015], which provides a high degree of modularity. The technology is being developed to supplement AI techniques and allow greater computational power [Yuan et al. 2019]. Another possibility of a similar nature is the use of Graphics Processing Units (**GPU**) [Madiajagan e Raj 2019], which can accelerate the process of training networks. Thus, if an approach is prosperous using a GPU, it would be possible to use the same concept in an FPGA. However, in the case of the FPGA, it would still allow the description of the embedded circuit in Hardware Description Language (**HDL**).

The FPGA has simple processing parts instead of complex computational blocks and, your internal organization is in a two-dimensional way. There is routing between the interconnections, and this allows the elements to connect in a varied way. Classification algorithms such as the Convolutional Neural Network (**CNN**) and Long Short-Term Memory (**LSTM**) are built with inputs and outputs arranged in parallel, successive layers that make calculations like convolution and perform simple mathematical operations [Indolia et al. 2018], [Li et al. 2015]. Thus, the FPGA has become a great candidate to embark on solutions that require AI techniques since the technology (arithmetic units and simple logic blocks) of it meets the characteristics of the most used networks [Lacey et al. 2016]. Concurrent computing, parallelism across multiple units, energy efficiency, the possibility of developing varied topologies, and the ability to reconfigure to create different products are the points that combine the two technologies [Lacey et al. 2016], [Nurvitadhi et al. 2017], [Shawahna et al. 2019].

The current manufacturers of FPGAs (Xilinx and Intel) already sell devices aimed at the AI branch, with several features developed at the hardware level to meet the mentioned requirements. Figure 1.1 presents a justification regarding the computational increment, note that each year the price and consumption become more convenient, while the computational power only increases [Trimberger e Stephen 2015]. The FPGA offers resources that can mitigate the limits of prosthesis development and allow for a broad and complex study of DL techniques in real-time systems. The parallelization with cost and reduced consumption are the attractions that will allow more complex applications.

In this sense, the hardware will be made to accommodate such techniques and mitigate the problems of time, battery savings, network complexity, and allow high efficiency combined with the more profound power of abstraction. The application of DL techniques to the myoelectric signal problem arises due to the clear developments that the technology will

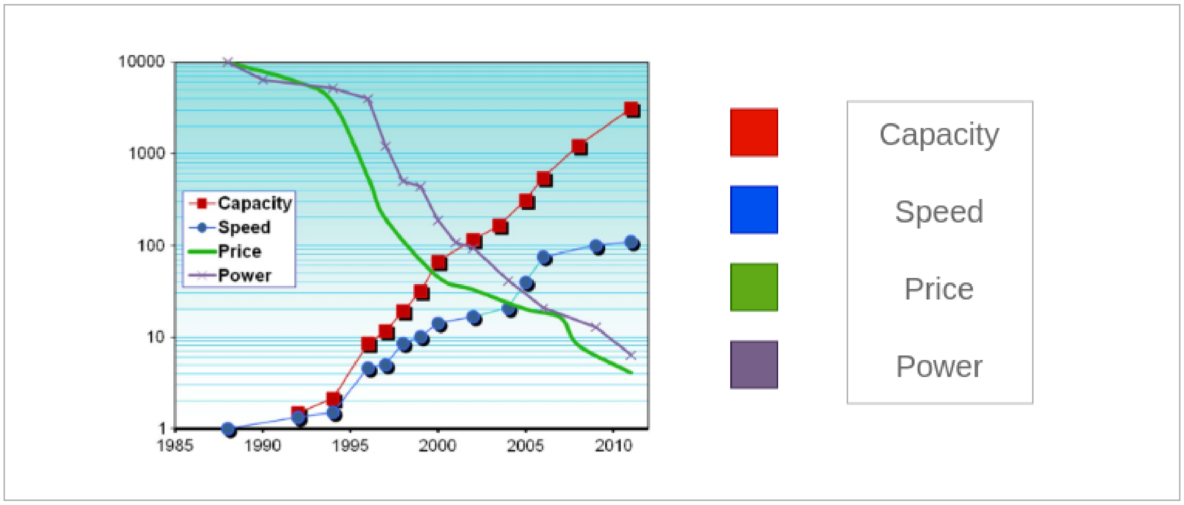

Figure 1.1 – Evolution of the characteristics of Xilinx's FPGAs. Illustration developed based on [Trimberger e Stephen 2015].

allow and as a possible solution to current limiting factors. In this way, the efficient junction between myography and deep neural networks will be made thanks to the evolution of the hardware. The devices will accommodate more complex arrangements at an acceptable cost and in real-time. The possibility of creating sophisticated contexts at a low price will allow the use of algorithms such as LSTM and CNN in products that have many functional limitations. The integration of parallelized devices, real-time applications that employ AI techniques in their design, and pattern recognition models in the myoelectric signal tend to converge since the features offered by FPGAs/GPUs will provide a more solid application base for such systems and, thus, take the research to fields that have not yet been explored.

Knowing that FPGAs/GPUs will fill a gap that allows performance, complexity, speed, low consumption, and real-time does not release us from the commitment to develop software capable of facilitating the learning process in a precise way. AI is a varied branch and has different fields of application and, its study is not limited only to the development of the model. It is also necessary to describe it optimally so that it does not incur a waste of resources. In this sense, techniques such as features engineering, feature extraction and filtering, are employed aiming a compensation through the software within the limits of the device's operation. When the network is no longer able to obtain satisfactory results (regardless of the metric), it is feasible to employ means to allow the method to be adapted to its functional requirements.

Before applying a technology, it is necessary to understand how it works, outline what it aims at and create means that allow its proper conception. Briefly and considering the case of EMG, it can be said that it will be necessary to develop an accurate, fast, real-time system capable of operating under adverse conditions, such as electrode displacement, humidity, impedance variation, and others. Thus, it is based on the premise that no computational method alone can efficiently deal with this classification problem, and it is necessary to study and implement more sophisticated means. Before applying a DL solution to an embedded system, it is indispensable to check its performance in a software system, optimize it, and check its feasibility of operation in comparison to traditional methods and then proceed to the development of the circuit. The next step must be the implementation of the parallelized network. In this way, it will be possible to assess whether the system obtains real gains with this methodology.

Deep Learning (DL) techniques can be used in a system that labels the myoelectric signal according to the movement that a person wants to perform, and unlike current intelligent methodologies, they can provide a more effective resolution and less subject to the system's natural variability. However, a computational cost that you want to mitigate. At first, networks such as CNN and LSTM would be able to solve the signal classification problem in an offline way, in a second step, the mentioned algorithms would be provided with information capable of better abstracting the system (using techniques that previously dealt with data), reducing training time, increasing accuracy and being able to operate under adverse conditions. Finally, knowing that the algorithms employed can adequately address this issue in a system based on the Von Neumann architecture, such methods would do so much more efficiently in a parallelized system such as the FPGA/GPU.

It is possible to improve classification methods with these characteristics to obtain greater precision and reduced workload. The use of algorithms like CNN would not be limited to systems with excellent hardware power and, it will be allowed to use it in domains whose specifications are more restricted. Thus, if any DL technique is capable of reaching or surpassing the traditional means of classifying EMG in a controlled environment, it would be effective in the case of practical use and less subject to its untimely circumstances, mitigating the deviations arising from the interaction of the prosthesis with the operating environment. In addition to the above, the embedded hardware systems of the future are built with native characteristics capable of optimally accommodating the implementation of products that use in-depth learning in their design. The trend is that AI models are parallelized at the hardware level, and their complexity is not a limiting point in the construction of a real-time prototype.

In summary, the approach will allow advanced classification algorithms to operate in situations where resources are scarce, obtaining results above the current average and at an acceptable cost since the dedicated circuit will guarantee the operation. The concept would not be restricted only to the case of EMG and can be extended to any product that has similar specifications, such as a real-time health monitoring system embedded in a watch.

1.2.1 Thesis Proposal and its Justification

The Thesis proposes the possible investigation of replacing these less accurate and single forms of pattern recognition with more complex ones such as the CNN and LSTM network. It developed a way to compensate the computational load through Feature Engineering, allowing to combine the best of both worlds: robustness and low hardware demand. Then, the research will investigate the limits of system operation on a GPU, thus considering a parallelized environment. For this accomplishment, this research created a platform called BioPatRec-Py [Souza, Moreno e Pimenta 2020] that allows the optimized consortium between the referred neural networks and the parallelized hardware. Finally, the investigation develops a generic and global model capable of operating in different individuals. The concept introduced is original since most researchers carry out individual training (to the detriment of the global one). When they do it similarly to the one proposed, they do not obtain satisfactory accuracy, which makes its practical application unfeasible.

1.3 RESEARCH OBJECTIVE

The objective of this research is to use LSTM and CNN networks in consortium with a feature pre-extraction technique and an adaptive Kaufman filter (usual in finance - this step guarantees that the system works homogeneously), which together allow to carry out the sequence of classification of the movements according to the EMG in a more precise way. Finally, it is expected that the approach will be able to make a more robust mapping of characteristics and, thus, can be trained in a population of individuals instead of carrying out individualized training. The objective is to train a network that operates uniformly in the entire population, despite the variability of the signal inherent to each one.

The scientific contribution will be the availability of software that allows the classification of EMG using DL techniques. In addition, the program uses feature extraction to lighten the system load and adaptive filtering to make the signal more homogeneous. The study will present a concept capable of allowing the generalization of network input and execution on a GPU instead of a serial CPU. Also, instead of using the raw signal in the form of an image, it will be represented through the features, which are tracked linearly and then arranged in matrix form. The concept can be extended to a specialized FPGA, which is also parallelized, allows the circuit description at a digital logic level, has hardware dedicated to DL techniques, and adopts the same frameworks used in this research.

1.3.1 Specific objectives

The development of this study begins with the bibliographic contextualization and has as specific objectives the in-depth understanding of the items discussed in this research, the correlation between them and their motivation, the presentation and testing of a new classification method, and, finally, the evaluation of the introduced concept. Thus, the following objectives are part of the research:

1. Study of the myoelectric signal, its nature, and its medical uses. Review of classic AI techniques, in-depth study of DL techniques, and Features engineering. Correlate and speculate the possible interactions on the three mentioned sciences;
2. Integration of Matlab's CNN and LSTM networks with BioPatRec software (advanced prosthesis development);
3. Development of a computational method capable of adapting the input of a CNN and LSTM network to receive a matrix of features instead of the raw signal;
4. Use of a more robust filtering method, making the input more homogeneous;
5. Adoption of an efficient standardization method, capable of making the distribution of characteristics uniform;
6. Development of a heuristic to adjust network hyperparameters through a grid protocol;
7. Conduct a comparative performance study between the most used algorithms in this field and perform statistical analysis;
8. Develop the Python approach itself so that the entire system can be evaluated in a parallelized hardware;
9. Perform the performance evaluation of the introduced parallel system;
10. Discuss and compare the results of the serial model and its counterpart executed on a GPU;
11. Create a global training method so that the network is highly accurate, considering any individual who uses the prosthesis.
12. Develop a proposal for future work, based on the results obtained;

1.4 RESEARCH HYPOTHESES

The hypothesis of this text is based on the following premises:

1. The classification algorithms used in the development of prostheses are not capable of delivering robust results. They do not create a model capable of correlating inputs so that uncertainty can be eliminated;

2. Embedded real-time classification problems are not likely to be solved by complex methods, as there are hardware requirements that are not easily met, as such networks require many resources to function accurately;

3. There is a tendency to develop specialized hardware for parallelization and advanced low cost and high-performance neural networks;

4. With this available hardware, there will be no gap between real-time applications and DL techniques;

5. Complex networks like CNN and LSTM can overcome general-purpose classifiers in a wide range of problems;

6. Thus, these algorithms are capable of being used to accurately predict the movement of a limb and without having to worry about the overall performance of the application;

7. The approach can be optimized through the engineering of feature;

8. The introduced concept will be generalized to other domains whose requirements are similar;

9. The system can be embedded in dedicated hardware that allows parallelization.

This thesis hypothesizes that myoelectric signal classification systems can benefit from deep learning techniques, allowing a more accurate functional model that accommodates real-time requirements and their respective operational restrictions. Therefore, this work aims to solve one of the gaps in intelligent prostheses, capable of predicting a movement based on characteristics extracted from that signal and used to feed a classifier.

The extraction of features *a priori* happens to allow deep neural networks to obtain lean and representative information, reducing their workload without degrading the system performance. Then, the proposal deals with the actual increase that the parallelization of such networks will allow so that the real-time execution of the algorithms is ensured by the hardware technology that the study adopted.

1.4.1 Question to answer

The development of intelligent prostheses uses simple artificial intelligence techniques in their prototypes, but their real applicability is limited. Would it be possible to use Deep Learning techniques in the EMG classification system accurately and quickly enough, so that its disadvantages are mitigated by the appropriate combination of software and hardware? In extension, is it possible to develop a comprehensive model that works on a range of individuals at the same time?

1.5 MAIN CONTRIBUTIONS

Instead of providing the machine with the raw signal, the investigation will introduce a concept of pre-processing, which admits that the information is previously treated on and thus has a more representative format. In addition, the study adopted a procedure for normalizing the characteristics that make the distribution uniform. This step facilitates the functioning of the objective function of most neural networks. This way, the classification will occur with precision and without demanding too much hardware. The research will integrate these prototyped networks in the Matlab software with BioPatRec prosthesis development software and make a comparative study with renowned classifiers in this domain of knowledge. The next stage of the investigation will develop a Python approach derived from BioPatRec but with a focus on parallelization through GPU use. Finally, the study will develop a global training approach.

At the beginning of the investigation, there were still no scientific efforts towards this direction, but as a growing trend, researchers began to use some of the ideas presented here but using the raw sign in the form of an image. This fact happened naturally, as such networks handle raw input and perform resource sizing by themselves. However, the results, despite being satisfactory, took a long time and the accuracy was below the classical methods. During the study, the author realized that treating a sign as an image did not look attractive for two reasons: First, the signs had obvious visual similarities and, consequently, this would make the network need more iterations to map the hidden features. Second, treating the input as an image implies that the data volume is significant. As most classification systems track the signal linearly and there is already solid literature on the best features, the research opted to verify if the previous extraction could improve a CNN network, known to dispense with this step. The work also proposed to carry out a comparative study and the adjustment of parameters through a grid protocol or factorial work.

Up to the time of this research, the use of DL techniques has not yet been able to reach the most respected algorithms in terms of accuracy and time. In addition, the features engineering has been dispensed with when the classifier can perform this step.

The main contribution of this research is the availability of a myoelectric signal classification method using Deep Learning techniques, capable of delivering a global model that works in the entire population rather than specific individuals. General training indicates the network's ability to perform complex resource mapping, which would ultimately provide a more robust model for day-to-day use. In addition, the research suggests that the pre-extraction of resources benefits models that use CNN in the case of EMG and serves to save computational resources, making the classification more objective and leaving the complex mapping of resources under the responsibility of the network. This work resulted in the following publications:

1. Pattern Recognition in Myoelectric Signals using Deep Learning, Features Engineering, and a Graphics Processing Unit. IEEE Access, v. 8, p. 1-1, 2020;
2. From AI and Electromyography to Financial Market: A Philosophical Perspective. IOSR Journal in Humanities and Social Science, v. 25, p. 27-35, 2020;
3. Netlab MLP - Performance Evaluation for Pattern Recognition in Myoletric Signal. PROCEDIA COMPUTER SCIENCE, v. 130, p. 932-938, 2018.

The following articles are under review:

1. Long Short-Term Memory for Pattern Recognition in Myoelectric Signal - Research on Biomedical Engineering;
2. Myoelectric Signal Classification Using Convolutional Neural Networks with PreExtracted Features - Signal, Image and Video Processing.

1.6 THESIS STATEMENT

Smart prostheses for amputees have not yet completely solved the problem due to the system's natural variability. Advanced myoelectric signal classification techniques can better handle the system's inherent noise and achieve more accurate classification. Current classification systems still need to carry out individual training, which makes the prosthesis use restrictive.

1.7 DOCUMENT ORGANIZATION

The theoretical basis will be divided according to the topic treated. However, the text may use concepts from other items when it is necessary to show the interaction between them. The research structure will have the following organization:

- Chapter Two will introduce the concepts of myography, AI, and some of its techniques then the two topics will be related. In the section on DL networks, two algorithms will be studied in detail, presenting their attractive features and limitations. The other classifiers will be discussed more succinctly. Finally, the bibliographic review will allow the reader to understand the current state of the subject.

- In Chapter Three, the text will describe the research methodology. Then, the reader will be presented with the step-by-step necessary to develop the methods, codes, test, and evaluation protocol, so that it is possible to establish a line of reasoning that allows the experiment to be repeated clearly and concisely.
- Chapter Four will present the results in a qualitative and quantitative comparison scheme. The presentation criteria will follow a line of interrelationships. Networks will be compared with each other and with themselves (considering the adjustment of their hyperparameters). This step will allow two different assessments in the same spectrum, where it will be possible to answer two questions: what is the best network, and what is its ideal operating range?
- Chapter Five will discuss the results and, where necessary, further investigation will be used to validate the hypothesis and support the analysis. This item will follow the pattern of the previous one, using two types of comparison.
- Finally, Chapter Six concludes on the main points of research throughout its development. The article will present the reader with an outline of future research based on everything that has been presented. The author will suggest a potential investigative intervention to go beyond the limits of this research.

2 THEORETICAL REFERENTIAL

Philosophy is the theory of practice.

Alonso Barros

The presentation of the theory necessary for the general understanding of the thesis will start with the electrical physiology of EMG. The text will present the fundamental concepts of Artificial Intelligence and Artificial Neural Networks for a broad understanding of the subject. Finally, the thesis will study the relevance and status of the topic through a bibliographical review.

2.1 BACKGROUND

2.1.1 Physiology of Myoelectric Signal

The muscle is built of motor units that are formed by muscle fibers and have terminal branches connected to the spinal cord [Merletti e Knaflitz 1992]. The nervous system controls muscle activity and modulates the number of motor neurons and their respective recruitment rate, thus differentiating one activity from others. The synaptic reaction that happens between the motoneuron cells and the terminal branch of the cells that reach the muscle fiber determines the movement. Such neurons are capable of initiating a reaction that allows depolarization of the medium and propagates in two directions opposite to the fiber [Merletti e Knaflitz 1992], these reactions generate two action potentials. EMG arises precisely from the aforementioned potential, which leads the muscle fiber to contraction [Ortolan 2002].

Luigi Galvani discovered the phenomenon in 1780 [Piccolino 1998], during an animal physiology class. He found that the muscle fiber was excited by electrical signals when it touched a dead frog with two different metals, and it underwent muscle contractions. Electromyography consists of EMG monitoring over time, methods of evaluation, and diagnosis of problems of a muscular nature using the technique. The technique is applied in disease diagnoses, such as weakness, muscle pain, cramps, and involuntary movements. Currently, the concept is used in different branches of knowledge, notably in medicine, dentistry, physiotherapy, physical education, and in the case of this research in Rehabilitation Engineering, notably in the AI field and pattern recognition through classification algorithms.

2.1.1.1 Action Potential

The neural action potential is an electrochemical activity charged with conducting neural nerve signals. It arises due to the rapid variation between the internal and external potential of a neuron [Ortolan 2002] and [Farina et al. 2010]. Three distinct electrical phases describe the potential:

- **Depolarization:** In this phase, the permeability to sodium ions of the cell membrane is more prominent [Krueger-Beck et al. 2011]. This characteristic allows a high flow of sodium ions (Na+) to cross the membrane through the simple diffusion process. The intracellular medium starts to contain a large amount of positively charged ions (cations) and this inverts the cell's potential when confronted with its resting state. The measured voltage is around +30 to +50 millivolts.

- **Repolarization:** The previous process is reversed, and at that phase, there is an increase in the permeability to potassium ions (K+), to the detriment of sodium ions [Krueger-Beck et al. 2011], which tend to normalize. During a short period, a high flow of potassium ions occurs from the inside to the outside of the cell medium. This depletion of cations by the sodium-potassium pump causes the potential in the cell membrane to return to negative (somewhere around -75 millivolts).

- **Rest:** At this point, the return to normal resting conditions found on the membrane takes place before the process of excitation and depolarization. In this stage, the permeability to potassium ions returns to its initial level, and the voltage comes into equilibrium when it reaches -70 millivolts.

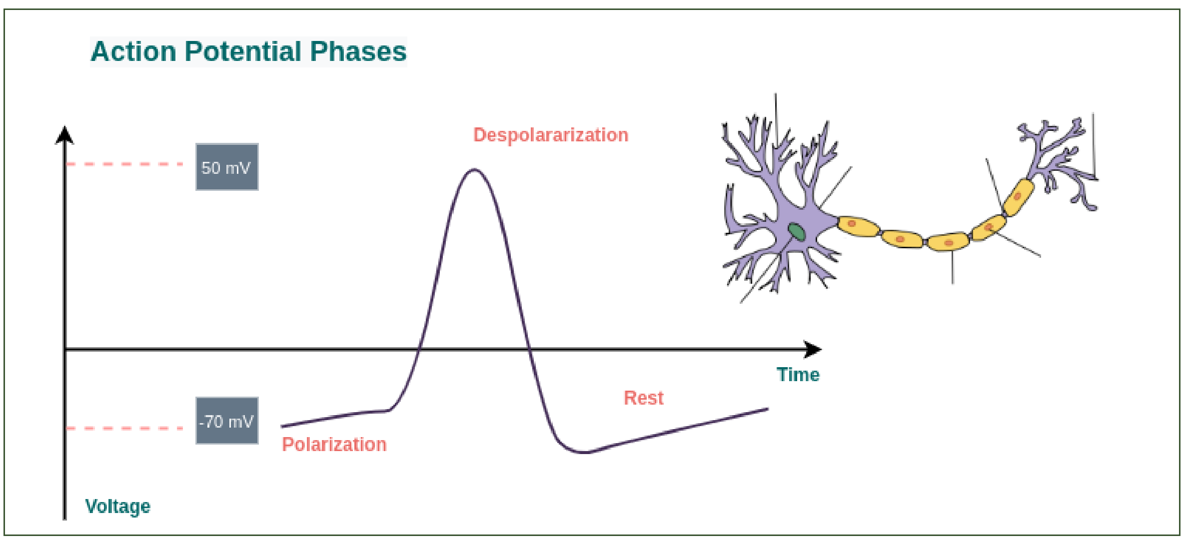

Figure 2.1 – Evolution of electrical voltage as a function of time, according to the permeability of the neuron membrane and its respective phases. Illustration made from a figure obtained in *Creative Commons*.

Figure 2.1 illustrates the potential change of the cell membrane during the respective phases. For depolarization to occur, some stimulus must occur (electrical, thermal, chemical, or mechanical).

2.1.1.2 Neuromuscular Junction and Motor Unit

The neuromuscular junction is the connection between the axon of a motor neuron and a motor plate [Borges 2014], which is the region of the plasma membrane of a muscle fiber where the nerve and muscle meet and use acetylcholine as a neurotransmitter. The nerve fiber branches to form the endplate, which innervates into the muscle fiber [Ortolan 2002]. Acetylcholine is a chemical mediator that allows the electrical stimulus to be interpreted and can trigger movement.

When an action potential reaches the axon, acetylcholine is released, and the receptors thereby make the muscle membrane permeable to sodium ions (Na+). Then, the membrane is depolarized, and acetylcholine is transformed into acetic acid and choline. The process takes 200 milliseconds and is necessary for the motor plate to receive another stimulus. This value is also used as a reference in the process of extracting information from the electric wave.

The motor unit group consists of muscle fibers and a single motor neuron controlling them [Farina et al. 2010]. A nerve fiber can innervate from one to hundreds of muscle fibers and, this number is called the innervation rate. These motor units and motor neurons that innervate them have different sizes. Small motor neurons innervate few muscle fibers, which is why these motor units generate less force. Large motor neurons are capable of innervating several groups of fibers, creating more powerful and larger motor units. Strength is controlled according to the number of units recruited during contraction and the specific stimulation frequency.

After a given frequency, the stimuli will overlap, which results in the phenomenon called tetanization, which provides a smooth and gradual contraction of the muscle. Stimulation frequencies range from 20 Hz for slow muscle fibers to 100 Hz for fast fibers [Ortolan 2002]. Figure 2.2 shows how the motor neuron binds to muscle fibers.

2.1.1.3 EMG Characteristics

When a neuron sends an action potential, the muscle fibers of the respective unit are excited [Farina et al. 2010]. However, the fibers are not stimulated simultaneously, and there are short delays between contractions. This phenomena occurs because the propagation times are not uniform due to the random nature of the neurotransmitter discharges at the neuromuscular junctions. The action potential of the motor unit is represented by the

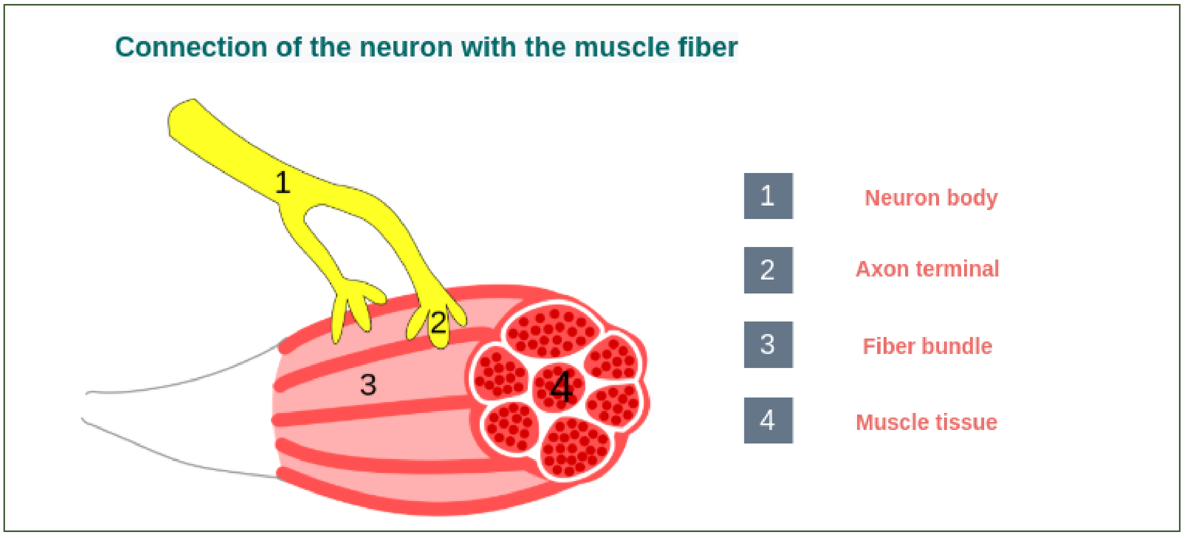

Figure 2.2 – Connection scheme between the terminals of neurons and the set of fibers. Illustration made from a figure obtained in *Creative Commons*.

algebraic sum of the action potentials on the n fibers of a motor unit, and its acronym in English is MUAP (Motor Unit Action Potential. MUAP has a short period (between 2 and 10 ms), so the units must be activated constantly. In this way, the contraction is maintained for a longer time. The potential sequence is known as the motor unit action potential train, and it generates an electromagnetic field close to the fibers. Thus, by placing an electrode on the perimeter of this field, it is possible to measure the electrical potential related to muscle contraction [Ortolan 2002].

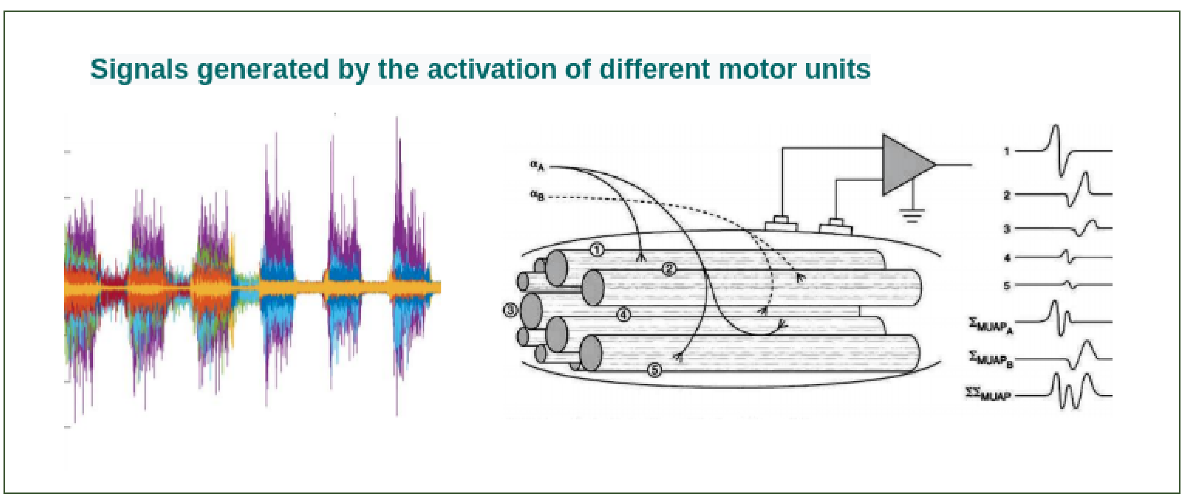

Figure 2.3 – Set of signals that are generated according to the muscle group that is recruited by its motor units. Source: [Howard 2016]

The sum of several action potentials generates the myoelectric signal [Luca et al. 1982], which is different for each group and represents only one movement. The activation of several motor units creates different EMG patterns. Figure 2.3 presents this concept. In this way, the combination of the motor units will generate different electromagnetic fields, where each of them describes an electrical waveform related to a respective movement. This signal is easily sampled, being the main component of this research and necessary substrate to the Artificial Neural Networks.

2.1.2 Artificial Intelligence

Artificial Intelligence (AI) is a scientific area that proposes to create computational methods that abstract the neural model together with other biological concepts and thus can reproduce some intelligence, although there are several definitions [Wang 2019]. The term was introduced by John McCarthy in 1956 during a conference at Dartmouth College, New Hampshire, United States [Andresen 2002]:

> "We proposed that a group of ten men conduct a two-month study on artificial intelligence during the summer of 1956, at Dartmouth College in Hanover, New Hampshire. The study is based on the idea that every aspect of learning or any characteristic of intelligence can, in principle, be described so precisely that a machine can be created to simulate it."

The idea is to make the machine learn to solve repetitive issues and thus be able to automate some activity. Instead of delegating the work to the human being, execution is attributed to the algorithm, which has decision-making power. The first concept implication is the ability to automate a process, which can be monotonous, like a surveillance system, for example. The second attraction is the saving of human resources. In addition, AI provides ways to guarantee homogeneity in the execution of tasks, which does not happen with human individuals. In practical terms, AI is already used by several domains [Vinuesa et al. 2020], as long as it is possible and feasible. The concept of AI is broad and has many categories, such as:

- Genetic Algorithms;
- Artificial Neural Networks;
- Deep Learning;
- Fuzzy Logic;
- Data Mining;
- Pattern Recognition;

- Feature Engineering;
- Regression;
- Statistics;

The research will consider the following fields: Artificial Neural Networks, Deep Learning, Engineering of Features, Pattern Recognition, and Statistical Analysis for the development of the literature. Before going into each sub-items, it is relevant to describe the model that inspired the mathematical abstraction necessary to create this powerful concept. It is important to emphasize that each architecture had a biological basis as a source of inspiration. Each network implements a different mathematical model or makes a combination of approaches. For example, the following networks and their respective concepts can be briefly mentioned:

- **CNN:** This algorithm performs a process of successive filtering of the image and was created based on the visual system of cats, which has an elaborate scheme of visual overlap in their cerebral cortex;
- **LSTM:** Such a network does not analyze only the current context of a sample but its evolution over time. In this way, it can simulate memory;
- **Genetic Algorithm:** The method used by the algorithm is based on the evolution through successive generations of an initial population, where through a process that imitates natural selection, an individual is evolved enough to solve the proposed challenge.

Before explaining networks and their mathematical properties, it is essential to understand in which modeling the problem fits. In the subsequent sections, the work will provide this basis for the reader.

2.1.2.1 Pattern Recognition

Pattern recognition is an AI field where the objective is to classify objects in different classes. The research fits precisely in that niche. Through the characteristics that the system presents, it is possible to distinguish such patterns [Bhamare e Suryawanshi 2018]. The objects of study can be varied, such as signals (voice, radio, or myography), pictures, videos, or any information that you want to separate into different classes [Asht e Dass 2012] (movements of a member from known data - supervised learning), but, when labeled information is unavailable, other algorithms are employed to discover unknown patterns (unsupervised learning). The object classification is an example of recognition, where, for each characteristic sample, a corresponding class is assigned. For example, think that the system input is a figure and that the model can distinguish between objects in traffic, so for each image, there can only be one solution (pedestrian, cyclist, car, motorcycle - multi-label classification).

Such an approach consists of a data capture unit (electrode, camera, positioning sensor, or others) that collects the data, a unit that handles the input, and a classification algorithm. After acquiring the sample, some methods perform the extraction of the characteristics (Features) [Bhamare e Suryawanshi 2018], which describe the problem in question and feed the classifiers with the information. Then the training process is carried out and, finally, the network can classify an object according to its intrinsic characteristics.

Classifiers should not be too generalized or too specific because in the first case, their classification would be vague, and in the second, it would exclude some relevant data, directing the result to inaccurate domains. The basic definitions of pattern recognition are:

- **Pattern:** An entity, event, or object, with a given definition that places it within a group with similar characteristics;
- **Class:** Group of objects or entities whose attributes are similar;
- **Feature:** Representative data obtained in the process of extracting features, usually a numerical or mathematical value, in some cases a geometric feature;
- **Classification:** Assignment of classes for entries considering their Features.

There are two types of algorithms capable of performing pattern recognition. What distinguishes them is their supervisory engine. In the **supervised** classification, the developer supervises the object training process. Each sample of features is assigned to a class, so the relationship must be known in advance [Carrizosa, Martín-Barragán e Morales 2011]. Each example is formed by a pair of interests, containing the input vector and its related output [Fabris, Magalhães e Freitas 2017]. The training process non-linearly correlates input and output and produces an output function capable of mapping new items. Some approaches that use supervised classification are:

1. CNN-type Recurring Neural Networks;
2. LSTM-type Recurring Neural Networks;
3. Support Vector Machines;
4. Decision Trees;

In the unsupervised classification, the labels are described by the computer [Ishola, Nawawi e Abdullah 2015]. The programmer selects the number of classes, and the algorithm subdivides the space into clusters or groups according to the numerical information of the data. This method is also known as clustering, as it creates groups according to the natural or statistical relationship of the data. The algorithm group the objects according to their spectral similarity [Saxena et al. 2017], and the system uses the features to analyze and group the data. Although the process is automated, the developer controls the process parameters, such as the number of classes, the maximum iterations (how many times the classification algorithm is executed), and the change limit, which specifies when the pattern recognition procedure is finished.

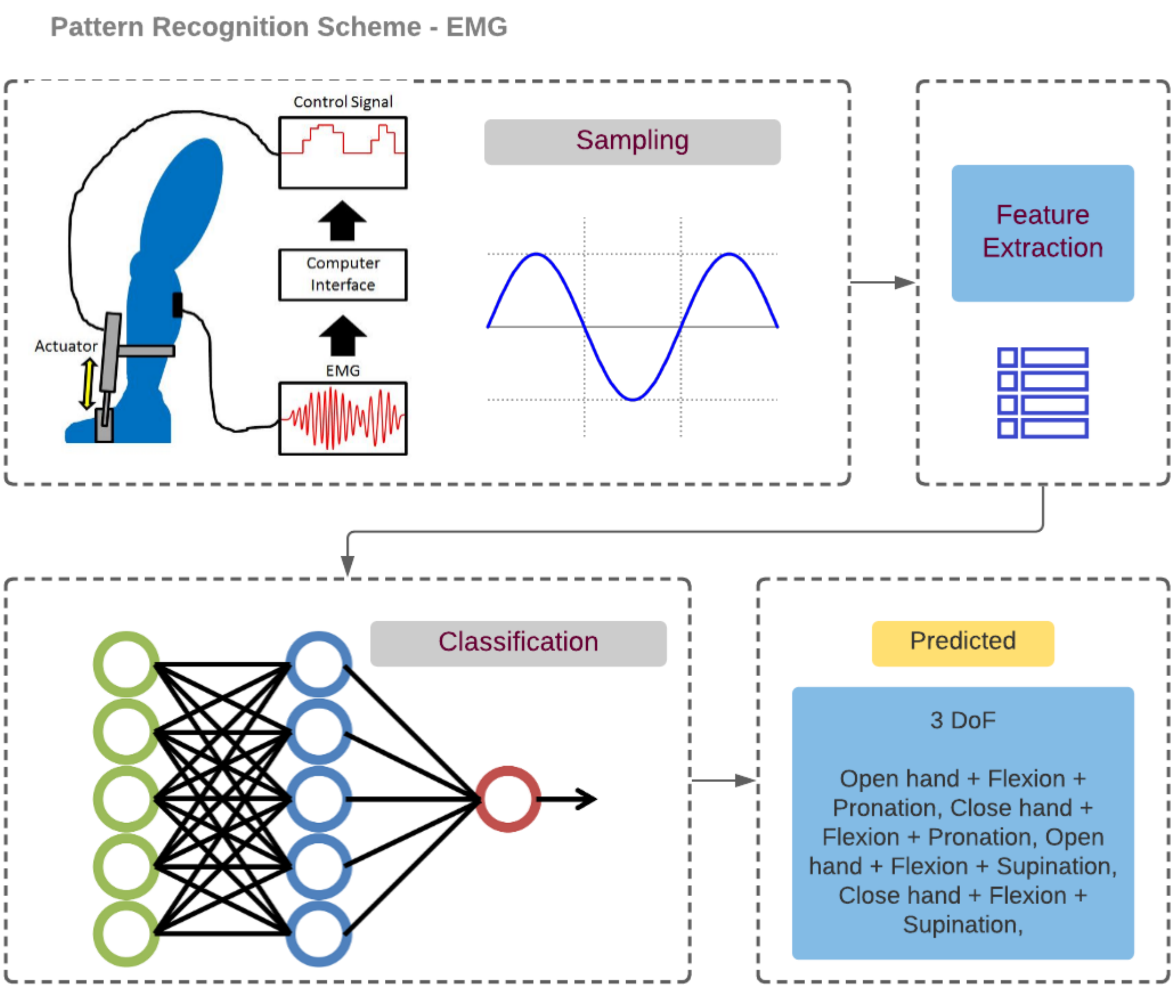

Figure 2.4 – Steps in the pattern recognition process. Illustration made from a figure obtained in *Creative Commons*.

After creating the clusters, the developer must interpret, label, and code the groups. Unsupervised sorting is quick and easy to perform. Below are some algorithms that use unsupervised classification in their architecture:

- *K-means*;
- *Mixture Models*;
- *Spectral clustering.*

Supervised classification is a different tool than unsupervised and depends heavily on training, data selection, and model type. In other words, they are distinct mechanisms whose purpose is similar, so it is not possible to say which one is better, as this response is linked to the context of the problem.

If the objects evaluated are too similar, the wrong classifications will tend to be high. This technique requires the features to be representative [Rawat e Khemchandani 2017] allowing to increase the separability boundary between classes. One of the points of this research fits here, where the investigation will propose a hybrid concept of extracting features. Figure 2.4 illustrates the points discussed in this section. The captured entry must be digitized and will go through a filtering process. Then the features are extracted using known techniques or statistical methods. The programmer trains the network using the functional model obtained, and it is possible to recognize a class from a new entry. Feature extraction allows you to alleviate the computational burden and utilize a simpler deep neural network, makes training faster, and keeps accuracy high.

2.1.2.2 The Artificial Neuron

Among all approaches to AI, it can be said that one of the pioneers relied on the animal neuron to structure its functioning [McCulloch e Pitts 1943]. The neural cell, succinctly, operates by modulating an electrochemical signal. The neuron receives an electrical impulse through its synapses, performs processing inside the nucleus, and sends the information ahead through the axon exit synapse. The process is repeated in a continuous chain of neurons in a complex network. Through this process, it is possible to change the information load over time and adjust its functioning to the organic need of the individual or organism.

An artificial neural network is a circuit composed of a vast number of neurons (processing units) [Abiodun et al. 2018], where these units can store and manipulate knowledge [Abiodun et al. 2018]. This system is parallelized and distributed. From this definition, an analogy can be established with the artificial neuron [Hassabis et al. 2017, Guresen e Kayakutlu 2011]. The comparison has its limitations and should not be understood strictly.

It is possible to compare the inputs of the network with the signals that arrive through the dendrites. The network weights are equivalent to the electrochemical modulators that transform the input and are regulated according to the problem. The function that minimizes the error between input and output can be thought of as the kernel. The weights adjustment according to the input is the mechanism that allows the arrangement of the network and leads to learning. The change in weights during the learning process is comparable to brain plasticity, that is, the biological capacity to adapt and circumvent unwanted situations.

Artificial neural networks are developed using mathematical techniques capable of changing their weights and changing the relationship between the input and output [Guresen e Kayakutlu 2011] so that they can describe the function that takes your image to the appropriate domain. Figure 2.5 draws a parallel between the artificial model and its biological counterpart. The network inputs represent the dendrites and will receive the *(x1, x2, x3 ... xn)* excitation signals. The chemical modulators represent the weights, and the core is a function that will minimize the error between the input and exit. Finally, the axon terminal represents the output, which has an activation function.

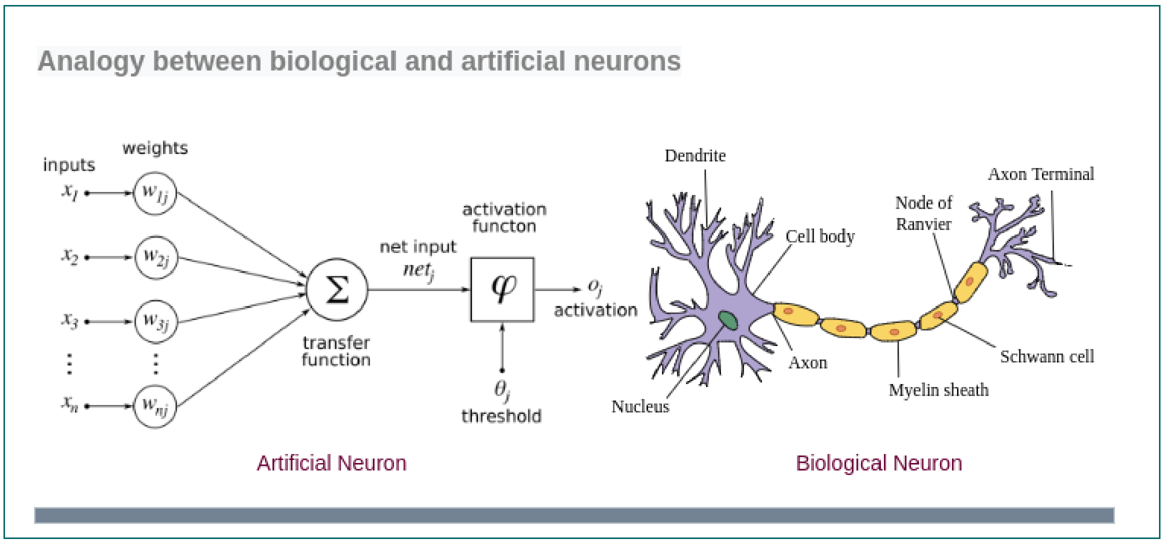

Figure 2.5 – Comparison between biological and artificial neuron concepts. Illustration made from a figure obtained in *Creative Commons*.

Warren McCulloch and *Walter Pitts* introduced an approach that was based on the biological concept so that an artificial neuron would activate its output signal if some inputs were active [McCulloch e Pitts 1943]. This concept was first implemented in electrical circuits. An Artificial Neural Network (**ANN**) is an algorithm created from a set of neurons that are interconnected in different topologies in a parallel and distributed way. The neuron is the processing unit and is responsible for manipulating information. Through the weights, knowledge is stored and adjusted during the iterative training process.

In practical terms, weight is a multiplication factor, which allows a given level of relevance to the input it controls. The learning occurs with the fine adjustment of these factors over time. When the optimal objective function is reached, the network can map the input to the output. From a new sample, the algorithm can recognize the pattern. The designer has complete freedom to architect the model, and no rule defines the ideal concept. However, very complex networks tend to increase training time and do not always result in better models. An ANN is usually made up of three layers: an input layer, one or more hidden layers, and the output layer. Each of them has its number of neurons, and these are densely connected with other units, and they must belong to layers different from the current one. An ANN can consist of hundreds of processing units; an individual's brain has billions of neurons. The concept is made by Illustration 2.6.

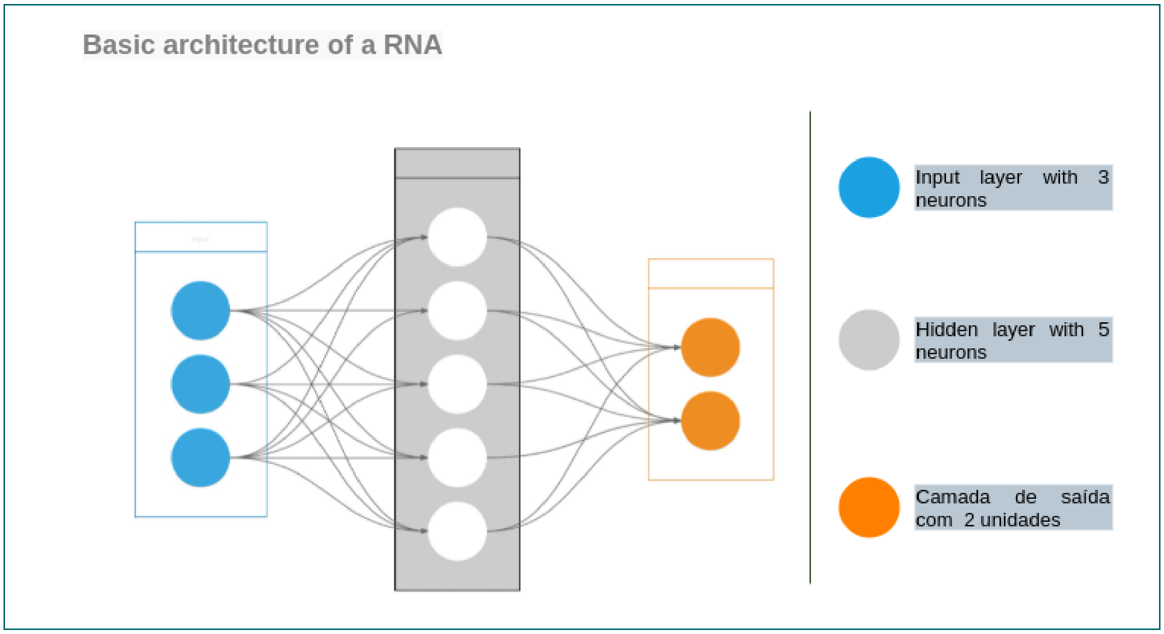

Figure 2.6 – The topology of an ANN is organized through parallel layers, which are densely interconnected by a group of edges, which play the role of the synapse and distribute the information across the units. Illustration made from a figure obtained in *Creative Commons*.

The concepts presented in this text are not recent, but they are gaining a lot of attention and relevance today. When *McCulloch* and *Pitts* published their first research on the topic [Morais 2010], the beginnings of neuro-computing began, aiming at the construction of brain-inspired methods. Between 1957 and 1958. The first computer directed to the branch appeared (to obtain success), the Mark I Perceptron was created by Frank Rosenblatt, Charles Wightman, and other scientists. The *IEEE International Conference on Neural Networks* took place in 1987 and was the first of its kind, since then, the subject has consolidated and is widely researched.

2.1.2.3 Multilayer Perceptron

Perceptron Multi Layers (**MLP**) is an ANN with retro propagation [Alaeldin Suliman e Yun Zhang 2015], which is composed of one or more hidden layers (between the input and output layers) [Marius et al. 2009], such a classifier is widely used in the case of myography. The backpropagation algorithm was proposed in the 1970s by Paul Werbos and is considered a robust training method [Alsmadi, Omar e Noah 2009]. Retro propagation means that the data flow towards the input to the output layer, and then the machine creates an error signal. The error propagates in the opposite direction to allow adjustment of the weights according to the measured error. The MLP network is a directed graph with a distributed and parallel processing structure. Graph nodes are neurons and, edges are directed paths that carry data from one layer to another. Input signals come from the outside world, and the output flows to the external environment in the same way. In general, the network works as follows:

1. - The units receive the input signals, x_1, x_2, x_n, through the synapses;
2. - The data is multiplied by the existing weights (random at the beginning), w_1, w_2, w_n. A $[-wt, +wt]$ range can be used;
3. The appropriate error function $E(W_{jk})$ and the learning rate η must be selected;
4. The equation that updates the weights is applied. The descending gradient [Andrearczyk e Whelan 2017] method is generally used, where the weights are increased in the negative direction of the gradient according to the relation:

$$\Lambda w_{jk} = -\eta \frac{\partial E(w_{jk})}{\partial w_{jk}} \tag{1}$$

5. The same is done for all other units in the other layers;
6. - An activation function is applied to the signal, for example, *threshold*, sigmoid, linear, or hyperbolic tangent. The result of this step forms the output signal y of the neuron in question;
7. - The output signal y is sent to the next layers;
8. - After many iterations of the process, the weights w_i are adjusted and thus describe the mathematical relationship between input and output.

In supervised learning, networks learn through examples and therefore make another analogy with a biological concept. Figure 2.7 illustrates the convergence process that the gradient method guarantees.

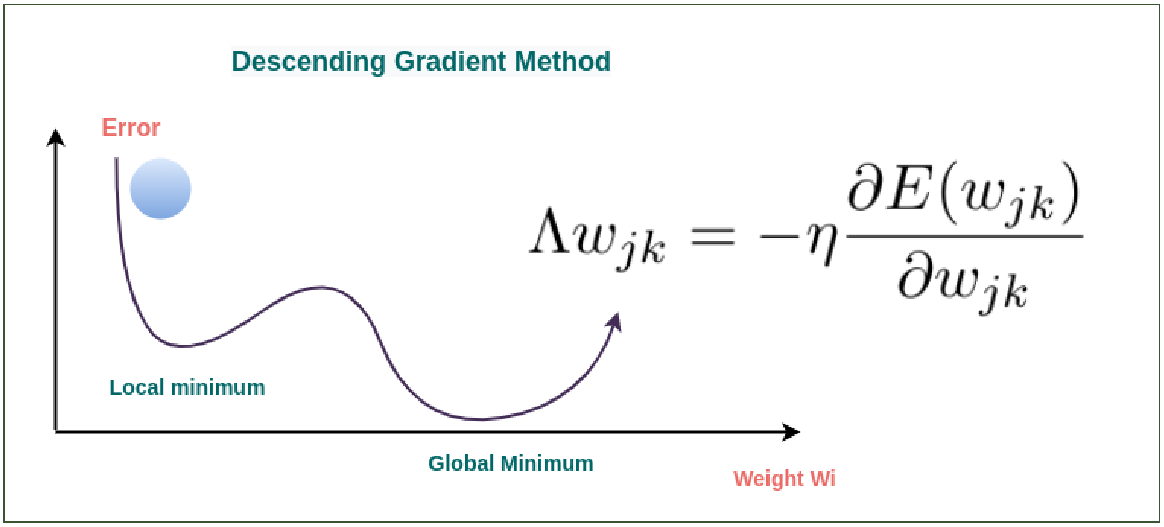

Figure 2.7 – The initial choice of the learning rate will define the convergence speed, low values tend to evolve slowly, but they sweep better and the solution space.

2.1.2.4 Learning Processes

The most relevant concept of neural networks is their ability to learn to improve their performance [Carvalho 2017]. For this, the network uses an iterative process capable of adjusting the weights of its units. Learning occurs when the network obtains a generic solution for a certain class of problems. The learning algorithm is nothing more than a set of well-defined rules for the problem in question, where the weights are adjusted to minimize the error between the input and the output [Li-Chiu Chang et al. 2012]. The training methodology employs most of the data to train the network (usually 70%), and when there is no temporal relationship, the data must be arranged randomly so that the network learns and does not memorize the rules. The samples that were not presented to the algorithm must be used to validate the [Li-Chiu Chang et al. 2012] model. This process is an example of supervised learning and is performed using the backpropagation algorithm, a generalization of the least-squares algorithm. The y output of the neuron k, is described mathematically by equation (2). Where the Activation function *g (.)* can be one of those mentioned and others: threshold, sigmoid or linear. The θ_k is an adjustment value known as *bias*.

$$y_{(k)} = g(u_k) = g(\sum_{j=1}^{n} w_{jk}x_{jk} - \theta_k) \tag{2}$$

The iterative training process is done through increments called epochs, and in each epoch, there is an adjustment in the multiplicative factor of the units. Generally, the training tends to converge more quickly in the initial stages, and in the sequence, more iterations are necessary to refine the model.

It is essential to consider the network generalization when the training data is insufficient or there is too much training. In this case, two problems can arise. When the model is too simple, it may incur a poor resolution, and the system is said to be under training [Ghasemian, Hosseinmardi e Clauset 2019, Jabbar e Khan 2014]. On the other hand, ANN can deliver an overly complex solution (overfitting), being unable to make predictions outside the known set [Ghasemian, Hosseinmardi e Clauset 2019, Jabbar e Khan 2014]. One way to assess and possibly avoid both problems is to divide the input into three groups of data: training set, validation, and test, and then check the ability of prediction, comparing the results obtained by the training set with those of the test. If the divergence between the values is high between the phases, it is necessary to make some intervention in the approach. Most of the data is left for training so that the network has a desirable basis for comparison and increases its field of adjustment. The validation data is used by the algorithm during training and is used to check if the resolution meets the expectations. Finally, the test data are samples that have not been submitted to the classifier and thus can confirm that the program meets your requirements. There is no general rule for dividing the data, but experience shows that employing 70% for training, 10% for validation, and 20% for testing is a good heuristic. Figure 2.8 illustrates a typical training process and a solution that presents the overfitting problem.

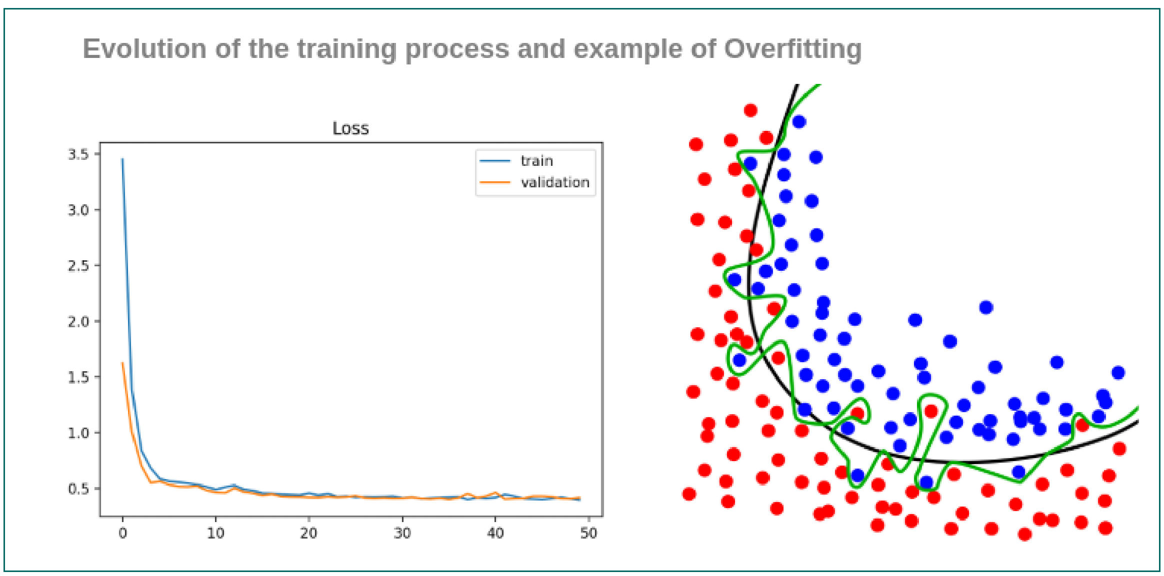

Figure 2.8 – Although the results obtained by the validation were close to the training values, this does not mean that the tests have the same accuracy.Illustration made from a figure obtained in Creative Commons.

As mentioned, learning occurs due to changes in synaptic weights. After processing the information and based on the error resulting from the output, the network adapts the weights to minimize the difference (between the expected and obtained result). This process is repeated according to a number pre-established by the programmer. Many seasons can result in slow training, and few iterations may be insufficient to achieve minimal accuracy.

This characteristic is the classic case of supervised learning, and it is employed through the *backpropagation* algorithm. The error in the output j is represented mathematically by equation (3), Where $e_j^2(n)$ is the instantaneous error energy in neuron j:

$$\varepsilon = \frac{1}{2}\sum_j e_j^2(n) \tag{3}$$

2.1.2.5 Self-Organizing Maps

In 1982 *Teuvo Kohonen* developed an unsupervised algorithm called Self-Organizing Maps (**SOM**) [Kohonen 1982], the method is based on Competitive Learning and has a neurophysiological appeal [Miljkovic 2017]. Consider sensory perception (visual, motor, or auditory) as the basis for an analogy, where each is mapped to a specific area of the brain. When a neuron receives a signal, the area around it (limited) experiences some excitation, and, in contrast, regions that are not related to that impulse tend to inhibit that signal. Thus, brain cells compete to become active and have different responses to the same wave.

Thus, with each iteration, there is a winning neuron, which, with the progressive adjustment of weights, tends to specialize in a given pattern and can detect a class of interest [Kohonen 2013]. The algorithm is capable of transforming a high-dimensional space into a two-dimensional one, placing similar elements geometrically close to each other (as in the case of sensory perception). Similar points are processed separately and placed in a region close to space and distance from the others. Its structure is simple, composed only of two layers, being the exit and the layer connected to the entrance.

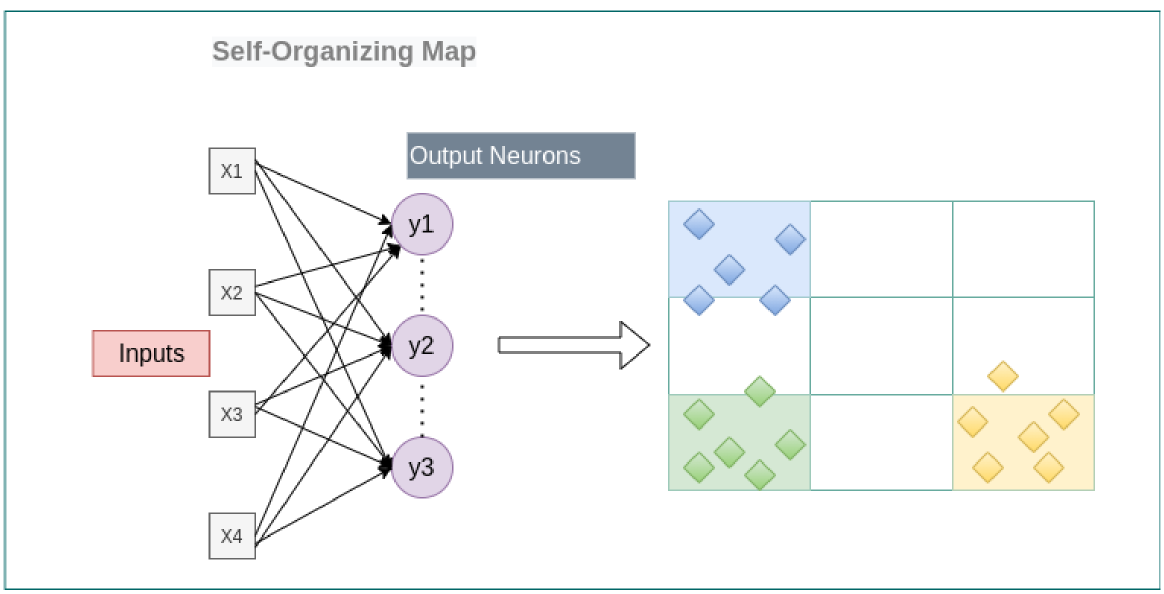

Figure 2.9 – Configuration of the self-organizing network and the result of the prediction of 3 classes in two-dimensional space.

Figure 2.9 shows the network architecture and map organization after training. The connections between the inputs and neurons are exciting, while the dashed connections between the output neurons themselves are inhibitory. The neuron that will activate the output will be the one that has the largest induced local fields, given a respective input. Such an induced result field is the combination of its data, and the output value can only assume two states (0 or 1) so that only one unit can reach level 1 at a time. The algorithm can be summarized [Chaudhary, Bhatia e Ahlawat 2014] as follows:

1. Initialize the weights w_j at random;
2. Feed the network with the sample vector;
3. Find the neuron $W(x)$ that has the weight vector closest to the entry, that is, whose value of $d_j(x) = \sum_{i=1}^{D}(x_i - w_{ji})^2$ is minimal.
4. Update the weights according to the equation $\Lambda w_{ij} = \eta(t)T_{j,I(x)}(t)(x_{ij} - w_{ji})$, where $T_{j,I(x)}$ is the Gaussian neighborhood in the neuron j and $\eta(t)$ is the learning rate;
5. Return to step 2 and repeat the procedure until the map stops updating.

2.1.2.6 Generalized Linear Model - GLM

A statistical method that is used in pattern recognition of the myoelectric signal is the Generalized Linear Model (**GLM**). The algorithm was developed by Nelder and Wedderburn in the 70s [Nelder e Wedderburn 1972]. The method is an extension of linear models and directly impacted the statistics. The algorithm uses other distributions for the error and a link function that correlates the average of the output variable to the linear combination of the input variables. Thus, the options for distributing the output are increased and may be part of the exponential family [Müller 2012]. It is a generalization of the ordinary least squares regression, where among its features one can mention the capacity to increase the separability frontier between the classes and decrease the training time. The models developed by *Nelder* and *Wedderburn* were synthesized based on others. Some specific cases of the GLM model are:

- Logistic regression;
- Analysis models of covariance and variance;
- Poisson regression;

Some studies in statistics are focused on the relational analysis of variables and can be seen as regression methods, where the output variable is tracked according to the explicit input characteristics. GLM emerged to address problems that the linear model proposed by Gauss and Legendre was unable to solve. Like the Linear Discriminant Analysis (**LDA**), this algorithm adopts an exponential distribution and is based on linear regression. Objectively, a distribution is exponential if its density can be described as follows in Equation (4) [Müller 2012]:

GLM 1: Exponential Family Structure

$$\mathrm{f}(\mathrm{y}|\theta, \phi) = exp\left\{\frac{yd\theta - b(\theta)}{a(\phi) + c(yd, \phi)}\right\} \tag{4}$$

where:

- yd is the target variable;
- The natural parameter is θ_n;
- The scale parameter is ϕ;
- The functions that determine the distribution in question are $a,\ b,\ c$.

To obtain the mean and variance of the variable of interest yd, just calculate the first and second-order derivatives of the function $b(\theta)$, according to Equations (5) and (6):

$$E(y) = \mu = \frac{\mathrm{db}}{\mathrm{d}\theta} \tag{5}$$

$$V(y) = -\frac{\mathrm{d}^2\mathrm{b}(\theta)}{\mathrm{d}\theta^2} a(\phi) \tag{6}$$

Therefore, GLM is formulated based on three components, described below:

1. A random component as described in (4) that has an exponential family distribution and is parameterized by θ and ϕ, so the variance does not have to be homogeneous and is relative to the vector mean;
2. A systematic component or linear model, $\eta = X\beta$, where X is the input matrix containing all observations;
3. And the function $E(y)$ that relates the expected value of the output μ to the linear component η.

2.1.2.7 Linear Discriminant Analysis - LDA

Linear discriminant analysis is a statistical technique capable of finding a linear combination between the characteristics that separate the objects, which are previously labeled (supervised), ensuring maximum separability between the classes [Tharwat et al. 2017]. It is used in artificial intelligence, mainly in pattern recognition. The algorithm can perform the dimensionality reduction and then make the classification. The method projects a set of features in a smaller space and allows for a desirable margin of separability without incurring overlap.

Ronald A. Fisher developed the LDA in 1936 (*The Use of Multiple Measurements in Taxonomic Problems*) [Varella 2017] and originally the LDA was used to solve systems involving two classes, and then the algorithm was generalized to various classes. The concept is similar to the analysis of variance and regression analysis, where a dependent variable is described as a linear combination of features [Tharwat et al. 2017].

Considering the input vector $\vec{x}$ containing the features and an output vector y, the LDA tries to find a predictor for the class y given an observation of $\vec{x}$, considering that the probability functions are normally distributed. Consider the probability functions of each class, their respective averages and variances [Cai et al. 2018, Shashoa et al. 2016], where Σ_k is the covariance matrix and $\vec{u}$ is the average:

$$p(\vec{x}|y = k) \tag{7}$$

$$(\vec{u}, \Sigma_k) \tag{8}$$

$$p(\vec{x}|y = k + 1) \tag{9}$$

$$(\vec{u}, \Sigma_{k+1}) \tag{10}$$

So the LDA applies a limit function based on the concept of Bayes (probability of an event happening based on a data a *priori*), and if the value obtained from this relationship is higher than a limit T the predicted class is the second, to the detriment of the first, the formula is as follows:

$$(\vec{x} - \vec{u_k})^T \Sigma_0^{-1} (\vec{x} - \vec{u_k}) + ln|\Sigma_0| - (\vec{x} - \vec{u_{k+1}})^T \Sigma_1^{-1} (\vec{x} - \vec{u_{k+1}}) + ln|\Sigma_{k+1}| > T \tag{11}$$

This relationship describes a particular case called quadratic discriminant analysis to arrive at LDA it is possible to simplify and assume that:

$$\Sigma_k = \Sigma_{k+1} \tag{12}$$

That is, consider that all random variables have the same variance (homo-cedasticity). Thus, equation 9 can be reduced by formula (13) for all k.

$$logP(y = k|x) = -\frac{1}{2}(\vec{x} - \vec{u_k})^T \Sigma^{-1} (\vec{x} - \vec{u_k}) + logP(y = k) \tag{13}$$

The term $(\vec{x} - \vec{u_k})^T \Sigma^{-1} (\vec{x} - \vec{u_k})$ corresponds to the distance of Mahalanobis [Maesschalck, Jouan-Rimbaud e Massart 2000] between a given sample x and the mean of the distribution. This allows us to know the distance between x and the class.

LDA is used in medicine to separate different groups according to the statistical correlation of the variables. For example, consider the problem of classifying a person as sick or not, according to the statistical relationship of several laboratory tests. Thus, the program provides an objective function capable of predicting a complex diagnosis. Figure 2.10 shows what the linear separation between three classes would look like after applying the method. As the algorithm does not need successive iterations for its use, it becomes fast and, therefore, is commonly adopted in the myoelectric signal classification.

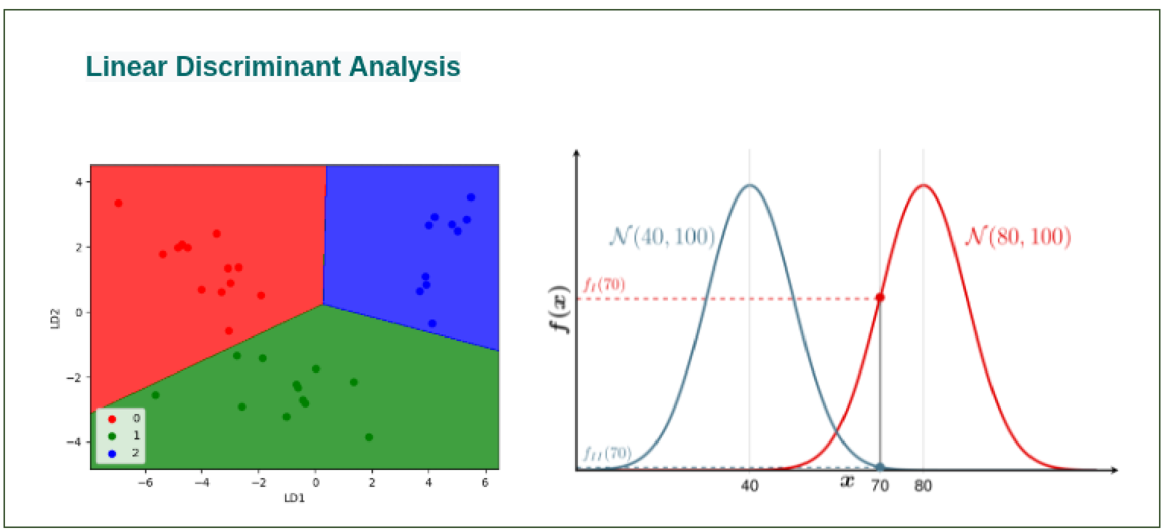

Figure 2.10 – Linear separation of three classes. Illustration developed from images at: commons.wikimedia.org.

The algorithm is applied when it is desired to separate classes in a linear way and is used to:

- Clustering;
- Anomaly detection;
- Reinforcement learning;
- Supervised learning.

2.2 RECURRENT NEURAL NETWORKS

When modeling involves sequential data, there is a class of neural networks recognized for their remarkable distinction in problems involving temporal dependence. Such networks are called Recurrent Neural Networks (RNNs) and are primarily divided into: Vanilla RNN, Long Short-Term Memory and Gated Recurrent Units (GRU). This work focused on LSTM-type networks as at the beginning of the research the option seemed to be adequate for the EMG classification problem. Although, other techniques such as GRU can be used in the same direction.

2.2.1 Long Short-Term Memory Networks

Classic ANNs have a limitation: they are unable to preserve information that has a long-term relationship. Variables that depend on old data and that have already been discarded in the training process are impaired. The LSTM algorithm allows to remove this limitation and the network to memorize the relations [Gers, Schmidhuber e Cummins 2000] no matter how long, as the network has many hyper-parameters it is necessary to create a protocol that permits its fine adjustment. Another problem that the algorithm solves is the Vanishing Gradients [Hochreiter 1998]. During the training and error minimization process, two unwanted situations can happen. The first occurs when the gradient obtained is very small. Multiplying the value by the learning rate results in a lower value and, consequently, the weight change practically does not change and, the training time degrades. In the second case, when the gradient is too high, the weights can extrapolate the optimal value. Thus, LSTM proposes to use a fixed scale factor, which avoids both situations.

LSTM networks were initially designed by *Hochreiter et al.* In 1997 in their original article [Hochreiter e Schmidhuber 1997]. To understand how the algorithm works, it is possible to make an analogy with thought, where the understanding of a sentence is based on the previous words. Likewise, the LSTM network has persistence and does not analyze only the current state but the entire context in which the problem is inserted. The architecture is composed of 4 fundamental elements: a cell, an entrance gate, an exit gate, and an oblivion gate [Yu et al. 2019]. The cell is responsible for storing and manipulating information. As data flows, they are modified and regulated by structures called gates. The entry gate allows new information to be added to the cell and uses a hyperbolic tangent function (*tanh*) for activation [Yu et al. 2019]. The exit gate is responsible for choosing the most relevant information to present for the next cell (*tanh*) [Lipton, Berkowitz e Elkan 2019].

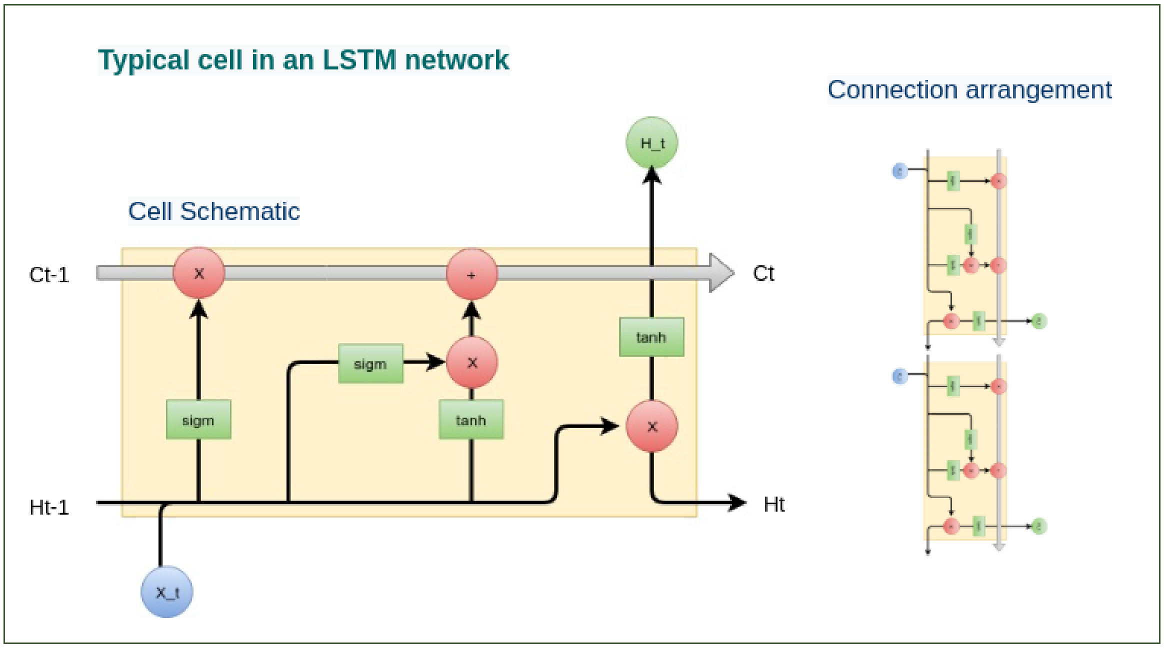

Figure 2.11 – Artistic design of an LSTM cell and a chain connection scheme between different units.

The forgetting gate is responsible for allowing or not allowing certain levels of information, so the cell can redefine its internal state from time to time [Gers, Schmidhuber e Cummins 2000].

Through a sigmoid activation function, the network can assign a certain level of prominence to the information, thus ensuring that only the most important data in the model can flow to the next cell. The concept is illustrated in Figure 2.11.

This technique of Deep Learning has gained prominence in problems of regression and classification. For this reason, it was one of the chosen ones to perform the recognition of patterns in the myoelectric signal. As classical neural networks are unable to maintain complete information, it was decided to assess the extent to which data was lost in the model and thus investigated the capacity of such an algorithm in the context of this research. To summarize its mathematical functioning we define the following variables [Greff et al. 2017, Lipton, Berkowitz e Elkan 2019, Yu et al. 2019]:

- x_t represents the input vector, where t is the cell number;
- h_t it is the hidden state of the cell;

- f_t it is the function that defines what information will be "forgotten" (eliminates irrelevant information);
- i_t updates the data from the current state;
- $\tilde{C}_t$ is a vector created by the hyperbolic tangent function to define the activation candidates of the cell;
- o_t represents the activation vector of the exit gate;
- c_t is the state vector of the cell;
- $\mathbf{W}$ is the matrix of weights, which can be input or output.

The forgetting gate calculates what data will be left behind. Using a sigmoid function it is possible to assign a level of relevance to the information (a value between 0 and 1). This step lists the hidden state of the previous cell, the current entry, and the weight vector as follows [Yu et al. 2019]:

$$f_t = \sigma(W_f * [h_{[t-1]}, x_t] + b_f) \tag{14}$$

Later, it is necessary to define what information the cell will store. This step is done in two steps. First, a sigmoid function will define, in a similar way to the previous step, what information the cell will aggregate. Then, you must calculate which activation candidates will be added to the state. Equations (15) and (16) describe this relationship [Yu et al. 2019].

$$i_t = \sigma(W_i * [h_{[t-1]}, x_t] + b_i) \tag{15}$$

$$\tilde{C}_t = tanh(W_C * [h_{[t-1]}, x_t] + b_C) \tag{16}$$

With that step, we are ready to define the new state C_t of the cell in terms of the previous state and the entry. Thus, formulas (14), (15), and (16) can be related as follows [Yu et al. 2019]:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{17}$$

Note that the forgetting function will select which information will be discarded by the network while at the same time adding new candidate values for the cell's state. Now it is enough to define the hidden state and, for that, it is necessary to be based on the current state. Again, the architecture performs a procedure to select what should be passed on, again using the sigmoid function.

$$o_t = \sigma(W_o * [h_{[t-1]}, x_t] + b_o) \tag{18}$$

Finally, the hidden state is calculated based on a hyperbolic tangent function, which will place the cell's state in a range (-1 and 1) and then multiply by the output activation vector calculated in equation (19). The mathematical relationship is described by [Yu et al. 2019]:

$$h_t = o_t * tanh(C_t) \tag{19}$$

The tangent function serves to regulate the values that are flowing through the cells so that they are always within a known range. The sigmoid function is used similarly in the context. However, it serves to eliminate irrelevant data since any value multiplied by zero will result in zero and, its component will be canceled by equation. Figure 2.12 shows the difference between the activation functions [Nwankpa et al. 2018].

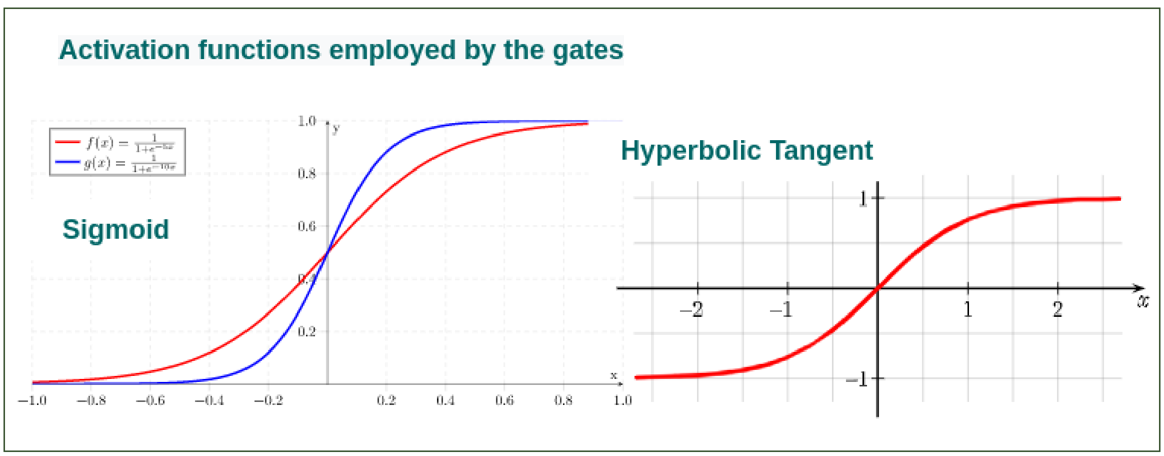

Figure 2.12 – Activation functions used by an LSTM cell. Illustration developed from: commons.wikimedia.org

The training process of the algorithm is done in a supervised way using the gradient method [Lipton, Berkowitz e Elkan 2019] and back-propagation. The adjustment of the weights is made based on the error calculated in each iteration.

2.3 CONVOLUTIONAL NEURAL NETWORK - CNN

For many years, researchers have adopted approaches to classifying the myoelectric signal based on classifiers that require data to be previously treated. Generally, research tracks EMG as a numerical sequence and, thus, one can explicitly extract its characteristics. The CNN network is an algorithm used in image recognition of patterns and has attributes that give it a high classification capacity [Zhou, Jin e Dong 2017]. In addition, CNN does not require the previous extraction of characteristics. The research tested the network on two aspects. First, the algorithm recognize the pattern using the signal image and not its sequence in time. The network will be delegated the task of automating the extraction of characteristics [Liu 2018], then the author introduces a new method, where the network is not fed with the image but with features extracted from the signal. Thus, it will be possible to verify the feasibility of composing the extraction process and balance the accuracy and training time.

The natural concept that inspired CNN is similar to the visual system of cats (conceptually) [Liu et al. 2015]. The image that arrives at the animal's cortex is superimposed on others in the sequence, and this allows the algorithm to filter these images. The CNN network simulates this mechanism through the [Indolia et al. 2018] convolution operation and is thus able to identify complex patterns, even under adverse conditions as in the case of insufficient shuffling and lighting [Li et al. 2018]. CNN was designed to learn the relationship between input and output, storing its knowledge in the weight of the filters in each successive layer. The CNN algorithm shares some similarities with other neural networks, as it also has neurons and weights but, its architecture is different. The first network of the type was proposed by *LeCun et al.* at work [LeCun et al. 1999], and its typical structure consists of an input layer, hidden layers, and an output layer.

CNN's, in general, are described by convolutional layers, pooling, and fully connected layer [O'Shea e Nash 2015]. The **convolution layer** is similar to a typically hidden layer. The difference is that they work as a filter through geometric patterns that slide over the image during layer convolution. Thus, the layer maps the features in a more abstract way [O'Shea e Nash 2015]. The filter or kernel that slides through the image is made up of randomly initialized weights and, at the end of the operation, the result is saved in the activation map. Thus, the characteristics are extracted automatically by the network [Emmert-Streib et al. 2020], [Szegedy et al. 2015]. This process allows to capture the spatial and temporal dependence, in addition, the number of parameters is reduced, and the weights can be reused.

The work detail its operation now. Imagine an input image with the dimension of 6x6x1. We must select a value for the kernel and another for the operation step (Stride). We will assume that the step is 1 (Non- Stride) and the kernel is 3x3x1. Figure 2.13 exemplifies the process of sliding the kernel over the image, which allows the convolution to be performed and results in the mapped feature. Consider a grayscale image (the concept can be expanded to any shade - RGB). First, the figure must be decomposed in grayscale so that each pixel has a level of information. The process takes place as follows:

a - Each pixel in the image is assigned a value between 0 and 255, where 0 represents white, and 255 is assigned to black;

b - The 6x6x1 image is decomposed into 36 pixels with the values previously assigned, and then the network receives this entry;

c - Different types of kernels are applied to the images. Each of them representing a pattern that you want to recognize (rectangles and squares). In this step, the geometric patterns are simple;

d - The characteristic patterns are recognized in the first layer and propagated to the subsequent ones;

e - In the following layers, more complex kernels are used, such as circles, curves, borders, and others. In this way, it is possible to identify details of the image, such as a mouth or eye;

f - The last kernels, when activated, allow the recognition and classification of the complex image through simple patterns.

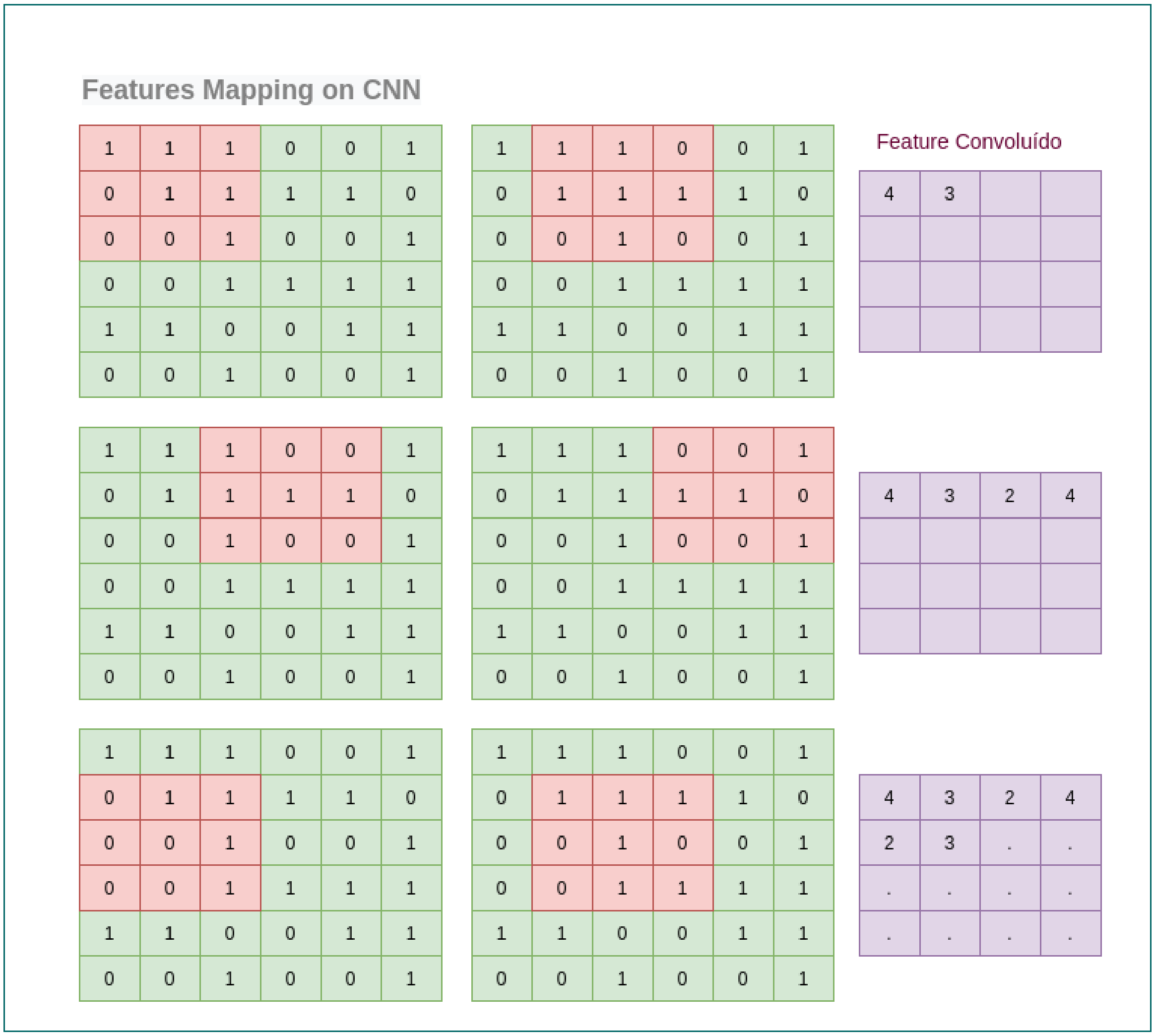

Figure 2.13 – Convolution operation and its respective features mapping.

The mathematical manipulation of the convolution is simple and is denoted by the operator *. Consider the operation described in Figure 2.14.

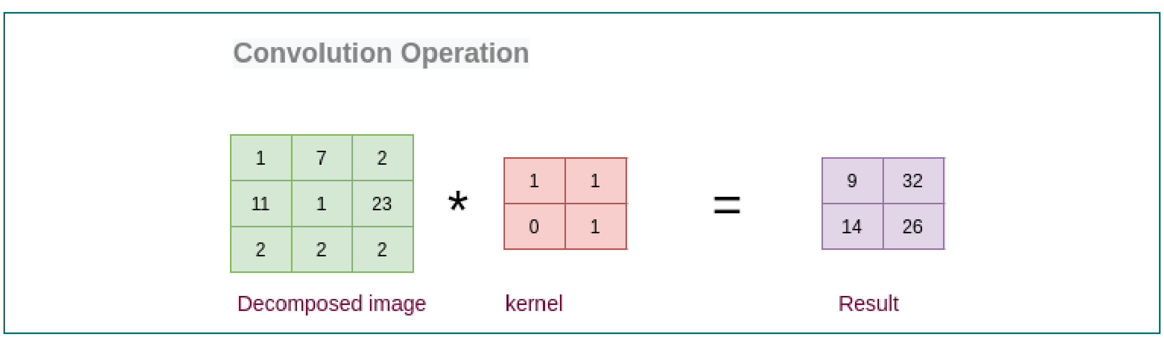

Figure 2.14 – Convolution of pixels with the chosen kernel.

The filter runs through the samples in the image, performs the multiplication by the elements, and then the values are added together. In mathematical expressions, we have:

Convolution 1: Step by step of the process.

(1x1 + 7x1 + 11x0 + 1x1) = 9
(7x1 + 2x1 + 1x0 + 23x1) = 32
(11x1 + 1x1 + 2x0 + 2x1) = 14
(1x1 + 23x1 + 2x0 + 2x1) = 26

Next, we have a layer called **ReLu** (Linear Rectified activation function) that is employed to rectify values. Many neural networks adopt it as a standard activation function, as training is simple and generally achieves good performance [Agarap 2018]. If the input is positive, the output will receive the same value. Otherwise, the assigned value will be zero [Agarap 2018]. Remember that in each node, the inputs are multiplied by the weights and then added together, then the activation function is applied to that node, which assigns the effective output value.

The simplest way to implement this step is to use a linear function, but this does not allow the network to obtain a complex mapping. In this case, a non-linear manner such as sigmoid and hyperbolic tangent improves the model. However, such functions incur the saturation problem [Rakitianskaia e Engelbrecht 2015], where large values always receive one and small receive zero or minus one (tangent and sigmoid). In addition, they are sensitive to values that oscillate around the mean. When saturation occurs, the network is unable to adapt the weights and increase performance. Thus, the error is propagated, and relevant information about the gradient is discarded by the network (leak gradient) [Hochreiter 1998].

The strategy to mitigate the problems described is to use a similar function to a linear one. However, it is a non-linear function that allows the complex relationship and reduces saturation. With this approach, the backward gradient method can be applied to deep networks and permits training to be efficient [Ide e Kurita 2017]. ReLu was an essential step for deep learning and allowed to considerably increase the complexity of such systems without incurring the problem of saturation. This technique is illustrated in Figure 2.15.

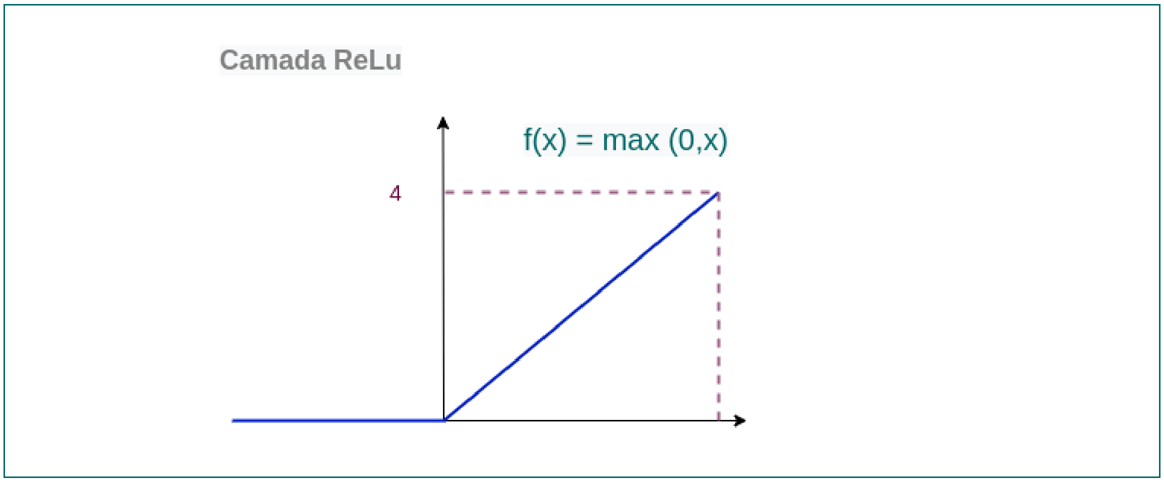

Figure 2.15 – The geometric conception of ReLu and its mathematical formulation.

The next layer of the architecture is known as Pooling. Its function is to reduce the number of parameters, especially when the image is large [Dominik, Andreas e Behnke 2010, O'Shea e Nash 2015]. Through a subsampling process, the layer can reduce the dimensionality of previously learned maps and still retain information. There are three possibilities to perform this step: Max Pooling, medium Pool, and add Pooling. The layer runs through each entry the size of a specified window called pool [Dominik, Andreas e Behnke 2010];

The last layer in a conventional CNN structure is Totally Connected. This structure is responsible for generating the final result from the Features previously learned [Szegedy et al. 2015]. In practical terms, the layer is nothing more than a traditional layer of a neural network [O'Shea e Nash 2015]. Consider the 2D transformation of the output Features to 1D as follows.

$$\begin{vmatrix} 5 & 7 \\ 8 & 13 \end{vmatrix} \rightarrow \begin{vmatrix} 5 \\ 7 \\ 8 \\ 13 \end{vmatrix} \tag{20}$$

The network first formats the output so that the pattern can proceed to the Fully Connected layer, where then two transformations take place: one linear and one non-linear. The linear transformation is described by the formula (21) [Singh 2020]:

$$Z = W^T * X + b \tag{21}$$

In this case, W represents the vector of weights, which are started randomly, X is the vector Features, and b is an associated trend (bias). The number of weights will be proportional to the number of neurons in that layer. The complete process, considering a Fully

Connected layer with two neurons, is described by Figure 2.16.

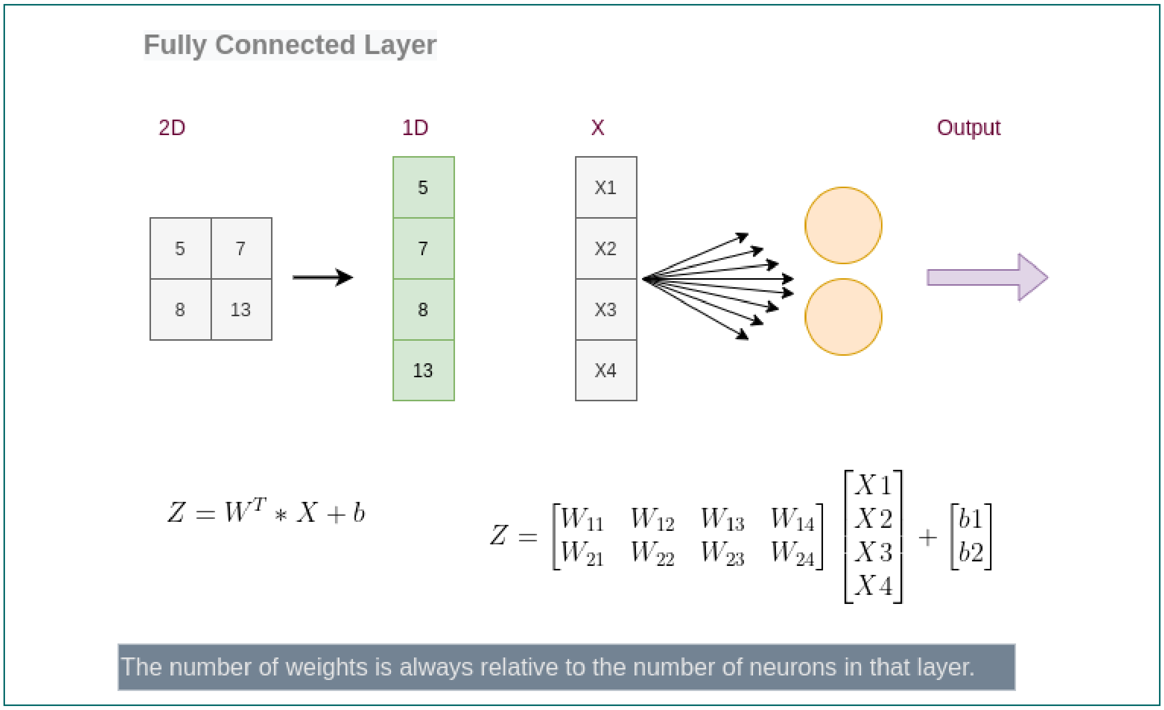

Figure 2.16 – Mathematical scheme carried out in the last layer of the architecture.

Then it is necessary to make a non-linear transformation, and for that, an activation function is used, as in the case of LSTM, there are many possibilities of choice. One of the most used is the sigmoid, described below.

$$Z_2 = \frac{1}{1 + e^{-x}} \tag{22}$$

In 1986, *Rumelhart et al.* published an article that described an elegant way of training a neural network [Rumelhart, Hinton e Williams 1986]. The technique called Backpropagation uses gradient descent to update the algorithm weights and ensure convergence. This fact occurs through two steps or passages (forward and backward), where the algorithm can calculate the error gradient of each parameter. In practical terms, it happens as follows:

1. The first forecast of the network is not accurate, because the weights are random at the beginning;

2. To update the weights, it is necessary to know the direction (increase or decrease). That is, from the obtained point and the expected one, it is possible to distinguish the gradient of the curve;

3. If the gradient is negative, the parameters will be increased and vice versa. The learning

rate controls how prominent the increment will be;

After the network chooses the initial values for the weights and calculates the error, the differences must be sent backward, mathematically the gradient is obtained as described by [Singh 2020]:

$$\frac{\partial E}{\partial f} = \frac{\partial E}{\partial O}\frac{\partial O}{\partial Z_2}\frac{\partial Z_2}{\partial A_1}\frac{\partial A_1}{\partial Z_1}\frac{\partial Z_1}{\partial f} \tag{23}$$

The first term of the equation is found by differentiating the value calculated by the network and the actual value. That is, it is the error between the input and the output. This step takes place in the fully connected Layer, as shown in (24).

$$\frac{\partial E}{\partial O} = -(y_d - O) \tag{24}$$

Then we have to calculate the derivative of the activation function (sigmoid - in layer Z_2) about output O, described by (25).

$$\frac{\partial O}{\partial Z_2} = \left(\frac{1}{1+e^{-x}}\right)' \tag{25}$$

The next derivative is concerning changes in Z_2 according to the weights W, equations (26) and (27) detail this relationship.

$$\frac{\partial Z_2}{\partial A_1} = \left(W^T * A1 + b\right)' \tag{26}$$

$$\frac{\partial Z_2}{\partial A_1} = W^T \tag{27}$$

Again, we have to derive concerning the activation function. In this case, it is the function of the respective convolutional layer and not the output layer, according to (28).

$$\frac{\partial A_1}{\partial Z_1} = ReLu(Z1)' \tag{28}$$

The last derivative is due to the variation of Z_1 concerning the filter, and formula (29) summarizes the step.

$$\frac{\partial Z_1}{\partial f} = X \tag{29}$$

And so, from this gradient it is possible to update the weights according to rule (30):

$$Weight_{New} = Weight_{older} - (learningrate * Gradient) \tag{30}$$

The layers and their respective mathematical concepts can be graphically presented in Figure 2.17.

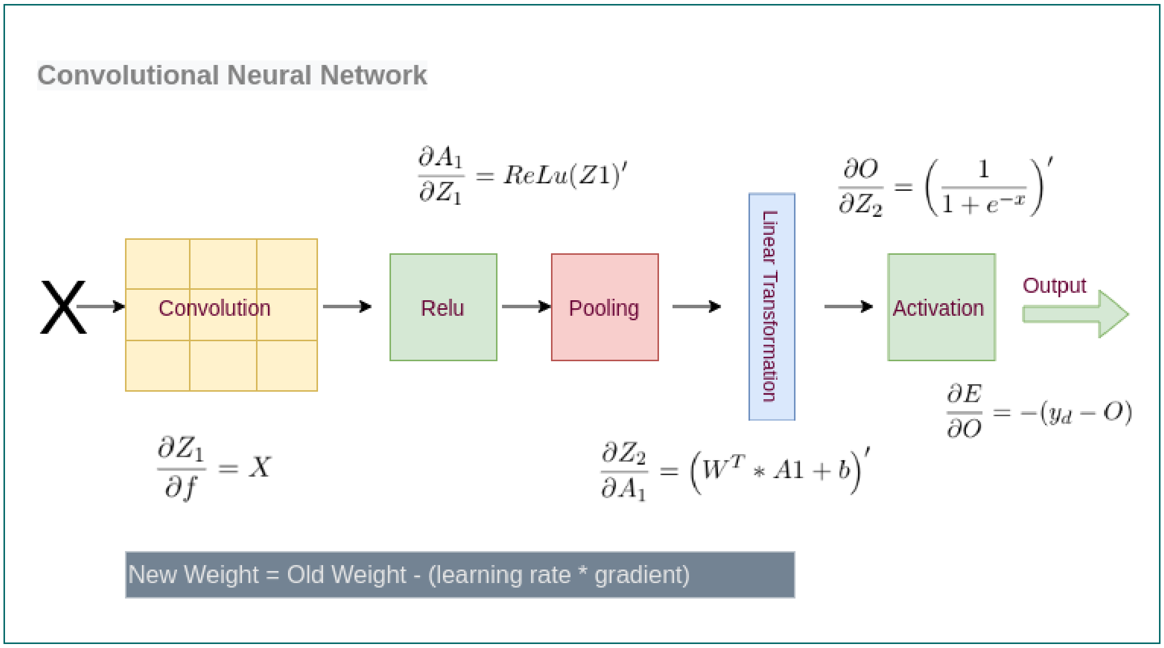

Figure 2.17 – Main layers of a CNN network and their respective mathematical definitions for the formation of the descending gradient.

2.3.1 Hyperparameter

A hyperparameter is an indicator or an internal control variable of the network that acts on the learning process [Jia et al. 2019]. That is, it is a parameter that allows when developing the fine-tuning of the algorithm. They are important because they permit knowing the influence of each one in the general behavior of the classifier. Second, they make it possible to adjust the metrics like time, accuracy, and cost. Each network has its own set, and some hyperparameters are common among them. The following is a list of the main ones:

Learning Rate: This hyperparameter is the adjustment increment in the weights at each iteration. Learning Rate is directly related to the convergence speed to the descent of the gradient [Yu e Zhu 2020]. Low rates make training slow, and when they are too high they can cause the model not to converge, as some regions of the solution space will not be reached. Figure 2.18 shows the four most common situations. Some workers adopt a scheme where the rate is not constant during the evolution of training [Yu e Zhu 2020], generally a higher value is used in the initial stages, and the proportion is then iteratively decreased, in an intuitive way, this ensures convergence while accelerating the training process.

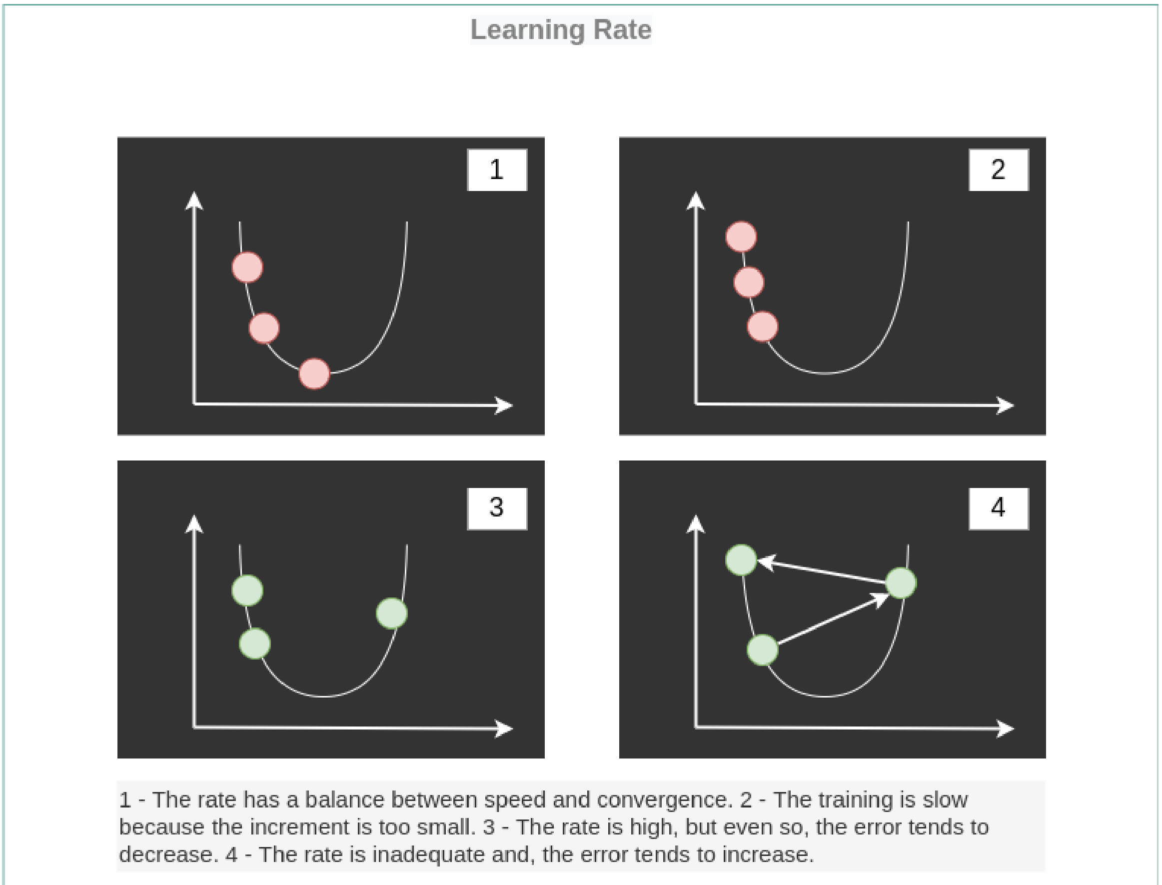

Figure 2.18 – Impact of the learning rate on the training process.

Number of Epochs: Denotes how many times the weights will be updated [Carney e Cunningham 1998]. That is, how many times the learning algorithm will be passed over the entry. This parameter is directly linked to the learning rate because if it is small, many times will be necessary (the contribution of the rate is low with each iteration). This factor is also linked to the over-training problem, as many times can increase the error value in the test set. One way to know if the choice was correct is to do a plot of the training evolution and check from which point the validation error starts to diverge from the test error.

Batch Size: This hyper-parameter relates to speed and the number of iterations required for convergence. It defines the number of samples delivered to the network in an iteration [Kandel e Castelli 2020]. It is customary to present only one sample of the set at a time, calculate the error, back-propagate and adjust the weights. Today it is possible to choose the lot size (or use it as a whole) and calculate the gradient using the error generated by all samples. A high batch implies an excessive use of computational resources and reduces training time, and a small batch slows down the process. However, it decreases the chance of the solution falling to a local minimum. For this parameter it is common to choose values based on the power of two (1, 2, 4, 8, 16, 32, 64, 128, 256) [Kandel e Castelli 2020].

Hidden Layers and Units: There is no mathematical consensus on this issue and, each problem can have its solution. The simplest method to choose is based on a heuristic, brute force, and luck. If the problem is simple, few units can solve it. If it is complex, a more dense architecture will be necessary and, this fact may incur the obstacle of over-training. In addition, the training time tends to increase as new layers and neurons are added.

2.3.1.1 HyperParameter Adjustment

With so many parameters that can be tested and their respective combinations, the adjustment process becomes an exponential problem [Wang et al. 2018], at least from a mathematical point of view. There are ways to search, and they are based on two principles: brute strength and experience of those who develop [Wang et al. 2018]. The programmer's knowledge helps to diminish the dimension of the problem and thus reduce the number of tests. For example, nobody who dominates the subject would test the network for the number of neurons using a unitary increment. Instead, it is preferable to increase it by 10 10 or even 50 out of 50, depending on the initial results. Thus, to minimize the number of attempts that will not increase the results, it is necessary to set up a test configuration that is, at least, objective. The main methods are:

- **Grid Search:** This method is the exhaustive search itself [Liashchynskyi e Liashchynskyi 2019]. The algorithm will scan the n-dimensional space of the parameters and use a function that calculates the score for each configuration. Then the method will employ a scheme to perform cross-validation between the results and, thus, define the candidate who obtained the highest score;

- **Random Search:** random combinations are tested. But in this case, the solution space is not scanned completely [Liashchynskyi e Liashchynskyi 2019]. Instead, the methodology uses several tests a priori. Let's suppose that there are 10,000 possible combinations in a schema in *Grid*. What the random search does is select arrangements within that space. However, it only tests a pre-defined number of possibilities (a thousand, for example). This step decreases the search time but does not guarantee that the best configuration is chosen, although the heuristic has shown that the results are satisfactory.

2.3.2 Features Engineering

The Features or characteristics are what the name says, the set of information that describes the system. That is, they are the inputs of networks capable of discriminating against a class. Feature Engineering is a branch of AI that aims to study ways of obtaining information in a more objective way [Ghojogh et al. 2019]. Like other fields of machine learning, the reader will realize that there is not a set of rules well defined within this field, as well as natural intelligence is the result of various interactions, according to the problem.

Generally, the process is done in two steps. The first step is based on heuristics and depends a lot on the experience of whoever builds the solution. For example, it is possible to extract characteristics automatically or in advance, and each one of them fits into a context. Thus, who will decide which concept to use will be the programmer. The second way is to use a set of consolidated techniques, which allow this step to be carried out with due mathematical elegance. For example, knowing that the data is at different intervals and that most networks are sensitive to this difference, you can normalize the input.

This model and other techniques will be discussed in this section, and its understanding is fundamental for the development of this thesis because one of the objectives is precisely to join two concepts of this topic that are antagonistic to some extent: using networks that perform the automatic extraction of characteristics along with classic methods extraction of features.

The initial motivation for this step is to model the network input so that the data is not noisy [Hira e Gillies 2015]. It's AI's rough work, most of the computation stays in this step, and the explanation for this is simple: if the data describes the original relationship between the input and the output, the search for the network capable of solving the problem can be done in an instant (since it is possible to test a lot) and training is not a problem for many approaches. Therefore, before testing a model, it is necessary to create its architecture with rigor and know-how to relate the available information according to the objective that it intends to achieve.

2.3.2.1 Feature Scaling

The first strategy is one of the best known and is a mandatory operation component of several objective functions. That is, the operation of some networks is sensitive to this step [Wan 2019]. This process adjusts the scale of values of features. The idea is simple each characteristic should have a balanced level of relevance. Thus, the individual contribution of every resource should be in the same range. In this way, the proper relative proportion between components will ensure that one attribute is not minimized to the detriment of another. In addition, the technique allows the gradient method to converge more quickly.

The practical concept is as follows: networks assume that high numbers will be more relevant, and this is due to the Euclidean distance function that they implement. It not always the case, sometimes, it is preferable that all data is equally considered [Singh e Singh 2019]. For example, consider a system whose inputs are the price and the quantity in stock of a product. If there are hundreds of items, the network can conclude that the amount in stock is more important than the price, and that would lead to a solution space that does not correctly reflect the dynamics of the modeled system. For this reason, this step is justified. We must ensure that the network does not assume any premise and that each feature has its degree of relative importance normalized [Singh e Singh 2019].

The two most common techniques are **normalization** and **standardization** (Z-score normalization). In the first technique, the characteristics are scaled in a range of values, something like [-1, 1] or [0, 1]. Normalization is a good choice when the standard deviation is small and the distribution is not Gaussian. The necessary mathematics is simple and is presented by equation (31) [Singh e Singh 2019].

$$X_{Norm} = \frac{X - X_{Min}}{X_{Max} - X_{Min}} \tag{31}$$

The standardization assumes that the features have a normal distribution, with zero mean and variance equal to one, and, therefore, it is a good choice if the input is normally distributed. In addition, some networks are sensitive to this step. For this, the sample's z-score is calculated as follows [Singh e Singh 2019].

$$z = \frac{x - \mu}{\sigma} \tag{32}$$

where the mean and standard deviation are described by (33) and (34):

$$\mu = \frac{1}{N}\sum_{i=1}^{N}(x_i) \tag{33}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{34}$$

The choice of the method is not always simple, and there is no single answer. Like other AI fields, this choice depends on prior developer knowledge, heuristics, and a solid statistical base. For example, you can hear that most classifiers work better or only work with standardization, but consider the case of the CNN network, where the entry is through the pixels of an image, and these generally have values that range from white to black. Thus, it would be more appropriate to use a maximum and a minimum range. You can make the following statements about the behavior of some networks concerning their sensitivity to normalization:

- The Linear Discriminant Analysis is not affected by this step, and the network has its own means of dealing with the situation, in which case it may not have much effect;

- The MLP network and any other algorithm based on gradient descent (Vector Support Machine, Neural Networks, and Perceptrons) are sensitive to the scale of the [Wan 2019] entry;

- Algorithms based on decision trees also do not require this resource, as their implementation does not depend on Euclidean distance;

- Clustering methods like K-nearest need this step [Singh e Singh 2019], as their objective function is based on Euclidean distance.

Geometrically, Figure 2.19 exemplifies the two concepts.

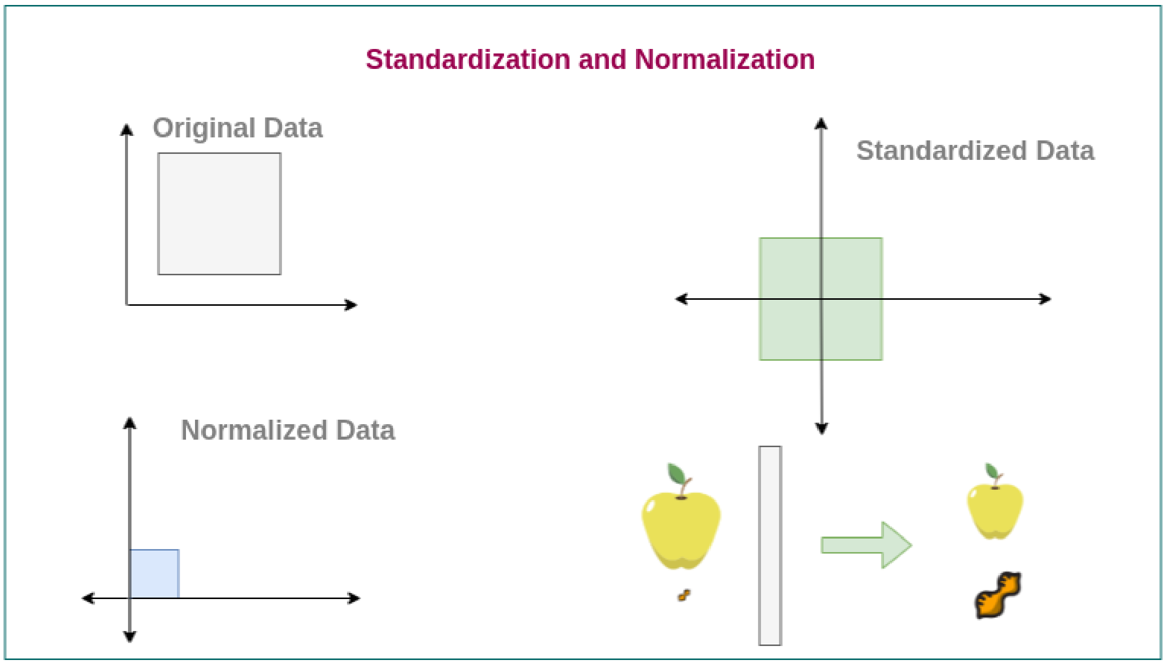

Figure 2.19 – Difference between the two most common techniques and a practical example of how input is processed.

2.3.2.2 Imputation

Imagine the following situation. The prosthesis collects the myoelectric signal through 8 electrodes and, suddenly, due to a displacement, contact with two of them is lost so that several features that originate from that signal will be absent. This scenario is one of the most common problems during data preparation and arises from various situations, so it is necessary to apply measures capable of circumventing the occasion. The simplest answer is to discard this observation by excluding the line representing the sample. But this is somewhat limited since in a real-time device the predictions must be constant.

A more sophisticated solution would be to replace these values with others that are at least representative. In the case of a regression system, you can change the missing sample with the last value of your moving average. In classification models, the point could be replaced by the average, mode, or median [Zhang 2016] of the column or be marked with a constant value. In addition, it is necessary to present the network with these data in the training process. The network must be evaluated with this artificial data, as the nominal model will be exposed to this type of situation and must deal with it. There are other ways to mitigate this situation, such as extrapolation, interpolation [Xu, Tao e He 2010] and even use AI techniques dedicated to this task. Again, there is no general equation, and the solution will depend on the model implemented.

2.3.2.3 Outliers or Spurious Values

Another problem that may occur is related to the presence of Outliers in the sample. These data are values that do not represent the behavior of the series and mess up the statistical measures [Escalante 2005, Paulheim e Meusel 2015]. These measures hinder the training process and can lead the functional model to unknown domains. In the case of the prosthesis, imagine a situation of high impedance that can take some samples to values that the network is unaware of, the operation of the application would not follow the pattern, and the model would be discontinued. But what if these Outliers are inherent parts of the system? In that case, you need to find them, and the easiest way to do that is through visualization. Using histograms it is possible to have an idea about the data distribution and check if many values deviate from the other observations. The sources of these deviations are the most varied [Escalante 2005]:

- Problems with measuring instruments;
- Errors during the processing of features;
- Human errors in the description of the data;
- Intentional and natural.

On the other hand, sometimes neural networks are modeled to detect Outliers [Xu et al. 2018]. That is, the system wants to predict whether a sample is outside the natural standard. The classic cases in this sense are fraud and invasion detection systems [Xu et al. 2018], although other domains may employ the technique, as in the case of regression systems that seek to reverse the trend in the curve.

One of the statistical ways that scientists used to measure these deviations is the standardization of the data because through the z-score it is possible to know how far the measurement is from the parametric average. Figure 2.20 exemplifies the technique that, although simple, is powerful and widely used cite Out4. There are also non-parametric methods that models employed in this regard, like Spatial Clustering of Noisy Applications (DBSCAN) [Ghallab, Fahmy e Nasr 2019]. Systems based on cluster allow to visualize and thus better understand the data. The method uses the frequency of the neighborhood [Senthilkumar e Metilda 2016] of an area delimited by a circle or sphere. The groups are classified into three fields according to the density of the point, the central region is called A and is the densest, then we have the border region or region C, which has a lower density, and finally, the region of Outliers, which is not densely connected and does not belong to any cluster. Density refers to the number of points a region has and that region is bounded by the radius defined by the developer.

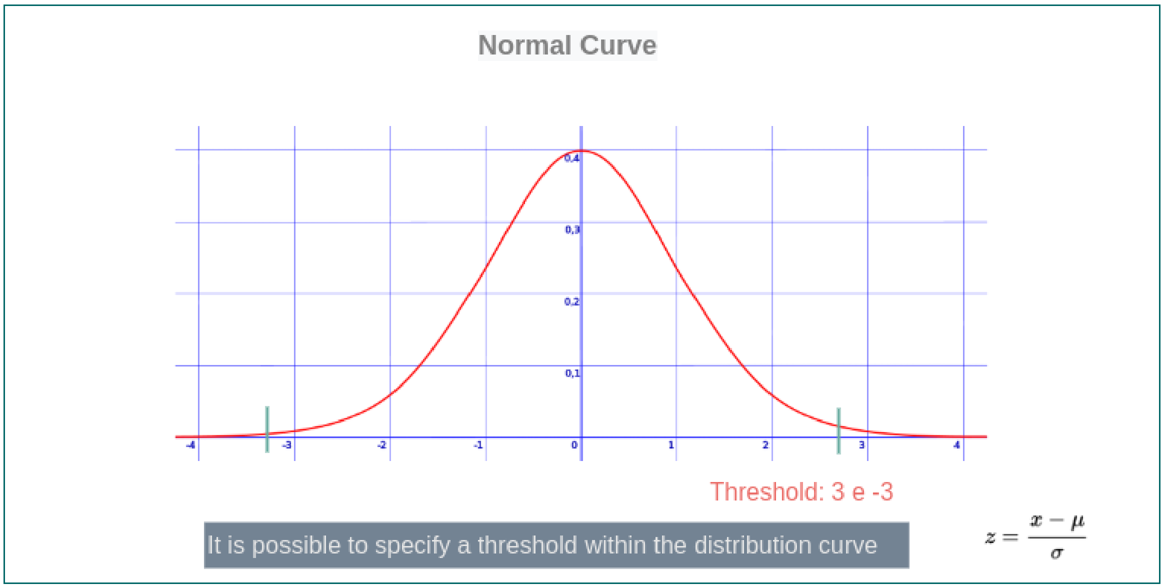

Figure 2.20 – Use of the standardization method to identify deviation in a sample, considering the mean and a limit value.

Although it is possible to make a more detailed mathematical definition of the method, the subject is somewhat beyond the scope of the Thesis. It is also important to note that other techniques with the same purpose are available in the literature.

2.3.2.4 Logarithmic Transformation

The features do not always follow a scale of linear values, and the magnitude of the data can vary widely. For example, consider the price of an asset that has appreciated a lot in recent years and follows an exponential scale. Doing a logarithmic normalization, in this case, would help bring the distribution closer to the normal [Feng et al. 2014] and facilitate the network learning process. Another effect of this procedure is to reduce the interference of outliers, as it normalizes the magnitude difference. This transformation can only be done with positive values and is generally used in problems that involve a non-linear variation.

Figure 2.21 presents the highest benefit of this technique (approximating the distribution to normal).

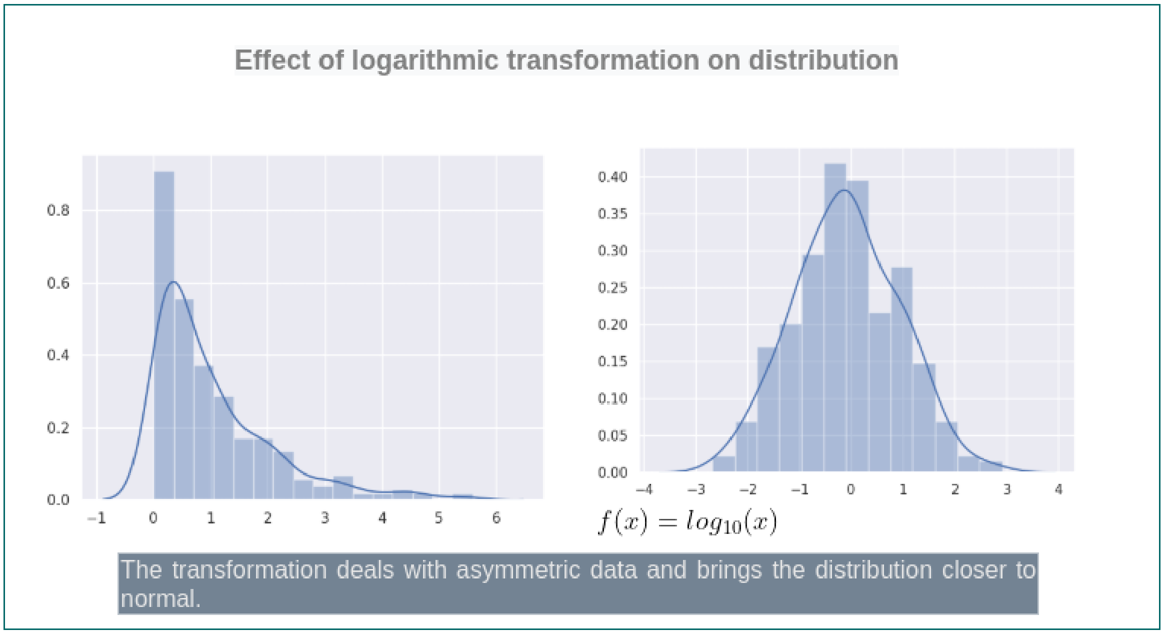

Figure 2.21 – Effect of logarithmic transformation on sample distribution.

2.3.2.5 Feature Creation

Features are not a natural part of a system and must often be created by developers using consolidated techniques or creative processes. Let's consider the case of Myography, wherefrom the electrical signal. It is possible to derive dozens of characteristics. One of the first advances made in the field of prosthesis development was precisely in this field when in 1993, *Hudgins et al.* [Hudgins, Parker e Scott 1993] proposed a group of 4 features capable of increasing the separability boundary between classes and thus allowing a more accurate prediction method. Since then, there has been a high increase in the number of characteristics proposed by the literature, and many studies are still being done in this sense [Phinyomark, Khushaba e Scheme 2018] because if the data present the nonobjective relationship that is hidden, the network will be able to find it.

The methodology used to create a feature is quite broad, based on several aspects, such as understanding the model, mathematical knowledge of the domain, creativity, and heuristics. For example, a method in the case of EMG is to use the mean square value of the wave (RMS), as it is related to the strength and fatigue of a contraction [Arjunan e Kumar 2010], so this characteristic brings valuable information to the network within the spectrum of the application. Now consider using the same feature in a call center system. It would make little or no sense. The methodology will present all the features that the thesis uses in the research and its mathematical formulation.

2.3.2.6 Feature Selection

It is easy to think that to obtain a good model it is enough to supply the network with a huge amount of characteristics, as long as they do not have outliers and are scaled. However, this is a common mistake that can lead to some unwanted situations. The first of these is the training time, which will drastically increase if this strategy is preferred. Second, the network may incur the adversity of over-training and create a non-representative model. And finally, most of the information can be redundant or even non-informative [Venkatesh e Anuradha 2019], which can lead to unnecessary expenses with computational resources. Thus, removing irrelevant resources is an essential step and ensures that the network only uses the variables that make the greatest contribution. In addition, it is possible to gain a broader understanding of the process that generated the data when variables are carefully selected.

The first method which the bibliography can cite is called **Filtering** [Sánchez-Maroño, Alonso-Betanzos e Tombilla-Sanromán 2007], where a statistical correlation is performed between the measures to remove those that do not contribute to the process. It analyzes sample distances, their dependence on other variables, and the consistency of the information. For this, statistical algorithms are used, such as:

- **LDA**: This network, besides performing the classification process, is also used to find the linear combination of the characteristics [Cui e Ji 2009] that can separate the classes objectively;
- **Pearson's Correlation**: It measures the linear dependence between two continuous variables, and its value varies between 1 and -1. An absolute correlation between the variables is worth 1, a negative relationship is worth -1, and if there is no linear dependence between them, the value is zero [Blessie e Karthikeyan 2012]. Its mathematical formula is as follows (35);

$$\rho_{X,Y} = \frac{COV(X,Y)}{\sigma_X \sigma_Y} \tag{35}$$

- **Chi-square**: It is a way of calculating the probability of statistical correlation between the characteristics and, for that, it uses the frequency distribution.

Another way to perform this procedure is through the Wrapper method. Although its computational cost is high, it is common to employ such a routine [Ibrahim et al. 2018]. The process consists of researching the best possibilities and is, therefore, a brute force method.

The programmer successively trains the network with several sets of data. At the end of each iteration, the designer saves the result and compares it with the previous ones. This strategy can be implemented in the following way [Ibrahim et al. 2018]:

- **Direct Selection**: Tests start with just one feature and then they are added incrementally;
- **Reverse Elimination**: It is the opposite of the previous item, the model starts with all features, and in sequence, they are eliminated iteratively. This approach may incur time overhead;
- **Recursive Elimination**: The algorithm looks for the best characteristics and, for that, it separates them into different groups according to their relevance in the tests. The steps are made recursively until all the features have been tested.

And finally, there is the Embedded Method that combines the features of the previous two. Some algorithms have their means of selecting characteristics and, for that purpose, use regularization techniques [Kim, Schug e Kim 2015]. Regularization is a strategy that, in practical terms, prepares or problem of excessive training and allows the creation of a generic model. A network penalizes the coefficients of a subset of features [Kim, Schug e Kim 2015]. Thus, the network can perform an iterative search process and still ensures that the local problem decreases. Two examples are:

1. The Lasso regression employs L1 regularization, where the penalty rate is equal to the absolute value of the coefficients;
2. The Ridge algorithm performs a type of regularization called L2, which penalizes the magnitude of the coefficients in a quadratic way.

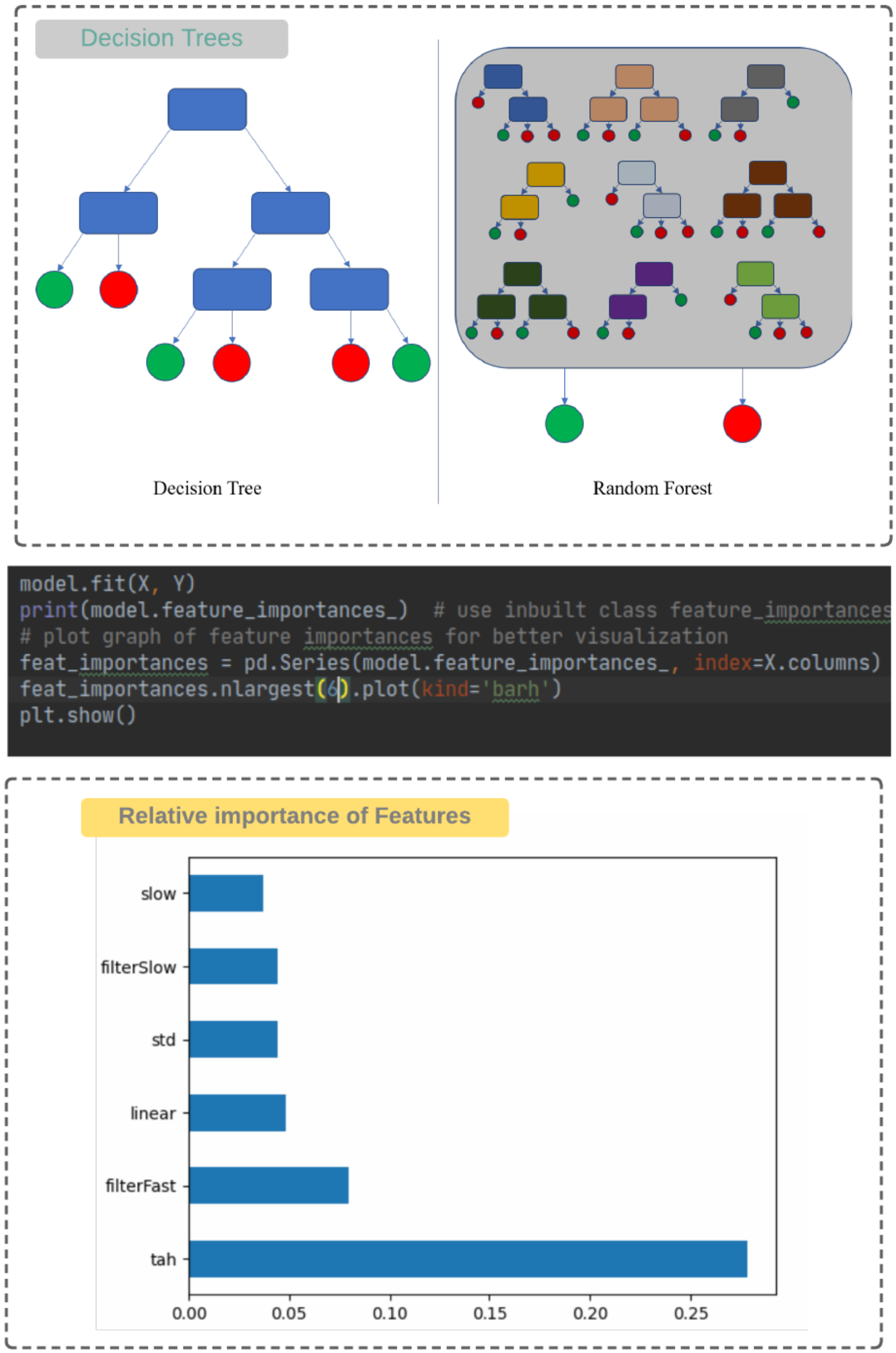

Figure 2.22 – The relative importance of Features given by a decision tree. Illustration made from a figure obtained in *Creative Commons*.

The method trains the network and manages to give importance to features. Then it removes those that do not contribute or have a small weight. It is possible and interesting to observe the relevance of each characteristic. This step helps the designer to have an idea about the model. For example, consider the case of the myoelectric signal, imagine that the developer wants to know which domain (time or frequency) is most effective.

For that, it feeds an algorithm capable of creating a ranking of importance. That is, to solve an AI problem, an AI technique is used, which allows the programmer to better understand the data relationship. Many algorithms can accomplish this task, and an example is in Figure 2.22.

2.3.3 Dimensionality Reduction

We live in the information age, and society produces countless data every day, but most of it is just garbage, and it is necessary to treat this raw volume to find valuable information. There is a field of research called data mining, and as the name says, it aims to find something of value in a given collection. As the amount of information is excessive, it is necessary to eliminate everything unnecessary and irrelevant, especially considering that there is a tendency for the database to grow horizontally (Curse of Dimensionality) [Altman e Krzywinski 2018], implying high processing and computational costs.

One of the problems with features is that they can be multi-collinear, that is, when they have a high degree of collinearity, staying thus linearly related [Dormann et al. 2013]. This fact directly impacting the least-squares [Dormann et al. 2013] method and can make solving the equation system impossible. Identifying characteristics that present this relationship is relevant for two reasons: it allows speed to the training process and allows the developer to have an idea about the guarantee of convergence of the model, since, if all inputs are multi-collinear, the equation system will hardly be satisfied.

Thus, it is desirable to reduce the dimension of the problem and still maintain the representativeness of the original data. This step is a particular case of the selection of features. There are many ways to achieve this principle, but this work will focus on Principal Component Analysis (PCA), as it is one of the most common and has been used in the study methodology.

The PCA algorithm is a statistical tool that makes a transformation that takes a multivariable data set to a different one, but that can explain the same problem [Morchid et al. 2014]. From a set of n variables you get another called m, where $m < n$, called the main components [Mishra et al. 2017]. The components are listed according to the variance, and each of them must be orthogonal to the others. This process eliminates the problem of correlation and allows a more elegant description of the data. The method's only constraint is the need to scale the *features* first. The steps for calculating the components from linear combinations of the original variables are [Mishra et al. 2017]:

1. Get the vector samples with dimension n, that is, the features;
2. Calculate the average of the data;
3. Subtract the average from all samples;
4. Use subtractions to calculate a covariance matrix. The matrix is the average of the product of each subtraction by itself, with dimension nxn;
5. Calculate the eigenvalues and eigenvectors of the previous matrix;
6. The eigenvector with the highest eigenvalue corresponds to the main component and so on until the other *m-1* components. Thus, the main axis 1 has the highest variance, 2 has the second-highest, and so until the last component.

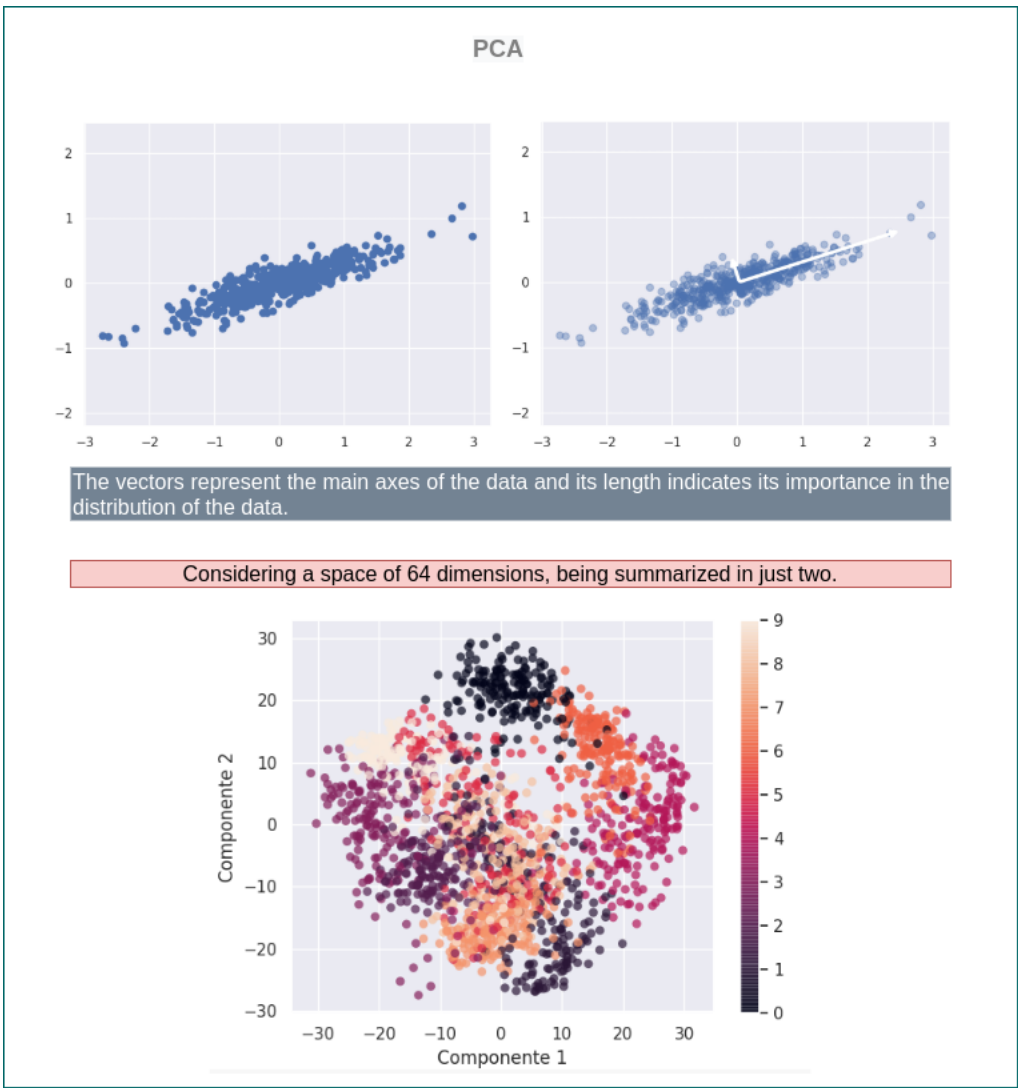

Figure 2.23 – A 64-dimensional space with 128000 is reduced by PCA to just two dimensions containing 4000 points.

Figure 2.23 exemplifies how the sample reduction and its geometric interpretation are carried out. Each point represents the projection of the original point along the direction with the highest variance. Note that the original sample is reduced, preserving its representativeness. In summary, the main techniques for Features Engineering are presented to the reader.

Methods for scaling the Features

1. Standardization;
2. Normalization;
3. Scale with Unitary Norm;
4. Scale based on the Maximum Absolute Value.

Missing data and imputation

1. Arbitrary Substitution;;
2. Sample removal;
3. Random imputation;
4. Interpolation and extrapolation;
5. Substitution statistics.

Outliers Treatment

1. Parameterization;
2. Non-parametric techniques, such as clustering;
3. Removal.

Transformações

1. Logarithmic;
2. Exponential;
3. Quadratic;
4. Box-Cox;

Feature Selection

1. Wrapper Method;
2. Embedded Method;
3. Relative Importance.

Dimensionality Reduction

1. PCA;
2. Random forests.

Other techniques;

1. Ordinal coding
2. Hot Coding;
3. Discretization;
4. Extraction of textual characteristics;
5. Extraction of characteristics from images.

2.4 BIBLIOGRAPHIC REVIEW

The World Health Organization (WHO) estimates that 35-40 million people need prostheses and that this tends to increase with the number of accidents. The WHO recommends that devices should take an integrated approach, including prosthesis fitting, user training, rehabilitation, community support, and repair services [Chadwell et al. 2020]. This section reviews the scientific literature on methodologies and technologies that have been used to develop upper and lower limb prostheses. For the development of such devices, it is necessary to understand how they are applied in everyday life and, for that, the research methodology has searched in various academic bases for the subject to raise the state-of-the-art.

Prostheses based on the myoelectric signal began to be investigated by literature in the 1940s, but due to the technology limitation at the epoch, progress only started in the 1960s [Iqbal, Subramaniam e Shaniba 2018]. In older versions, the operation was based on the amplitude of the electrical signal and, generally, the model considered only two operating states (on/off). Another problem is that the number of functions of the prosthesis did not meet the needs and users found it difficult to use the device, whose control was not entirely natural [Iqbal, Subramaniam e Shaniba 2018]. One of the most notable advances in the field was the emergence of control through pattern recognition (PR), where researchers assumed that each movement could be mapped according to signal characteristics [Iqbal, Subramaniam e Shaniba 2018].

In recent decades, there has been a significant improvement in efficiency classification using PR techniques, which made the approach the most promising. Despite the classification having achieved accuracy above 95%, its usability does not reflect the same result, as users tend to abandon the prosthesis due to the difficulty of use and wrong predictions due to the noise resulting from the system's variability. Currently, there is a divergence between the methods proposed by academia and those adopted by commercial systems. Although studies with more than one degree of freedom (DoF) have shown promising results, the lack of robust ones makes companies opt for simpler systems capable of meeting user needs in a limited but simpler way [Igual et al. 2019]. In this sense, the researchers realized that it would be necessary to create algorithms with some adaptive signal processing capable of mitigating the effects of muscle contraction, variations in the position of the electrodes, or user fatigue [Iqbal, Subramaniam e Shaniba 2018]. The literature considered several signal sources to estimate user intent. In addition, invasive and non-invasive methods have been evaluated by different studies [Igual et al. 2019], and the EMG signal acquisition methods on the skin surface have been predictable due to their simplicity and no need for surgery. EMG is used to obtain the features in the time or frequency domain that are the basis for the control algorithm.

2.4.1 Data Acquisition and Processing

Decoding the brain signal sent to the muscles is a complicated task. It is possible to access signals directly from the brain, using, for example, electroencephalography [McMullen et al. 2014]. However, the data acquisition process, as well as its necessary hardware, are not suitable for daily use due to the limitations of current technology [Igual et al. 2019]. Targeted Muscle Reinnervation (TMR) is considered a promising and relevant surgical technique to improve prosthetic control, especially in people with different amputation levels. The TMR technique makes it possible to obtain reinnervated areas that act as signal amplifiers and are less susceptible to noise [Mereu et al. 2021]. However, there are no accepted standards, and most methods adopt their assessment approach and methodology. The study identified the method's limitations by examining several articles on the subject and proposed a standard method for evaluating control performance metrics [Mereu et al. 2021].

The EMG acquisition method through field electrodes is the most commonly used to control the prosthesis and has been used since the earliest days on the subject. Due to their easy access and available information, superficial EMGs are the first control option, where non-invasive surface electrodes measure the electrical potentials generated in a muscle during its contraction. Several studies have investigated this approach [Igual et al. 2019]. Aiming to study the relationship between surface electromyography and to develop non-invasive robotic prostheses, the work made available a database of signals that simulate real operating conditions [Atzori et al. 2014]. The study [Kyranou, Vijayakumar e Erden 2018] investigates the causes of performance degradation in PR systems using non-invasive signal acquisition methods and argues that the main reason behind the instability of myoelectric pattern recognition control is that EMG signals are not stationary in the everyday environment of use.

Another factor related to signal information is the number of acquisition channels, which cannot be too limited or too high. In the first case, the capture of information would be harmful and, in the second, there would be many redundant sources. Therefore, before carrying out the acquisition, it is necessary to establish the number of channels to perform sampling [Igual et al. 2019]. Some works proposed to use high-density channels [Ison et al. 2016], but this approach is not strictly necessary to obtain high performance. *Young et al.* [Young et al. 2013] tested the effect of the number of channels, and the study suggests that having more than six channels did not reduce the estimation error, and performance did not improve. Methods are typically based on four to twelve channels using EMGs [Igual et al. 2019]. Within this range, models reach their peak performance with high efficiency during the data acquisition process.

After digitizing the EMG signal, it is necessary to process the signal before powering the system. In this sense, the next phase is segmentation, where the input signal is divided into windows for feature extraction purposes. The window moves continuously in time to accept new samples in a closed system. *Farrell et al.* [Farrell e Weir 2007] proposed a maximum windows of 300 ms to avoid delays in real-time operations, and *Nielsen et al.* [Nielsen et al. 2011] found that performance decreases with windows smaller than 100 ms.

2.4.2 Features and Prosthesis Development

For many years the feature engineering has been the object of study by researchers who aim to build intelligent prostheses. And it is not for less because if the variables have the correct information that describes the movement, then the network will be able to label them. The collected electrical signal can be tracked in two ways, as an image (atypical) or as a time series varying around the *(x, y)* axes, which allows many ways to extract the characteristics of the wave. The literature presented several *features* divided between the domain of time and frequency, and some of them ended up consolidating themselves over the years. This section will describe the characteristics that the research uses. The purpose of resource extraction is to highlight important information, rejecting irrelevant data and noise. The representation in the frequency and time domain can obtain the instantaneous energy of the signal [Patel 2016].

As mentioned earlier, feeding the system directly with myoelectric signals is not practical due to the randomness and non-stationary nature of the inputs. Feature extraction relies on condensing relevant information and is a process that is critical to the success of any PR-based model [Igual et al. 2019]. *Oskoei et al.* [Oskoei e Hu 2007] made an in-depth theoretical study of the different categories and features that can be extracted from EMGs. Time-domain features investigate the amplitude related to EMG signals, while frequency-domain characteristics focus on power spectrum parameters. Time-domain features are the most common in myoelectric controls because of their simplicity and since they are quickly calculated. *Phinyomark et al.* [Phinyomark, Khushaba e Scheme 2018] studied 26 different and individual features and eight sets of multiple features. In the following sections, the main features adopted by the literature will be presented [Toledo-Pérez et al. 2019].

Mean Absolute Value - MABS: the feature is used to detect muscle activity. MABS is the mean absolute value of the signal amplitude (time-domain). It is described by equation (36):

$$MABS = \frac{1}{N}\sum_{n-1}^{N}|X_n| \tag{36}$$

Standard Deviation - STD: measures the distance between the signal and its mean. It computes only the alternating current (AC) part of the wave. The feature is calculated by equation (37).

$$STD = \sqrt{\frac{1}{N-1}\sum_{i=0}^{N-1}(x_i-\mu)^2} \tag{37}$$

Variance - VAR: is the measure of the statistical dispersion of a variable and indicates how far the value is from the expected measure. It is the square of the standard deviation. Equation (38) represents this feature.

$$VAR = \frac{1}{N-1}\sum_{i=0}^{N-1}(x_i-\mu)^2 \tag{38}$$

Waveform Length - WL: this characteristic measures the length of a given waveform, providing information about the complexity of the signal. It is a robust feature in the signal classification process and is defined by equation (39) [Arjunan e Kumar 2010].

$$WL = \sum_{n=1}^{N-1} |X_{n+1} - X_n| \tag{39}$$

Effective Value or Root Mean Square - RMS: is related to the constant strength, and non-fatigued contraction [Arjunan e Kumar 2010] of the muscle. It defines the magnitude of the signal. Equation (40) represents this feature.

$$RMS = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x_n^2} \tag{40}$$

Zero Crossing - ZC: is a feature that counts the number of times the waveform crosses zero, changing the wave's signal, thus denoting the number of times the wave changes the value from positive to negative in a region. N is the frame number. The signal change is calculated according to equation (41).

$$Z(i) = 1/2N * \sum_{n=0}^{N-1} |sgn\left[x_i(n)\right] - sgn\left[x_i(n-1)\right]| \tag{41}$$

where:

$$sgn\left[x_i(n)\right] = \begin{cases} 1, x_i(n) \geqslant 0) \\ -1, x_i(n) < 0) \end{cases}$$

Signal Slope Change - SSC: calculates the number of changes between positive and negative slopes, considering three segments in sequence. The feature uses a threshold function to prevent noise. Equation (42) represents this feature.

$$SSC = \sum_{n=2}^{N-1} \left[f\left[(x_n - x_{n-1}) \times (x_n - x_{n+1})\right]\right] \tag{42}$$

$$f(x) = \begin{cases} 1, if\ x \geqslant threshold \\ 0, \qquad otherwise \end{cases}$$

Average Power - PWR: the signal strength measures the energy of the wave at a given moment t. Equation (43) shows this relationship.

$$PWR = \lim_{N \mapsto \infty} \left(1/(2N+1)) * \sum_{n=-N}^{N} \left|x(n)^2\right|\right) \tag{43}$$

Mean Absolute Difference - MAD: it is the average absolute difference between two successive values of the wave. Equation (44) has this characteristic.

$$\Delta X_i = \frac{1}{N-1}\sum_{k=1}^{N-1} |x_{k+1} - x_k| \tag{44}$$

Maximum Fractal Length - MFL: this feature is similar to the wavelength but is on a logarithmic scale and is less sensitive to background noise. It is used to indicate the density of the action potential in the muscle fiber [Arjunan e Kumar 2010]. Equation (45) describes its formulation.

$$MFL = log_{10}\left(\sqrt{\sum_{n=1}^{N-1} (x(n-1) - x(n))^2}\right) \tag{45}$$

Higuchi Fractal Dimension - HFD: Higuchi Fractal Dimension is applied to the EMG to identify active states and measure the complexity of the signal. The HFD can be calculated by linear regression [Garavito et al. 2016]. Equation (46) presents its mathematical formulation.

$$HFD = \frac{n\sum(x_k y_k) - \sum x_k \sum y_k}{n\sum x_k^2 - (\sum y_k)} \tag{46}$$

where:

$$\begin{cases} x_k = ln(1/k) \\ y_k = ln(L(k)) \end{cases}$$

Fractal Dimension - TFD: the equation (47) describes this feature, where a set N of square boxes is used to superimpose the sign, divided by the inverse logarithm of the L [Boccia et al. 2016] box area.

$$FD = \frac{logN}{log\frac{1}{l}} \tag{47}$$

Cardinality - CARD: is defined as the number of values that are unique in a set. For example, A = {2,4,5 } and B = {1,1,2,3,3 } have cardinality 3 [Ortiz-Catalan 2015].

Rough Entropy - REN: Rough Entropy is based on the idea that, for each object, there is a level of related information. The article [Zhong et al. 2011] discusses this theory applied to the development of characteristics. Equation (48) briefly introduces this concept.

$$REN = -\sum_{i=1}^{n} \frac{|R_i|}{|U|} log_2 \frac{1}{|R_i|} \tag{48}$$

where textit $\frac{|R_i|}{|U|}$ expresses the probability of the equivalence class Ri within the universe U and $\frac{1}{|R_i|}$ indicates the chance of one of the values in the textit Ri equivalence class.

2.4.3 Frequency Features

The last three characteristics are obtained in the frequency domain, representing the average (49), the median (50), and the peak frequency (highest frequency of the wave).Frequency Features.

$$FM = \frac{\sum_{i=0}^{n} I_i f_i}{\sum_{i=0}^{n} I_i} \tag{49}$$

where n is the number of frequencies in the spectrum, f is the frequency and I is the intensity. The median is the frequency that divides the EMG spectrum into two parts with the same amplitude. Where Pi is the EMG power spectrum at bin i.

$$FD = \frac{1}{2}\sum_{i=1}^{N} P_i \tag{50}$$

2.5 CLASSIFICATION ALGORITHMS

With the characteristics extracted and properly labeled, it is possible to use a classification algorithm. In this sense, the literature presents the most diverse methods, from statistical techniques to deep neural networks.

One of the simplest, easiest to implement, and most robust classifiers is the LDA statistical method [Campbell, Phinyomark e Scheme 2019]. However, the classifier suffers from the temporal evolution of the EMG signal, which requires aptitude that goes beyond the capacity of the method. In this sense, the study [Campbell, Phinyomark e Scheme 2019] implemented a modified version of the LDA to improve control and incorporate some missing features. The authors suggest remodeling the original cost function and adapting the algorithm to better deal with the temporal characteristics of the series. The methodology compares the results with the standard LDA classifier and suggests that active errors were significantly smaller, although the general error remained stable [Campbell, Phinyomark e Scheme 2019].

The Vector Machines Support (SVM) is one of the main pattern recognition methods in EMG. Its operation is based on identifying an n-dimensional hyperplane to separate a set of input resource points into different classes. Some authors argue that this technique is capable of recognizing complex patterns in various situations and, in the case of the EMG signal, they operate better than LDA and ANNs [Toledo-Pérez et al. 2019].

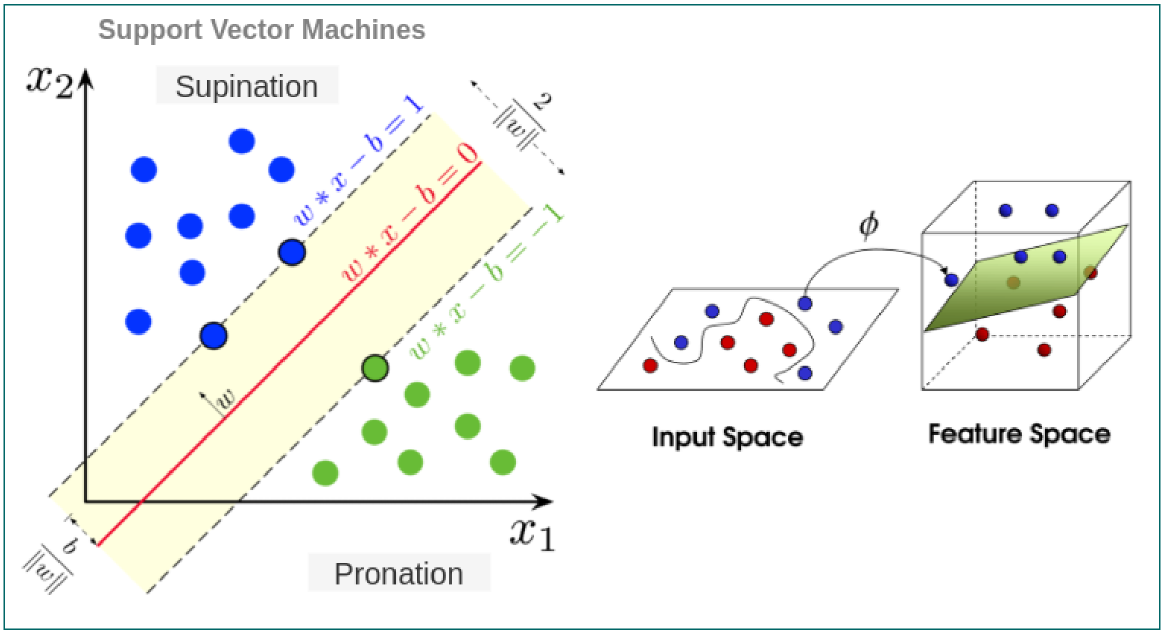

Figure 2.24 – Geometric design of the SVM operation. Illustration made from a figure obtained in *Creative Commons*.

The SVM theory was introduced by *Vapnik* and *Cortes* in 1995 [Cortes e Vapnik 1995] and is based on building a high-dimensional hyperplane capable of making the ideal separation between inputs [Toledo-Pérez et al. 2019]. The hyperplane must be able to unambiguously separate the classes so there must be an iterative method capable of finding the plane that best satisfies the separability between the classes. In Figure 2.24, the algorithm is exemplified. Figure 2.24 exemplifies the algorithm. Several authors recommend this method; This is mainly because the approach is flexible and can be used with other AI methods and techniques, which improves the quality of the classification. Many researchers have discussed PR-based classification methods for myoelectric control applications using the SVM [Toledo-Pérez et al. 2019].

In the investigation [Parsaei e Stashuk 2011], the authors used the network to assess whether the Motor Unit Potential or the Potential Train represents a single motor unit and obtained 95.6% accuracy. The research [She et al. 2010] used two different kernels (Gaussian and RBF) to classify five lower limb movements. The methodology sampled the data through 4 EMG channels, combining the characteristics MAV, WL, ZC and, SSC in such a way that, in total, 16 different entries are presented to the SVM. With this method, the authors achieved more than 90% accuracy. The SVM algorithm was applied to the NinaPro signal repository to classify 17 hand and wrist movements, the input vector was composed by the RMS obtained by Discrete wavelet transform, and the average energy of the spectrogram in each frequency range [Too et al. 2018]. Furthermore, the authors adopted the PCA algorithm to discriminate the three most relevant components. Accuracy reached 95% for normal individuals and 71.3%

for amputees [Too et al. 2018].

The MLP algorithm was also widely studied in the literature [Igual et al. 2019]. The work [Raheema, Hussein e Al-Khazzar 2020] used Perceptron Multi-Layer to classify five hand gestures according to 8 features extracted from the raw signal. The authors emphasize the need for pre-processing to improve accuracy. Hardware circuits were developed, and software was written to implement the intelligent myoelectric prosthesis.

To improve the accuracy in recognizing human movement patterns using an exoskeleton, the authors [Song et al. 2020] designed an EMG acquisition system on the body surface capable of identifying lower limb movement patterns. A supervised machine learning method is used to train an MLP and another LSTM classifier. The results of the experiment show that MLP reached 95.53%, against 96.57% for the LSTM network.

Recent literature has started to present classification solutions that use deep neural networks. Considering the case of the LSTM for estimating movements, *Quivira et al.* [Teban et al. 2018] built a regression model to predict the kinematics of movements and used the LSTM network. *Teban et al.* [He et al. 2018] claim that the LSTM algorithm has a higher performance than non-recurring networks in the case of myoelectric control and that its class separation mechanism can make a complex classification. Another algorithm explored in the case of myoelectric classification is the CNN network due to its ability to extract features.

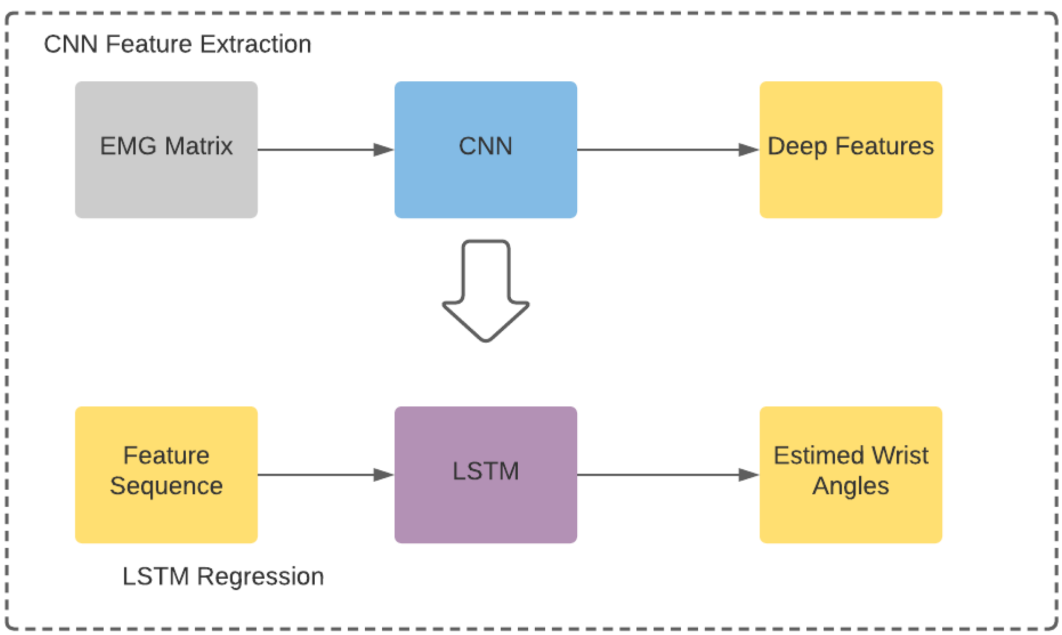

Figure 2.25 – CNN-LSTM hybrid approach applied to wrist movement.

However, muscular contractions have a strong temporal dependence, and CNN's were built to explore the spatial correlation. In this sense, the work [Bao et al. 2021] employed a CNN network with the LSTM algorithm to capture long-term dependence in a time series to classify complex pulse movements. The CNN network extracts features from the raw signal, and then the methodology organizes the input into a feature vector to make the regression and predict the kinematics desired by the user. The authors argue that the method has surpassed traditional machine learning techniques and a conventional CNN network. The Figure 2.25 illustrates the concept.

Motivated by the sequential nature of the electromyogram signal, the study [Hu et al. 2018] proposed a hybrid architecture based on CNN and RNN attention (CNN-RNN). The objective is to capture the temporal properties of the EMG in a more representative way, considering the gesture recognition problem. Similar to this work, the authors explore a new way to present the image to the classifier, where first, the input is arranged in a matrix form from a series of transformations of the original signals and, finally, performing the Fast Fourier Transform (FFT) [Hu et al. 2018]. The analysis methodology considered five signal databases, including BioPatRec and NinaPro, and achieved high accuracy. In the discussion, the article is compared with this work.

The investigation [Tam et al. 2021] uses high-density surface electromyography (HD-EMG) and a convolutional neural network (CNN) to individually classify specific patterns of voluntary muscle contraction. In addition, a transfer learning approach is used to reduce training time and allow for a more feasible calibration process. The gesture recognition system was evaluated in a group of 12 users without disabilities. A real-time test for six-movement classes resulted in predictive values averaging 93.43%. The fine-tuning of the network took less than 10 minutes to complete. According to the author represents a reduction of 89.4% in time, considering similar approaches that do not use the attention mechanism [Tam et al. 2021].

In final consideration, it is clear that the current literature has paid attention to DL methods and is looking for ways to mitigate the limitation of such networks in the case of EMG. The jobs aim to reduce training time, keeping accuracy high, and mapping complex features. However, unlike this Thesis, studies usually use the raw signal or perform some pre-processing before, instead of extracting the features that are already known in the literature. Furthermore, few studies consider the global training of a population, which makes the solution a specific and individualized product.

2.5.1 Research Differential

One of the limitations of this work is the robustness of the approach in a real use scenario since the system was not tested in an embedded system. It is worth mentioning the interest in future work of this type. First, it is necessary to establish criteria to define what robustness is in the scenario of myography focused on the construction of prostheses. During the research, the author found that the main metrics in the field are:

1. Accuracy: ability to distinguish between movement classes with up to 3-DoF. It is important to remember that accuracy decreases with everyday use and is sensitive to variability;

2. Recalibrated lightweight approach: a model can benefit from rapid training for continuous temporal adaptation;

3. Operate under complex conditions: the high variability of everyday use (humidity, temperature, electrode placement and motivation to perform the movement) reduce the usability of the prosthesis in practice. That is, we want to prove that the approach is capable of operating under a complex condition. With that she would be able to recognize complex patterns.

Many approaches have achieved high accuracy, but we are interested in expanding the robustness concept. The first differential of this work is to get a fast training using DL and GPUs, which makes it possible to test models capable of recalibrating the network from time to time. As there were no subsidies to test high variability scenarios, as in the case of real use, this research decided to verify the network's capacity to operate in an entire population. During the research, the author found that the researchers direct more efforts in individual training, that is, the authors seek to refine an individual product. But what if there are high-dimensional characteristics capable of moving generically in a population? Precisely in this gap lies the work differential, which proposes to train a network capable of operating in a population. This would help justify possible robustness in an embedded system. Another alternative to test the robustness of the system would be to introduce noise into the signal and check the quality of the results, which will be left to future work. Another alternative to test the robustness of the system would be to introduce noise into the signal and check the quality of the results, which will be one of the future works.

3 RESEARCH METHODOLOGY AND METHODS

> The important thing is not to be here or there, but to be. And being is a delicate science, made of small observations of everyday life, inside and outside people. If we do not carry out these observations, we do not become: we just are and disappear.
>
> Carlos Drummond de Andrade

This chapter details the methodological instruments and procedures used by the researcher to carry out the research. The type of research used in this Thesis was descriptive and explanatory. The descriptive methodology used books, articles, and academic works to raise the state-of-the-art on the subject and help in the construction of computational methods. In this sense, the hypotheses creation that guide this work was based on the study of the bibliography associated with the theme. Data collection procedures, such as myoelectric signals, were carried out through bibliographical and documental research, with a quantitative and qualitative approach, to relate the data for interpretation.

3.1 LIST OF ACADEMIC DATABASES

The first part of the study aims to raise current issues related to Artificial Intelligence aimed at the development of prostheses. The objective is to go deeper into the subject before building the computational models and methods necessary for the research. The author consulted the following scientific (but not limited) bases to increase knowledge:

- ACM Digital Library - Multidisciplinary (<https://dl.acm.org>);
- Crossref - Multidisciplinary - <https://www.crossref.org>;
- Europe PubMed Central - Biomedical - <https://europepmc.org>;
- Google Scholar - Multidisciplinary - <https://scholar.google.com>;
- IEEE Xplore - Computer Science, Engineering, Electronics - <https://ieeexplore.ieee.org/Xplore/home.jsp>;

- Index Copernicus - Multidisciplinary - <https://indexcopernicus.com/index.php/pl/>;
- Internet Archive Scholar - Multidisciplinary - <https://scholar.archive.org>;
- Elsevier - Multidisciplinary - <https://www.elsevier.com>;
- PubMed - Biomedical, life sciences - <https://pubmed.ncbi.nlm.nih.gov>; ResearchGate - Multidisciplinary - <https://www.researchgate.net>;
- SciELO - Multidisciplinary - <https://www.scielo.org>;
- Science Direct - Multidisciplinary - <https://www.sciencedirect.com>;
- Scopus - Multidisciplinary - <https://www.scopus.com>;
- Semantic Scholar - Multidisciplinary - <https://www.semanticscholar.org>;
- SpringerLink - Multidisciplinary - <https://link.springer.com>;
- Web of Science - Multidisciplinary - <https://www.webofknowledge.com>.

3.2 TECHNOLOGIES

The work employed three distinct technological fronts (software, hardware, and collaborative development). The programming environment should have dedicated software, such as:

- Python: Programming language with strong AI appeal and several libraries of the genre. Python allows for rapid and objective development;
- Scikit-learn: an open-source machine learning library (supervised and unsupervised learning). It also provides tools for model adjustment, data pre-processing, method selection, and evaluation.
- Keras is an open-source neural network library capable of running on top of TensorFlow and other libraries. It is easy to use, modular, extensible, and allows rapid development of Deep Learning applications;
- SciPy: Scientific computing tools for Python;
- Matlab: widely used in signal processing and engineering, the software is comprehensive and has a dedicated module to develop AI models.

Regarding the hardware, it is essential to emphasize that some models require computational power above the average and, therefore, it will be necessary a desktop with an Nvidia video card. This card allows you to parallelize the environment and execute training faster without using the processor. The methodology will prioritize pre-processing methods to decrease the load and make the training process more objective and cheaper. Finally, considering a collaborative production environment for the textual production of articles, the study adopted the Latex environment through the Overleaf website.

3.3 RESEARCH STEPS

This section will make a scope of the planning of this work. The research divided the planning into 9 phases:

1. Step 1 - Literature Review and Project Organization: The study selected the main articles and works on EMG, AI, and intelligent computational methods. The texts were then investigated to find out what the priority and potential areas are.

2. Step 2 - Survey of the Database: The methodology evaluated the most relevant databases (BioPatRec and NinaPro) regarding the feasibility of inclusion in the research project. Characteristics such as data representativeness, format, ease of access, and relevance to the topic were part of the selection criteria.

3. Step 3 - Pre-processing and data processing: Data processing can be the main task of AI. At this stage, the methodology treated the input before extracting the characteristics. The work employed some of the techniques discussed in Chapter Two to explore the potential of the data;

4. Step 4 - Features Development: The study studied the possibility of developing new features to improve the others and, for that, it used statistical techniques and deep neural models such as CNN. It is desirable to relate the characteristics of the signal to gain new insights;

5. Step 5 - Deep Neural Modeling: The methodology analyzed quantitatively and qualitatively the context of the problem (classification, regression, grouping, or a combination) to define the AI methods that should be used and to create the network architecture capable of achieving the performance metrics. Finally, the methodology adopted a parameter and hyperparameter adjustment protocol based on grid search and heuristics;

6. Step 6 - Testing and Validation: In this step, the author set up a test protocol where the models were exhaustively tested by the methodology. The goal was to find the

configuration capable of delivering the best result. The study also re-evaluated the literature to discover new opportunities and improvements;

7. Step 7 - Results and analysis: The research grouped the results into tables and graphs for easy visualization and analysis. The author provided the reader with a statistical evaluation of the outcomes and compared it with other studies. The reader must interpret the results in an interdisciplinary way, ensuring a broad judgment on the subject. At this stage, there were already subsidies for publication;

8. Step 8 - Writing and Submission of Academic Articles: With the help of the research group, the author prepared articles derived from the Thesis. The choice of journals was made according to the criteria of the institution and the Doctoral program;

9. Step 9 - Continuous Planning: The methodology adopted cyclical planning, and the research project was re-evaluated every three months to adjust to the demands.

3.4 INVESTIGATION OVERVIEW

One of the disadvantages of deep learning techniques is that they incur exorbitant computational costs, and this is a limiting factor for many products, which due to restrictions cannot benefit from this scientific increase. The study aims to establish a methodology that can create a bridge between the two worlds, that is, to allow embedded systems to use the concept elegantly, quickly, accurately, and without worrying about the hardware. In addition, knowing the increasing parallelization of devices such as FPGAs and GPUs, which increasingly tend to accommodate AI requirements, it is necessary to create an approach that allows the efficient use of resources, especially in systems with limited power supply.

It is important to mention that, in the particular case of myography, due to the deviations in the signal throughout the day and because of brain plasticity, two situations can occur. The first is the need to provide the network with the data in a homogeneous way, in which case the *features* engineering shows its relevance and need. Even though the DL technique can do it a priori, the algorithm would be exposed to unwanted situations, and this can take the model to an unknown state in the course of its operation. Concerning plasticity, many studies have concluded that the continuous use and effort employed by the individual play a crucial role in the usability of the prosthesis. This fact suggests that the recalibration of the network will be necessary, so, instead of performing it in a laboratory and controlled environment, it is possible to develop a model that is not costly and allows the user to use it whenever he starts to notice deviations.

The study methodology will provide the necessary basis for the experiment to be replicated faithfully by anyone who wishes. The work will implement two architectures based on *Deep*

Learning together with the BioPatRec prosthesis development software in a hybrid model, where instead of providing the signal or image directly to the classifier, which is common in the literature, four algorithms that will allow signal processing, feature extraction and the application of the features engineering techniques described in the previous chapter. The approach provides, regardless of the configuration of the input, the possibility of testing different arrangements, which is something fundamental for AI as seen in Chapter 2; the choice of the functional model is not obvious, being often the result of heuristics and brute force.

To carry out the comparative study with other known techniques, the research established different test configurations. The objective is, at first, to prove the capacity of both procedures (LSTM and CNN) when fed with pre-extracted characteristics at the expense of the raw signal, thus surpassing the traditional classifiers. In a second step, the study seeks to analyze the spectrum of operation of the network and provide the reader with its nominal model. For that, a protocol in grid and heuristics was employed in the assembly of the experiment. The investigation evaluated the techniques separately, as the algorithms were implemented and tested incrementally.

The second stage of the research rewrote part of the BioPatRec code in Python, thus creating its approach. The development in a high-level language ensured the parallelization of the networks using an Nvidia GPU this was done to test the hypothesis of optimized functioning in such devices. In addition, the entire filtering system of the previous software was replaced by a single Kaufman filter, and the data processing step also had some particularities changed.

3.5 BIOPATREC

The thesis chose the BioPatRec prosthesis study and development platform to implement the CNN and LSTM network in a hybrid way as there is excellent associated documentation. Developed by *Catalan et al.* [Catalan, Brånemark e Håkansson 2013] to be an open and modular project, the software allowed some stages of the study to be accelerated, as both the signal processing and the extraction and selection of characteristics were made dynamically by BioPatRec. In addition, the manufacturer's website already has some sets of signs that represent the movements of the limbs (upper and lower) in different configurations, so it was not necessary to acquire the data in laboratories. Figure 3.1 shows the software working in conjunction with the implemented LSTM network.

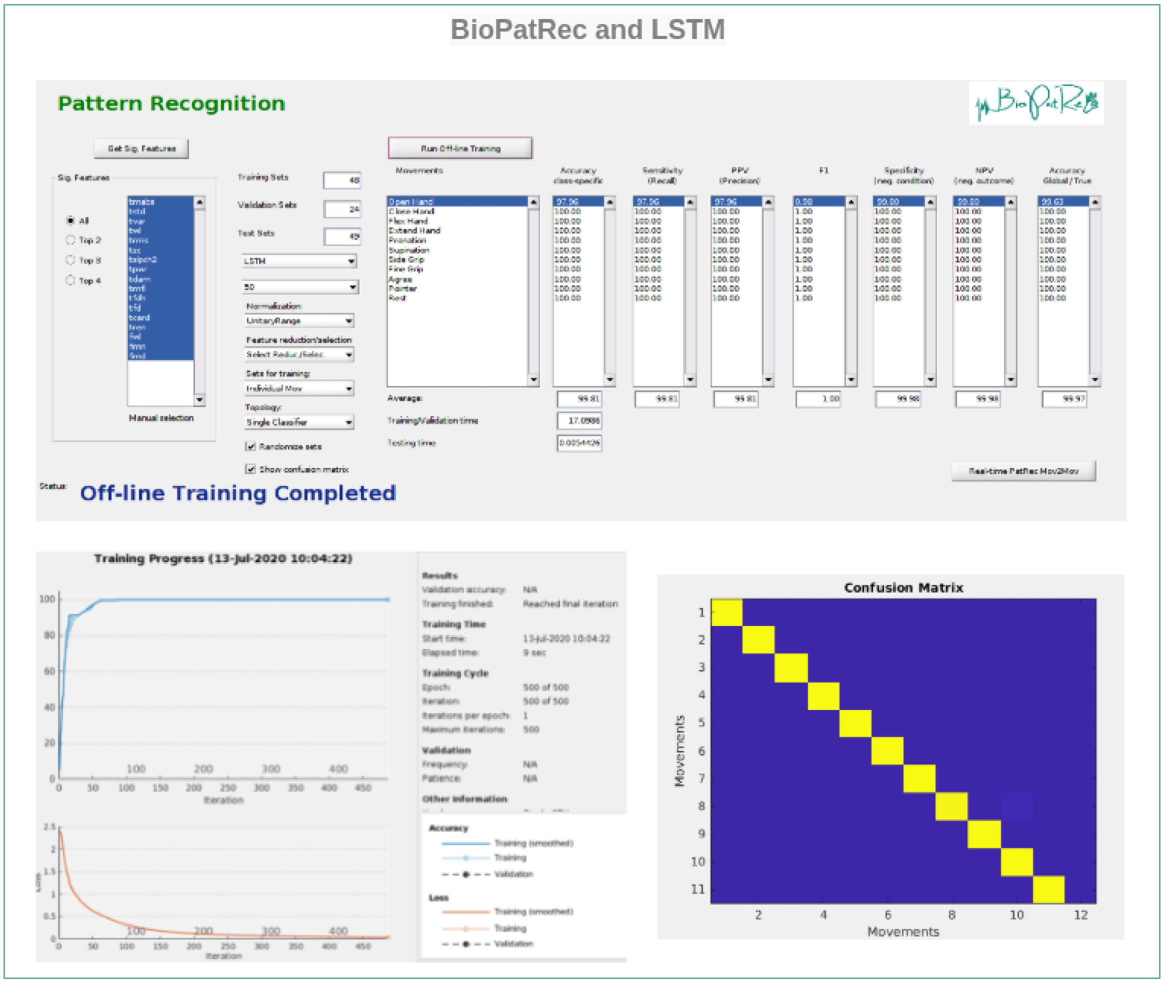

Figure 3.1 – BioPatRec and LSTM network carrying out the training process.

The software is divided into six modules, and the designer can make any intervention he deems essential, namely:

1. **SigRecordings:** This component is used to record the signals in the presence of an individual and is pre-configured for the use of some devices;

2. **SigTreatment:** The module allows the pre-treatment of the signal in a sophisticated way, where it is possible to remove some channel, remove a percentage of the transient period of the contraction, resample the wave, apply some filtering methods, do advanced processing of the characteristics. For example, use the PCA or the independent component analysis (ICA) and finally adjust the windows and overlays necessary for the segmentation of the wave. The research used these resources of the platform, mainly when choosing the assembly of the search grid;

3. **SigFeatures:** This part of the program allocates the collections of characteristics and allows the programmer to add or create new ones so that they can be used in the

classification process. In its original format, 27 features are available. However, the work employed only 18 (selection of characteristics). The methodology was based on heuristics, the methods discussed in chapter 2, and the results of other similar works. The approach had an impact only over time, so the selection had an immediate effect;

4. **PatRec:** This module shelters the classification algorithms, and it has made the necessary interventions for the DL networks to be added to their list of classifiers. Every procedure necessary to train and test the network must be made in this part of the code;

5. **Control:** It has the necessary routines for real-time control, the study did not make any intervention in this module;

6. **DataAnalysis:** It is a statistical tool that allows the developer to evaluate, visualize and interpret the results. It is interesting and practical to use this module because it provides the researcher with a solid means of analyzing the search in a grid;

3.5.1 Data Collection

During the bibliographical research, the author found two signal repositories, BioPatRec and NinaPro [Atzori e Muller 2015]. The methodology preferred the former because the associated software had many features that made it easy to test the approach. In addition, BioPatRec is modular and can be freely changed. Figure 3.1 presents the program's block diagram and exemplifies how the modules communicate.

Researchers collect signals from people with and without amputations. Then, the samples are treated through filters capable of removing unwanted artifacts from the EMG, such as noise. As mentioned before, the wave must be segmented into windows (values between 100 ms and 300 ms). This step allows the continuous functioning of the prosthesis. Each channel generates an activation signal for a specific region, and this data must be arranged in a matrix form. The Figure 3.2 illustrates the concepts mentioned.

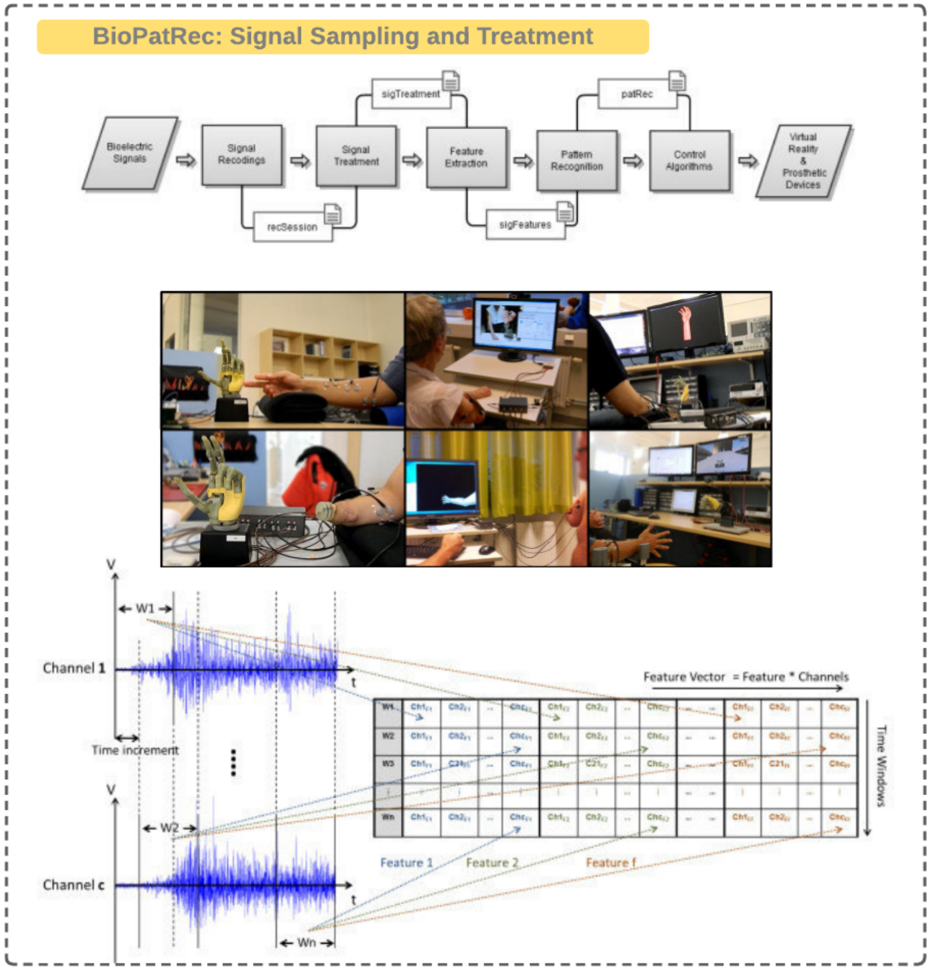

Figure 3.2 – Steps in the process of capturing, sampling and processing the EMG signal carried out by BioPatRec researchers [Catalan, Brånemark e Håkansson 2013].

The study employed six sets of data for methodology development, considering simple movements (a degree of freedom in space - 1 DoF) and compound movements with up to 3 degrees of freedom (combining several possibilities), which are related to the upper and lower limbs. The description of the protocol used is as follows:

- **10mov4chUntargetedForearm:** The first set was acquired from seventeen intact people, who performed ten individual movements of the upper limbs without composition. Each action was repeated three times, with an interval of three seconds between each test. The data were acquired using four uniformly spaced silver chloride electrodes and were digitized with a frequency of 2Khz and 16 bits of resolution. The acquisition device was a MyoAmpF2F4. The equipment applied a fourth-order high-pass filter with 20 Hz, a second-order low-pass filter of 400Hz, and a 50 Hz notch filter.

- **10mov4chUF-AFEs:** Eight individuals who did not suffer any trauma performed ten movements with only 1-DoF, all of them related to the upper limbs. However, in this case, the device used to collect the data was an ADS1299 (analog front-end). The rest of the configuration was identical, except for the fact that the instrument did not perform any signal filtering;
- **6mov8chUFS:** This collection represents 27 classes of movements with up to 3 degrees of freedom and is related to 6 individuals who do not have any injury type. The arrangement was the same as the first set of signs;
- **8mov16chLowerLimb:** The last three sets characterize only lower limb movements and had the collaboration of 8 participants. The collection of signals admitted only 1 degree of freedom. In this case, the experiment used a 4-second interval between each contraction, each movement was repeated three times at a rate of 2Khz. The difference between each set is related to the configuration of the electrodes used. In the first set, a monopolar configuration with eight channels was used (TMC - Targeted Monopolar Configuration), the second approach used a bipolar arrangement with eight channels (TBC - Targeted Bipolar Configuration), and the latter adopted a non-targeted monopolar configuration with 16 acquisition channels (UMC - Untargeted Monopolar Configuration).

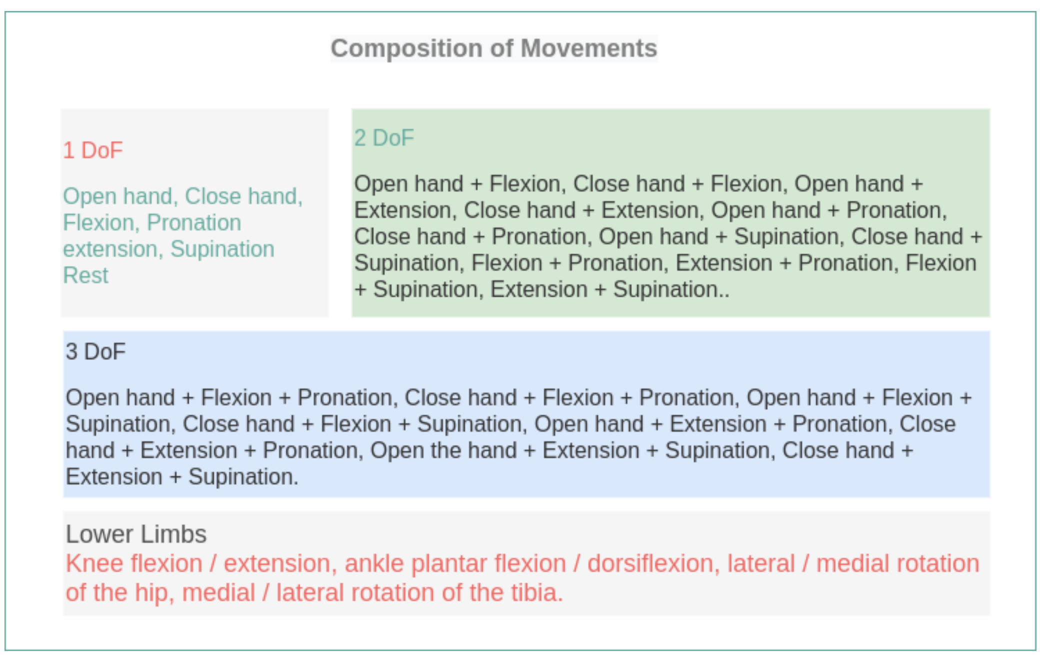

Figure 3.3 – Composition of movements and their respective degrees of freedom.

Figure 3.3 exemplifies the movement classes of the sets.

For the pattern recognition process to occur in a closed system, the characteristics must be extracted in a segmented way. The study adopted segments of 200 milliseconds with an overlap of 50 milliseconds between each window. To ensure that the system does not address the problem of overtraining the methodology divided samples as follows.

- 40 % of the data were exclusive to the training process;
- 20 % of the samples were left for the networks to validate the training process;
- 40 % was the test proportion these data are employed to obtain the statistics.

3.5.2 Classifiers

As comparison criteria, the study used the following classifiers: LDA, GLM, SOM, RFN, and MLP.

3.6 LONG SHORT-TERM MEMORY IMPLEMENTATION

The work implemented the LSTM network in the Matlab language, as BioPatRec was written with this tool. For this step, it was necessary to set up a configuration where regardless of the number of features, the network input would always accommodate the input. The study considered a network with only one layer since additional layers did not show significant advantages and degraded the time. However, it is possible to do it incrementally if the designer so wishes. The study integrated the option of the number of neurons graphically to BioPatRec, and it is enough to select the value directly in the corresponding tab of the program. The number of epochs, the size of the batch, and other hyper-parameters can be set in the code. The main algorithm is detailed in Appendix A1, where it is necessary to do some manipulations and matrix transpositions so that the dimensions of the features and the network input always agree. The Matlab function receives as input the training, validation, and testing set. The number of neurons in the hidden layer that the user wants also comes through the same function. The algorithm adjusts the input according to the number of channels and features. The user selects and marks the output according to the number of motion classes. The choice of configuration and set of signals can be done in the graphical interface of BioPatRec, and the matrix adjustments are automated by code. The Figure illustrates the process block diagram for the LSTM network.

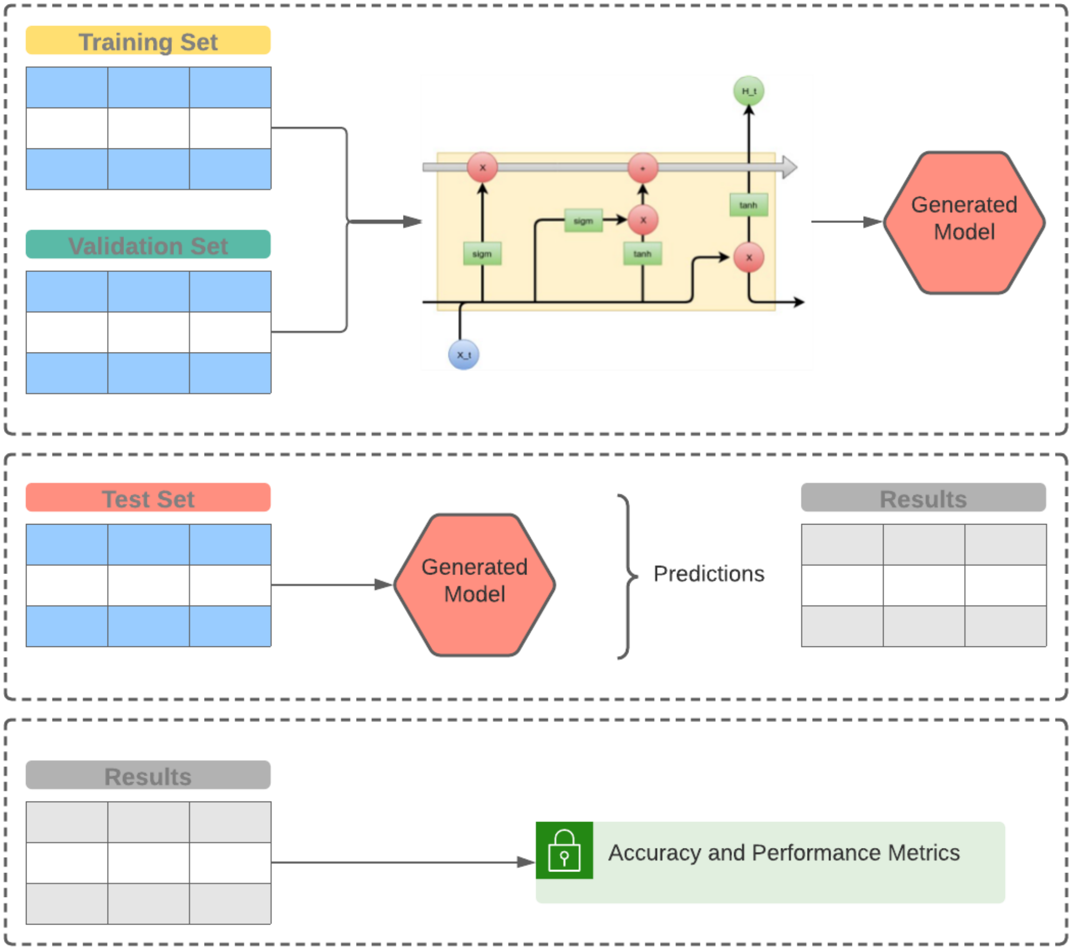

Figure 3.4 – Block diagram of the implemented LSTM model.

Then, the work implemented the test function, which follows the same principles of the previous one in matrix terms, and made all the adjustments in BioPatRec so that the classification and statistical modules could make the necessary references to the Deep Learning technique added by the study. With this integration, it was possible to obtain all the treated and normalized features. After the training process, the investigation takes advantage of the statistical facility of the tool to compute the results.

3.6.1 Adjustment Protocol

Regarding this algorithm, the investigation was carried out in two stages which have their particularities and objectives. The first was made considering the set of individual movements *10mov4chUntargetedForearm*, and the second is related to the collection of simultaneous collection. The first test evaluated two different aspects:

1. **Set of characteristics and population variation:** The survey tested the network on three groups of different characteristics. The first case used all the features described in Chapter Two, then the features that *Hudgins et al.* (MABS, WL, ZC, SSC) [Hudgins, Parker e Scott 1993] developed were tested. Finally, the methodology added cardinality to the Hudgins group. With this approach, it is possible to know the influence of each group on the convergence process during training. The study tested the capacity of the LSTM network on each individual in the sample separately. This procedure was made because the capability of some algorithms is limited in some individuals since each one of them has intrinsic characteristics related to the nature of the signal.

2. **Grid search and Hyper-parameter adjustment:** The work adopted a search protocol combining brute force and heuristic to obtain the functional model. The number of neurons varied between 10, 50, 100, and 200, the allowed times were 100, 500, and 1000, the optimization algorithm used was RMSprop. The other two parameters that the investigation adjusted were the learning rate and the Squared Gradient Decay Factor (decay factor used when adjusting the weights). During this assessment, the methodology employed a fixed network with 100 neurons and 100 times, the tested arrangements were made only on people 3,5 and 13, as they presented the worst individual results, so any increase in their outcomes would impact the overall result of the population (intuitive premise). After choosing the reference values, the experiment was repeated in the other individuals.

For the set of combined movements, the thesis adopted 17 features as the network's input. The neurons of the hidden layer chosen were 50 or 100, the number of allowed iterations was 50, 100, or 200. Then, the work employed the search in a grid using the optimized values of the learning rate and the Squared Gradient Decay Factor that were found in the previous experiment. Figure 3.5 summarizes the protocol with the LSTM network.

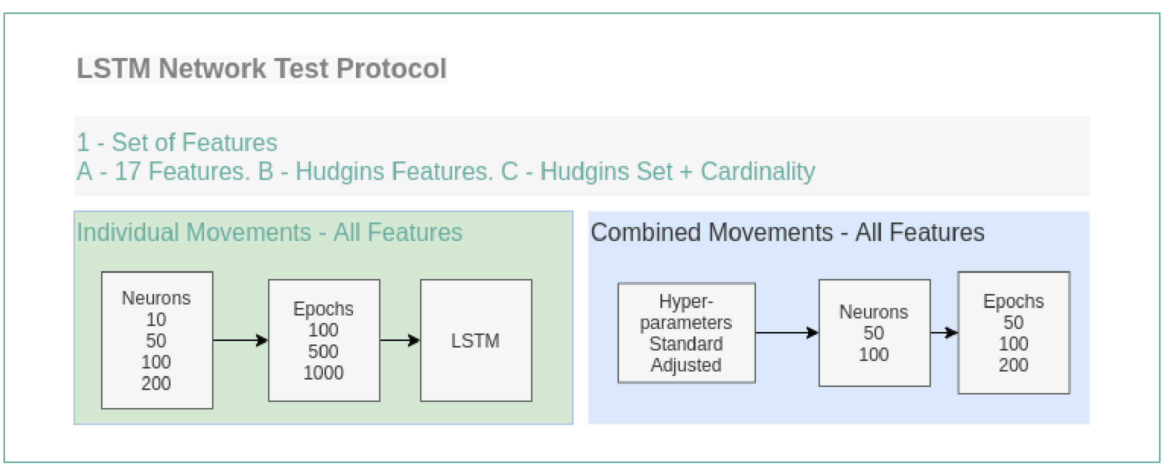

Figure 3.5 – Tested configurations of the LSTM network using a grid protocol.

The training model applied to all experiments in the LSTM network was of the batch type, as the method allowed the best relationship between accuracy and training time. The mini-batch approach could also have been used since the results were satisfactory (during the initial tests). However, the study opted for batch training as this choice improved time.

3.6.2 Statistical Analysis

In each search arrangement in a grid, the survey carried out ten network training sessions per individual. Each of the results is employed to calculate and average a single population test. So the experiment applies the ten averages to arrive at the overall average of the experiment. The work considered accuracy and training time as a metric (the prediction time was negligible and without any statistical impediment), with the computed values being the mean and the standard deviation. To verify if there was any statistical difference after changing some hyperparameter and to compare different classifiers, the methodology applied the paired t-test, which allows validation of the hypothesis considered. The statistical difference adopted was equal to $p < 0.05$.

3.7 CONVOLUTIONAL NEURAL NETWORK

Unlike the LSTM network, where the study adopted an architecture with only one layer, the CNN network was tested in two different configurations, with one (CNN-A) and two convolutional layers (CNN-B), respectively. The methodology submitted the networks to the same hyperparameter adjustment protocol. Unlike traditional networks that receive sample data sequentially, 17 resources are arranged in a row and input is provided to the network. The CNN network treats the input as an image and performs kernel convolution by making a slide. The technique always organizes resources in the same way. The order in which they are arranged always obeys the same geometry, and the input matrix will present the same configuration in all tests. With this fixed arrangement, the final result of each configuration will not be harmed.

Thus, the methodology created an algorithm capable of making the necessary adjustments to the matrix and tested some possibilities. For example, consider the case where 17 characteristics were sampled on 16 channels. One way to organize the input would be through a matrix of dimensions 68x4x1 or 34x8x1. The settings vary from one experiment to the next. However, there is no need to intervene directly in the code unless something more exotic is necessary. Thus, the general pattern is as follows:

$$Entrace = \begin{bmatrix} \frac{F*C}{n} & n & 1 \end{bmatrix} \tag{51}$$

Where F is the number of features (when not using PCA), C is the number of channels, and n is the second dimension, which must be a multiple of the previous relationship, as the matrices have integer values for the rows and columns. The following parameter to be dimensioned is the size of the filter (Kernel). It defines how much of the neighborhood each neuron in the layer can see. The filter is defined by two integer positives, and squares are generally adopted as a geometric pattern. The study adopted a 4x4 kernel for the tests. However, it is common to find values such as 3x3 or 5x5 when we have more inputs. Then it is necessary to define the number of filters, which represent the number of neurons in the convolutional layer that connect to the same input region. The parameter determines the number of channels at the layer's output. For this value, the investigation normalized according to the input, dividing the value by 4. So if the entrance has 136 samples (17x8) the number of filters will be 34. The next parameter is called stride and, it defines how much the filter will move in each iteration. For this, the work used the value 1, which is the most usual. The principal function that describes the implemented CNN network is presented below.

Each of the six data sets was evaluated by the methodology in two different architectures. The first had only one convolutional layer, and the second two. The research carried out the tests after the grid search process. After hundreds of iterations, the study adopted 0.003 for the learning rate, the maximum number of times was 20 (the hybrid CNN converged much faster than the LSTM network), and the decay factor was 0.9. Regarding the training mode, the mini-batch type was used, because empirically, it was found that high values decreased accuracy and low values degraded time. The reference value used by the thesis was 128 samples presented to the network at each iteration. Tables 3.1, 3.2, 3.3, 3.4 present the configuration of 4 networks adopted by the methodology.

Table 3.1 – Architecture A - Simultaneous Movements

	CNN-A One Convolutional Layer
Input	34x4x1 image
Convolution	34 4x4x1 Convolution with stride [1 1]
Normalization	Batch normalization with 34 channels
ReLU	Nonlinear Threshold Operation
Max Pooling	Pooling stride [1 1]
Fully Connected	Output with 27 classes
Softmax	Activation Function
Classification	Cross-entropy 26 classes

For the statistical analysis, the LSTM network protocol was employed by the approach. The survey used the same protocol concerning the algorithms used in the comparison. The

Table 3.2 – Architecture B - Simultaneous Movements

	CNN-B Two Convolutional Layers
Input	34x4x1 image
Convolution	34 4x4x1 Convolution with stride[1 1]
Normalization	Batch normalization with 34 channels
ReLU	Nonlinear Threshold Operation
Max Pooling	Pooling stride [1 1]
Convolution	34 4x4x1 Convolution with stride [1 1]
Normalization	Batch normalization with 34 channels
ReLU	Nonlinear Threshold Operation
Max Pooling	Pooling stride [1 1]
Fully Connected	Output with 27 classes
Softmax	Activation Function
Classification	Cross-entropy 26 classes

research adopted the same values of training, validation, and testing for all classifiers. The chosen normalization was the unitary norm. Some classifiers used the reduction of dimensionality through the PCA (some cases) because, during the search in a grid, it was found that the MLP, SOM, and LDA networks reached better values for accuracy while the training time reduced, this was observed in some data collections, and for that reason, the research adopted this mode of preprocessing selectively. Table 3.5 exemplifies the result of using the PCA in the set of characteristics.

3.7.1 Hardware and Software

The research carried out tests on a computer with an i7 8-core processor and 16 Threads, with 16 GB of RAM and a 500 GB hard drive. The operating system was Ubuntu running Matlab and BioPatRec, during that stage of the test.

Table 3.3 – Architecture A - Movements of the Lower Limbs

	CNN-A One Convolutional Layer
Input	68x4x1 image
Convolution	68 4x4x1 Convolution with stride [1 1]
Normalization	Batch normalization with 34 channels
ReLU	Nonlinear Threshold Operation
Max Pooling	Pooling stride [1 1]
Fully Connected	Output with 9 classes
Softmax	Activation Function
Classification	Cross-entropy 8 classes

Table 3.4 – Architecture B - Movements of the Lower Limbs

	CNN-A Two Convolutional Layer
Input	68x4x1 image
Convolution	68 4x4x1 Convolution with stride [1 1]
Normalization	Batch normalization with 34 channels
ReLU	Nonlinear Threshold Operation
Max Pooling	Pooling stride [1 1]
Convolution	68 4x4x1 Convolution with stride [1 1]
Normalization	Batch normalization with 34 channels
ReLU	Nonlinear Threshold Operation
Max Pooling	Pooling stride [1 1]
Fully Connected	Output with 9 classes
Softmax	Activation Function
Classification	Cross-entropy 8 classes

3.8 BIOPATREC-PY AND PARALLELIZATION

The author started the development of a GPL-licensed tool (General Public License) based on BioPatRec. However, written in Python (object-oriented) instead of Matlab. Python is a language that houses several AI libraries and has a strong development community. Figure 3.6 shows the project's UML Packages diagram. The idea is to allow, in a first moment, the mutual use of AI, GPUs, and FPGAs, in a second stage (future work of the thesis), to use the FPGAs to make the logical synthesis in the following sense:

1. Write routines in a high-level language, using Keras, TensorFlow, and other libraries;
2. Implement a parallelized environment, used primarily by GPUs and verify current performance requirements;
3. Use Intel FPGAs from Xilinx, which ship native AI units, to perform the HDL synthesis.

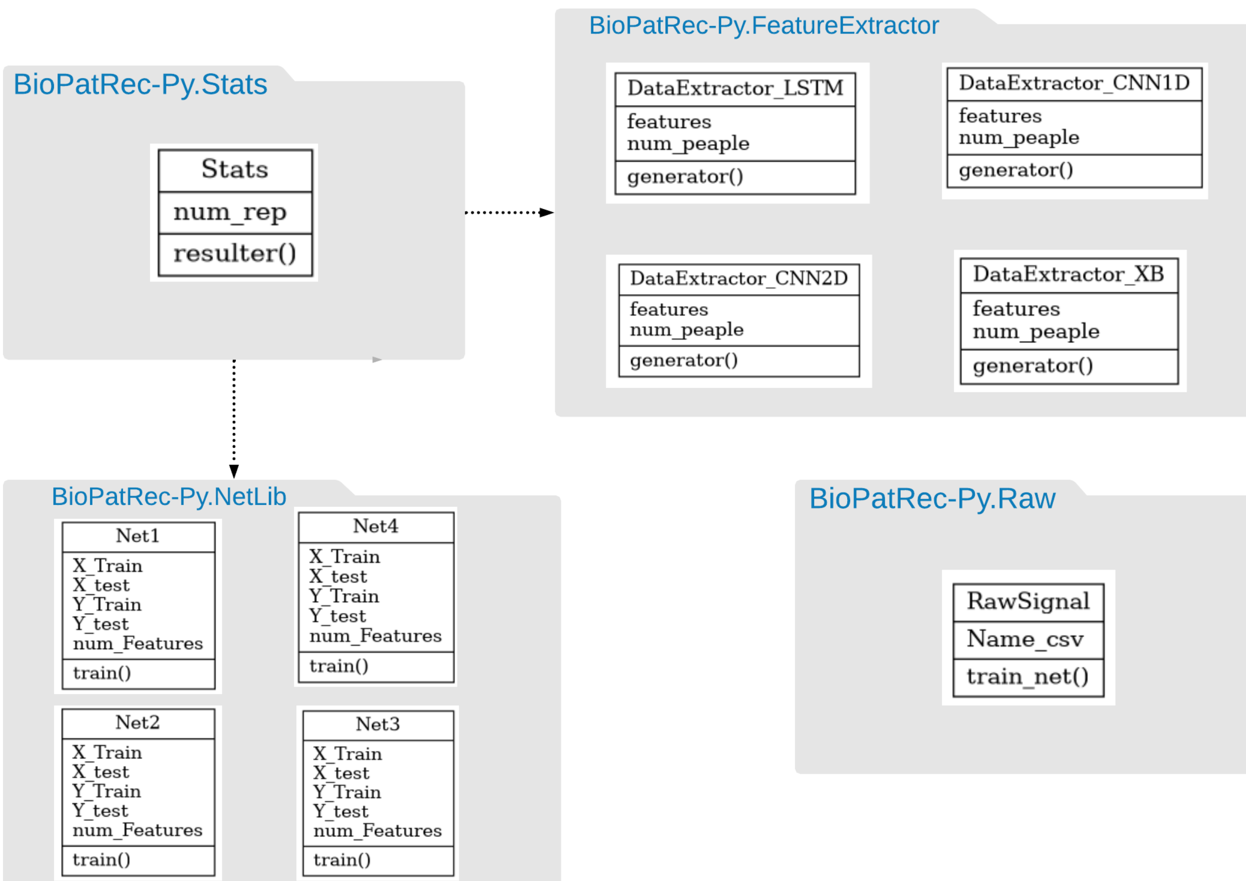

Figure 3.6 – BioPatRec-Py UML Package Diagram

In addition, with BioPatRec-Py, it will be possible to verify in the case of the myoelectric

Table 3.5 – Dimensionality Reduction

Effect of Dimensionalty Reduction - PCA		
With PCA		
1	Sequence Input	272 dimensions
2	LSTM	200 hidden units
3	Fully Connected	9 fully cnnected layer
4	Softmax	softmax
5	Classification Output	crossentropyex
Without PCA		
1	Sequence Input	160 dimensions
2	LSTM	200 hidden units
3	Fully Connected	9 fully cnnected layer
4	Softmax	softmax
5	Classification Output	crossentropyex

signal if it is possible to improve the performance metrics that the solution demands, considering the DL techniques. The program is not an exact copy of BioPatRec. In contrast, it has its characteristics. For example, the entire filtering scheme was replaced by a single Kaufman filter. The choice was due to heuristics. The old filtering schemes did not have clear benefits, and their combination was complicated, which increased the number of parameters to be tested.

The system introduced did not use the same statistical tools as the previous one. Instead, it used the Keras engine. This choice happened for two reasons. It was not clear to the author how the accuracy calculation was implemented in BioPatRec. When the research started testing the networks considering the BioPatRec metrics (Precision and Recall), the results were always close to 100% in all cases (using Keras Precision and Recall). The study chose to use the most conservative metric only, so Recall and Precision were not considered.

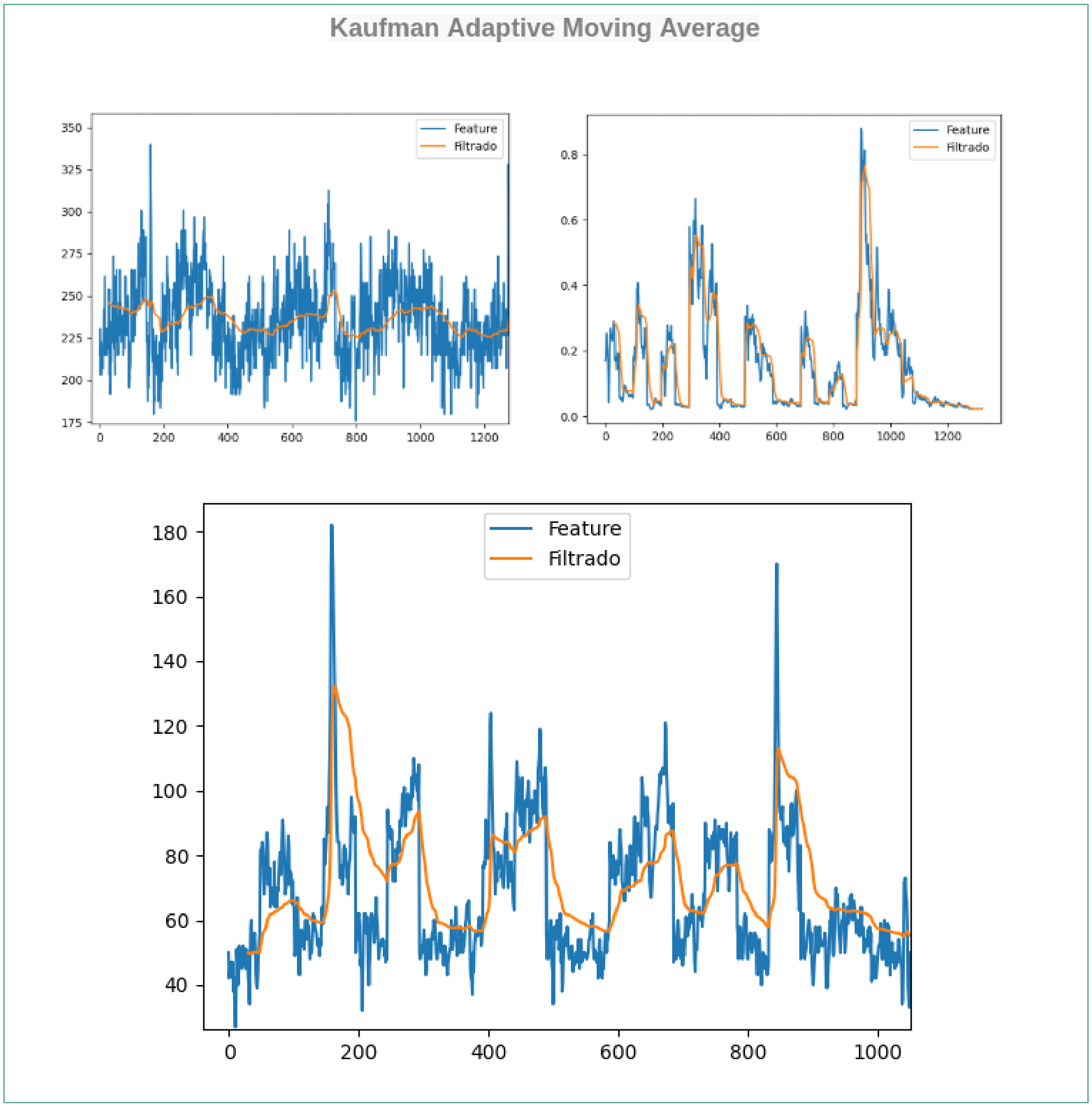

Figure 3.7 – Unique BioPatRec-Py filtration system

The study replaced all old filters (Low pass, high pass and notch) with the Kaufman Adaptive Moving Average (KAMA) - developed by Perry Kaufman. It follows the signal when the noise is low and softens the noise when the signal varies. The filter is based on the Exponential Moving Average (EMA) and responds to both trend and variation. Figure 3.7 exemplifies the use of this resource considering the characteristics of the myoelectric signal and not the original wave captured on the skin surface.

Given a sequence of m elements and a sub-sequence of n elements $P = (p_1,\ dots, p_m)$, KAMA is defined as in (52):

$$\alpha = \left(\frac{|p_{i+1} - p_{i+n}|}{\sum_{j=1}^{n-1} |p_{i+1} - p_{i+j+1}|} \left(\frac{2}{e+1} - \frac{2}{a+1} \right) + \frac{2}{a+1} \right)^2 \tag{52}$$

In this way, all characteristics are tracked as a time series and then filtered. This approach can be considered an Engineering of Features since it focuses on the features and not on the signal. Regardless of the configuration of the features that feed the network, all of them will be filtered. The method improved accuracy and reduced training time.

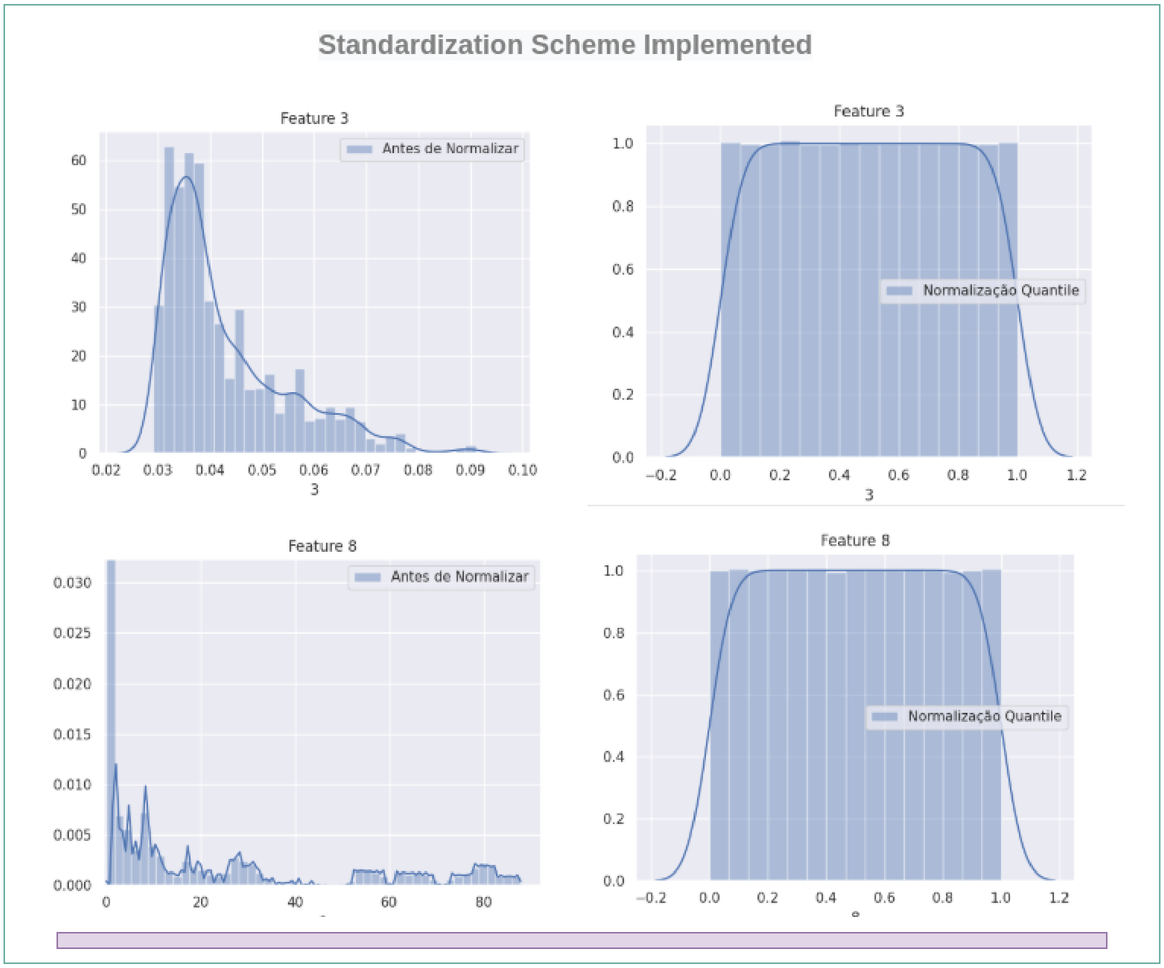

Figure 3.8 – Standardization technique adopted to achieve uniform distribution.

The new system has also implemented its standardization method. During plots, it was noticed that even after normalizing the features using standard methods, the distribution didn't faithfully approach the normal, and in some cases, it did this completely. In this case, the research adopted a normalization system implemented in Python and directed to AI, called Quantile Normalization. The method allows the characteristics to follow a uniform distribution and reduces the impact of marginal values (outliers). The pre-processing used is robust and has immediate effects.

For its use, it is enough to define the number of quantiles. Quantiles are reference points that discretize the distribution function. Figure 3.8 shows the before and after of the standardization adopted. Normalization speeds up the network's convergence rate and allows its objective function to function correctly.

3.8.1 Environment Setting

Before implementing the network, the entire environment had to be parallelized. The study acquired the NVIDIA GeForce GT 1030 entry card, with 384 CUDA cores (some even have 7,936 cores). CUDA is a parallel computing technology that allows you to use the video card engine precisely and freely. It provides the programmer with a means of switching between serial programming from a conventional CPU to the parallel of a GPU. The methodology carefully adjusted the environment and employed the following configuration:

- CUDA ToolKit 10.1 Software;
- CuDNN 7.6 software;
- TensorFlow GPU 2.5;
- Python 3.8 development environment;
- GCC 7.3 compiler;
- NVIDIA GeForce GT 1030 2GB - DDR4;

The first two software allow the operating system to switch the type of programming and provide the necessary means of abstraction for the card. The third is the high-level library that implements the AI networks, that were written in Python and parallelized.

3.8.2 Networks

The study considered only three networks at this stage, and the methodology implemented the algorithms through Keras. The networks adopted were: LSTM, one-dimensional CNN (currently is widely adopted), and a two-dimensional CNN (BioPatRec). The first network implemented was the LSTM, and the code in the Appendex A2 exemplifies one of its configurations. The Figure 3.9 exemplifies the steps used to carry out the training and evaluation of a network.

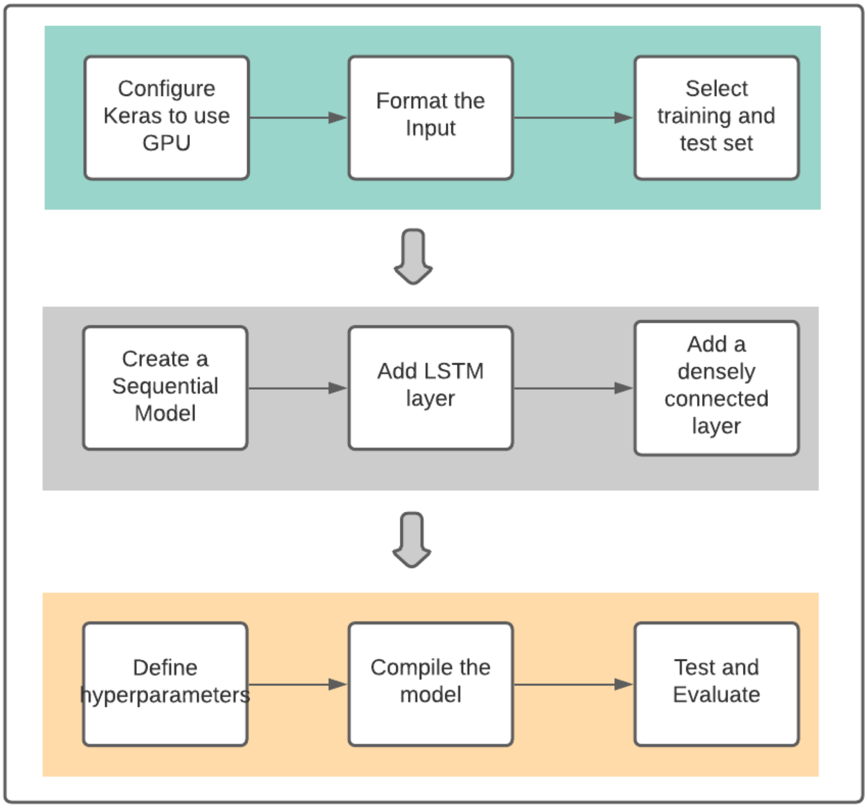

Figure 3.9 – Process flow performed by the algorithm, considering configuration with only one layer.

In the initial stages, it was noticed that the algorithms had different characteristics from their Matlab counterparts and, their configuration varied slightly. Also, the adjustment of hyperparameters was not as sophisticated as in the case of BioPatRec and was based much more on heuristics than on brute force. In other words, the combination of variables was made based on the developer's experience, as this allowed to drastically reduce training and still obtain expressive results. The study considered only simple configurations, considering the possibilities that Keras presents. The LSTM network was evaluated under the following aspects of its architecture:

- With 50 and 100 neurons;
- With 50, 100 and 200 epochs;

For the evaluation of the results, the investigation used only one set of data related to the simultaneous movements of the upper limbs.

The second network made available by BioPatRec-Py is CNN of one dimension. The algorithm tracks feature in a linear fashion instead of treating the input in a matrix way.

Table 3.6 – One of the CNN-1D network configurations.

	Format of Sets	
	Training	Test
Input	(1800, 1, 136)	(1200, 1, 136
Output	(1800, 1, 27)	(1200, 1, 27)
Layers	Output Shape	Param
Conv1D	(None, 1, 272)	37264
Dense	(None, 1, 27)	7371
Total params: 44635		
Treinable params: 44635		
Non-treinable params: 0		

Table 3.6 illustrates one of the implemented configurations and the data sets organized to feed the classifier.

The 17 characteristics sampled through 8 channels will generate four input vectors, two of which are related to training and validation, the others are applied to the test. In this way, the features are arranged sequentially, with 1800 samples and 136 characteristics for training/validation and 1200 observations for testing. The work considered the following possibilities:

- 50 and 100 epochs;
- Using 136 and 272 filters.

The study also considered a two-dimensional CNN network, as in the case of BioPatRec. In this way, the entry is not arranged in a vectorial way but in a matrix way. The characteristics were organized by the methodology in a rectangle shape, with each sample having a dimension of 136x8. The kernel value was fixed at 8, as this value accelerated the convergence, the other hyper-parameters were the same as in the previous experiment. The following code details the configuration.

```
model = Sequential()
model.add(Conv2D(filters=17, kernel_size=8, activation='tanh', input_shape=(8,17,1)))
model.add(Flatten())
model.add(Dense(27, activation='softmax'))
opt = tf.optimizers.Adam(learning_rate=0.05,
       beta_1=0.99,
       beta_2=0.999,
       epsilon=1e-3,)
model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])
```

Table 3.7 – CNN-2D network configuration implemented.

	Format of Sets	
	Training	Test
Input	(1800, 8, 17, 1)	(1200, 8, 17, 1)
Output	(1800, 27)	(1200, 27)
Layers	**Input Shape**	**Output Shape**
Conv2D	(?, 8, 17, 1)	(?, 8, 17, 1)
Conv2D	(?, 8, 17, 1)	(?, 1, 10, 17)
Flatten	(?, 1, 10, 17)	(?, 170)
Dense	(?, 170)	(?, 27)

The matrix dimensions of each layer are detailed in the Table 3.7. The network was evaluated concerning the filter size (17 or 34) and according to the maximum number of seasons (50 or 100).

3.8.3 Hyperparameters Adjustment and Statistical Analysis

This step did not use the grid search protocol from the previous section. The study chose to evaluate a limited set of values. The activation functions considered were the hyperbolic tangent and softmax. The optimization functions tested were Adam, RMSProp, and Adamax. The learning rate was adjusted freely by the author. At this stage, the research considered only arrangements with one layer since the extra addition did not bring noticeable benefits. The study also did not use pooling since the results were practically identical. One explanation for this effect is due to feature extraction, as this step considerably simplifies entry.

To calculate the training time and accuracy, the experiment used the same approach as BioPatRec, repeating the test 10 times for each individual. With these values, the study calculated the metrics and their respective standard deviations.

3.9 PRINCIPAL COMPONENT ANALYSIS AND GLOBAL TRAINING

One of the challenges of training RNAs for EMG signal is to be able to develop a system capable of operating homogeneously across the population. So far, the vast majority of research has focused its efforts on individual training methods. For each person, a training section is held, which is specific and characteristic of the individual in question. That is, the products currently supplied are individualized. This fact is somewhat limited as the calibration of the device becomes extremely specific.

Another point of the traditional approach is the generalization of networks. Each classifier is trained to learn the intrinsic characteristics of a particular person, rather than the overall population pattern. This problem makes the prosthesis very vulnerable to any variation in the signal. In other words, its usability is deteriorated due to daily weather conditions, and perhaps that is why there is still no robust model since the networks an individualized product.

Since DL techniques can describe more general and complex abstractions and patterns, it can be hypothesized that if there are global patterns in the signal it is possible to recognize them and train a generic network. What is desired by the study is to create a comprehensive training process, which considers the entire population, and at the end of the training, the developed system can be useful for anyone. This fact implies two things:

1. The product created would be able to operate in an equivalent and homogeneous way in the population, which would dispense with the need for individualization of each product and would facilitate the calibration of the device;
2. If DL techniques are successful in this proposal, they would not be subject to small fluctuations in daily use, since instead of the individual pattern, the network would be able to recognize the true and global characteristic of the movement in question;

This research focused efforts precisely to try to address this opportunity and, for that, purpose created a global training method. The technique consists of training the network in the entire data set and then assessing the overall accuracy of the population. It is enough to join the data sets in just one file and perform the other procedures in the usual way. In addition, it was necessary to employ Principal Component Analysis to select the most descriptive features for the ANNs considered. Figure 3.10 summarizes the concept [Souza, Moreno e Pimenta 2020].

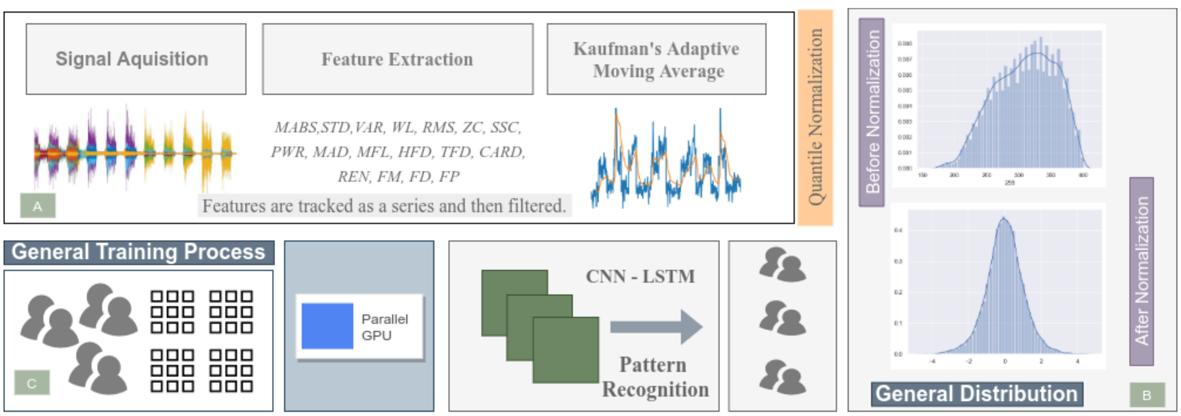

Figure 3.10 – (A) - The extracted features are filtered. (B) - Before performing the classification, the methodology standardized the entry through the quantile normalization process. (C) - The network is trained for the entire population.

4 RESULTS PRESENTATION

Success and love prefer the brave

Ovídio

The research divided the results into three parts. The first part is related to the BioPatRec tests and deals with a configuration that uses a serial processing mode through a traditional CPU. These first results have a focus on comparative statistics, on the arrangement and adjustment of networks through the grid search. The prior extraction of characteristics is tested in this stage, and the study makes a comparative study between approaches and configurations.

The second part focuses on the parallelized classification, where the proposed software BioPatRec-Py is tested by the research. This step aims to validate the feasibility hypothesis of the use of GPUs and RNAs to optimize a classification system for the myoelectric signal. The results achieved here are the result of configurations based on heuristics, to the detriment of more rigorous adjustment methods. In addition, the software introduced has its adaptive filtering scheme and a method of standardizing the data that adjusts the distribution evenly.

Finally, the results of the global training process are presented, considering the signals sampled in the entire population. This step used the same criteria as the previous one, considering the set of signals chosen for the tests and the configuration of parallelized hardware. In addition, the tests were carried out incrementally, analyzing the results with or without a given approach, such as: using the quantile normalization, using the moving average to perform the filtering, and, finally, adopting the Principal Component Analysis.

4.1 RESULTS OF THE LONG SHORT-TERM MEMORY

The LSTM network obtained high accuracy and surpassed traditional approaches, concerning the time the results were satisfactory and compatible with the functional requirements of the embedded applications. The time values achieved by the study were second only to the LDA and RFN. However, the last two classifiers were unable to overcome the network when the metric chosen is the ability to distinguish between classes. The algorithm has proven to work well with the complete feature set, so the following tests used all 17 characteristics in their settings. Initially, a feature selection protocol was adopted to choose the most representative. Therefore, instead of using the 27 features offered by the platform, the study only adopted the 17 described mathematically in Chapter Two.

4.1.1 Set of Characteristics and Population Variance

The first section of the tests examined the network capacity under different sets of features and verified in which configuration the LSTM algorithm obtained the best results. Then the study performed an analysis on each individual in isolation. Table 4.1 presents the results for the first set of tests, the values considered were the average accuracy of the population (%) and average training time (seconds). The first test considered three different feature arrangements. The first uses 17 features chosen through heuristics and tests in the initial stage. The second considers the features introduced by *Hudgins et al.* (MABS, WL, ZC, SSC) [Hudgins, Parker e Scott 1993] plus cardinality. Finally, the tests are made by analyzing the previous characteristics plus cardinality. Standard deviations accompany each of the results. The forecasting time was below 4 milliseconds in all cases and was therefore discarded in the experiments that followed.

Table 4.1 – Result of the First Experiment

	17 Features	**Hudgins + Cardinality**	**Hudgins**
Average Accuracy	96,03 ± 0,27	93,09 ± 0,22	92,48 ± 0,22
Training Time	9,69 ± 0,1	8,41 ± 0,02	8,41 ± 0,02
Forecast Time	0,005	0,004	0,004

Due to the biological variability that each individual presents, it is common to have differences in electrical signals, so the thesis investigated the capacity of the network in each person separately. These results confirm that the set of information is significant for these networks. Another relevant piece of information is that the same individuals who obtained the lowest scores in the study [Souza e Moreno 2018], considering other classifiers, also presented relatively poor results in this investigation. This observation suggests that, for this type of problem, the proper selection of characteristics, the creation of new features, and the study of techniques capable of obtaining the signal more homogeneously among the population are essential issues to ensure quality recognition of EMG patterns, both from the point of view of machine learning and myography. The results using all features were above 90% for the entire population, with some highlights being positive and others being negative. These results are shown in Figure 4.1.

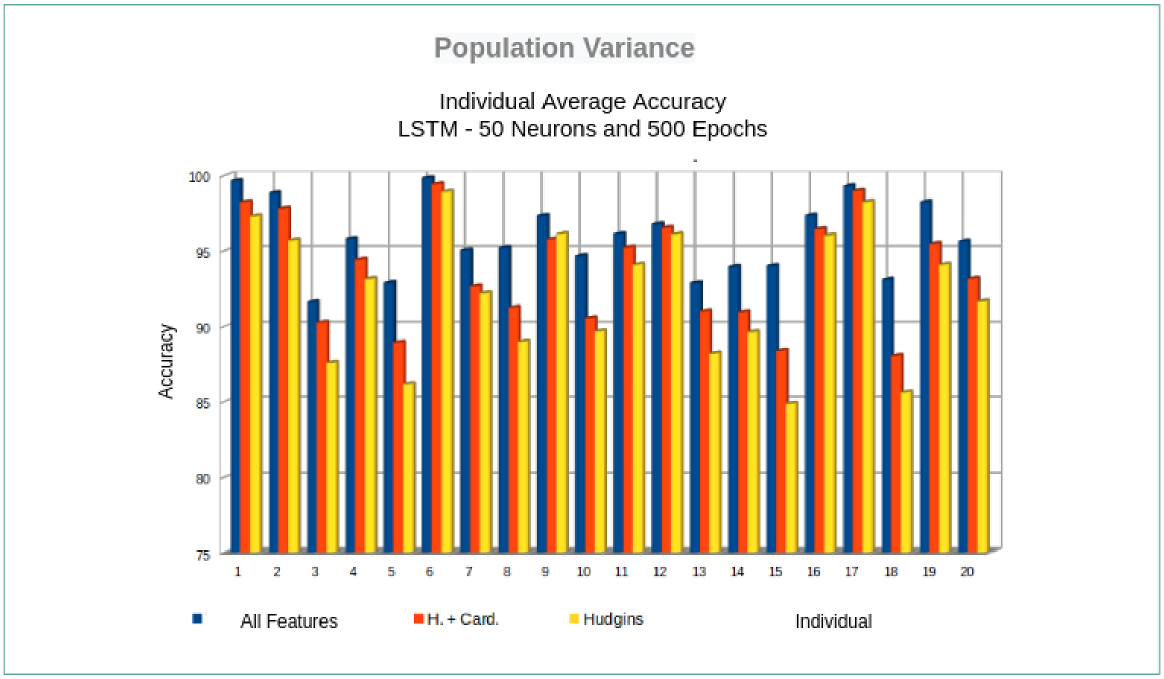

Figure 4.1 – Resultado individual obtido pelo classificador considerando diferentes grupos de *features*.

4.1.2 Number of Neurons and Epochs

The following results represent the grid search that the research adopted and considered some arrangements based on heuristics to find the best cost-benefit ratio between time and accuracy. Evaluating the number of epochs equal to 1000 the network obtained high accuracy and was not very sensitive to the relative variation in the neurons number. Therefore, the quality of the results is not strictly related to neural complexity. It is possible to obtain a balance between the number of iterations and neurons, ensuring a good separability between classes with low computational cost. The experiment analyzed the average time needed to train the algorithm, and contrary to the accuracy (except with ten neurons and 100 times), the results varied a lot. In this case, the more complex the network used, the longer the time, which followed an exponential trend. The last metric evaluated was the time needed to predict a movement. Times were less than 200 milliseconds in all cases, in fact, in the range of 4 milliseconds. Therefore, networks do not present any restrictions in this regard, regardless of the configuration. This relationship is described by Figure 4.2.

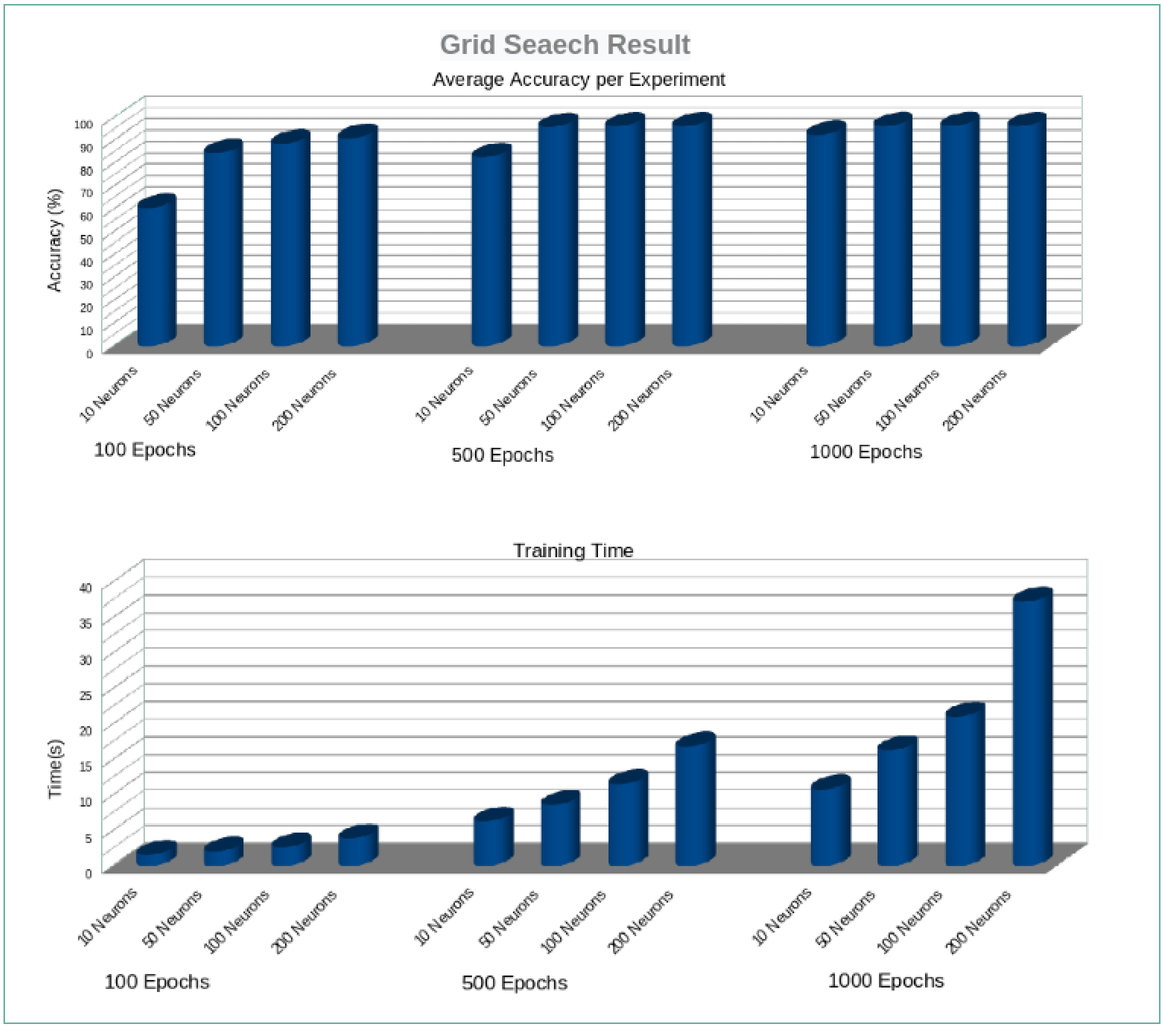

Figure 4.2 – Average accuracy obtained among the population between the tested configurations and the respective training times. The time increases as the complexity of the network increases.

Figure 4.3 shows that adopting only 50 neurons and 500 times, the results are practically equal to the maximum obtained, however, with a much shorter training time. In this way, the configuration should be preferable to others that burden the time and do not improve the classification result. These results can be used as a reference in future works and suggest that it is possible to embark on a DL technique in an intelligent prosthesis without worrying about adverse factors found in other surveys, which obtained training times over 17 minutes [Laezza 2018].

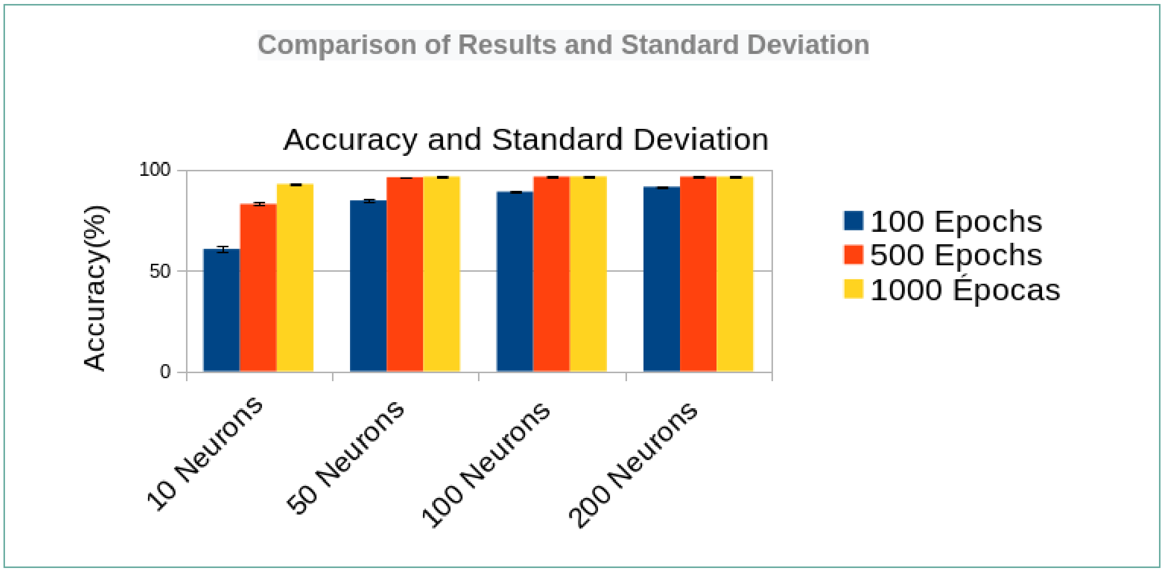

Figure 4.3 – Average accuracy obtained among the population between the tested configurations and the respective training times. The time increases as the complexity of the network increases.

4.1.3 Hyperparameter Adjustment

As there are many values to be adjusted, the methodology employed a network with 100 neurons and 100 epochs to test different Learning Rate values and the Squared Gradient Decay Factor. Several values were evaluated for individuals 3, 5, and 13. At the end of the tests, 0.0005 was chosen for the Learning Rate and 0.9999 for the Decay Factor. The experiment was repeated using these parameters for the entire population, and the result jumped from 88.78 % to 96.14 %. The average training time was 2.81 seconds against 2.63 considering the network without adjustment. To verify if there was a statistical variation the paired t-test *($p < 0.05$)* was used between the two sets of results, and the hypothesis proved to be true. Qualitatively the Table 4.2 summarizes the outcomes of this battery of tests.

4.1.4 Combined Movements

The following results illustrate the averages obtained using signals relative to movements with more than one degree of freedom. The values chosen were the same as in the previous experiment (decay factor of 0.9999 and learning rate of 0.0005), and this shows that there was a general empirical improvement. The times required for training are summarized in the Figure 4.4. Although the network takes more time in the training process with the parameters set, in no case did the time exceed 40 seconds. In addition, it is feasible to choose a configuration that meets the requirements of time and the ability to classify elegantly and assertively. Table 4.3 presents the quantitative results and the respective standard deviations.

Table 4.2 – Results Table - Individual Movements

	Neurônios	Accuracy (%)	Training Time (s)	Forecast Time (%s)
100 Epochs	10	60,65 ± 1,56	1,57 ± 0,02	0,0037
	50	84,62 ± 0,55	2,05 ± 0,01	0,0041
	100	88,78 ± 0,35	2,63 ± 0,001	0,0041
	100*	96,14 ± 0,03	2,81 ± 0,01	0,0042
	200	91,22 ± 0,32	3,89 ± 0,04	0,0041
500 Epochs	10	82,85 ± 0,75	6,28 ± 0,04	0,0038
	50	96,03 ± 0,14	8,63 ± 0,03	0,0042
	100	96,32 ± 0,19	11,51 ± 0,07	0,0043
	200	96,32 ± 0,23	16,81 ± 0,03	0,0041
1000 Epochs	10	92,59 ± 0,49	10,79 ± 0,02	0,0036
	50	96,35 ± 0,26	16,32 ± 0,08	0,0041
	100	96,4 ± 0,2	21,05 ± 0,16	0,0041
	200	96,33 ± 0,21	37,23 ± 0,34	0,0045

The investigation adopted the paired t-test between each pair of experiments, and the results confirmed the hypothesis that there is a statistical difference ($p < 0.05$).

Table 4.3 – Results Table - Simultaneous Movements

	Neurons	Adjusted hyperparameters		Standard Hyperparameters	
		Accuracy (%)	Training Time(s)	Accuracy (%)	Training Time(s)
50 Epochs	50	96,10 ± 0,14	2,65 ± 0,02	66,19 ± 0,86	2,52 ± 0,02
	100	96,77 ± 0,25	3,33 ± 0,04	77,17 ± 0,49	3,32 ± 0,05
100 Epochs	50	97,05 ± 0,13	4,75 ± 0,05	78,61 ± 0,76	4,17 ± 0,05
	100	97,42 ± 0,25	6,77 ± 0,07	89,02 ± 0,45	5,43 ± 0,04
200 Epochs	50	97,40 ± 0,18	17,61 ± 0,11	91,08 ± 0,37	7,59 ± 0,07
	100	97,62 ± 0,24	39,82 ± 0,15	96,13 ± 0,16	9,92 ± 0,08

Figure 4.4 presents the results for different configurations qualitatively. The image provides a comparison between the networks that used the parameters selected in the grid search with those where the architecture used the Matlab default values.

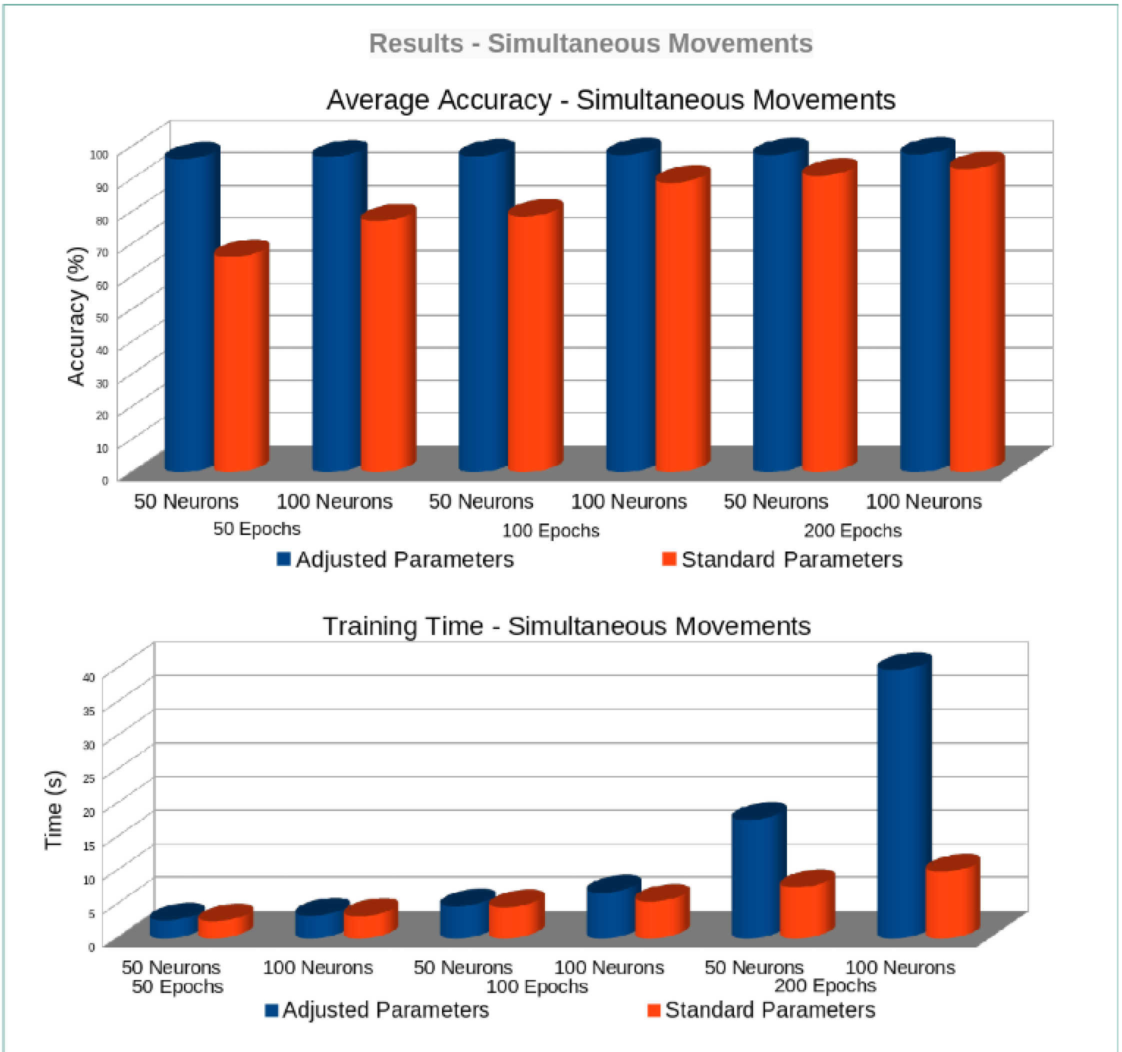

Figure 4.4 – Average accuracy per experiment and average training time considering the set of signals with composed movements and two distinct hyper-parameter configurations.

4.2 CONVOLUTIONAL NEURAL NETWORK OUTCOMES

The study tested the CNN network in parallel with other general-purpose classifiers available on the platform, and the experimental protocol was the same adopted by LSTM. The qualitative results of the entire experiment are available in Table 4.4.

Table 4.4 – General Result of the CNN Network Investigation - Accuracy

Dataset	10mov4ch	10mov4ch-2	Simultâneo
Algorithm	**Accuracy (%)**	**Accuracy (%)**	**Accuracy (%)**
LDA	89.16 ± 0.26	93.28 ± 0.29	95.18 ± 0.07
SOM	93.56 ± 0.23	95.59 ± 0.30	95.19 ± 0.61
RFN	87.25 ± 0.24	90.95 ± 0.45	83.84 ± 0.28
MLP	91.75 ± 0.40	93.28 ± 0.29	93.02 ± 1.00
CNN_A	96.34 ± 0.31	98.46 ± 0.17	97.03 ± 0.24
CNN_B	$\mathbf{96.91 \pm 0.27}$	$\mathbf{98.68 \pm 0.30}$	$\mathbf{97.42 \pm 0.16}$
Dataset	**Lower - TMC**	**Lower - TBC**	**Lower - UMC**
LDA	95.49 ± 0.25	96.03 ± 0.35	$\mathbf{96.76 \pm 0.24}$
SOM	79.34 ± 0.59	83.31 ± 1.06	81.61 ± 0.74
RFN	85.89 ± 0.60	88.82 ± 0.70	88.45 ± 0.57
MLP	91.66 ± 0.55	92.63 ± 0.48	91.83 ± 0.84
CNN_A	96.76 ± 0.32	97.40 ± 0.34	96.56 ± 0.32
CNN_B	$\mathbf{97.19 \pm 0.22}$	$\mathbf{97.91 \pm 0.24}$	96.42 ± 0.2

4.2.1 Individual Movements

In the two individual movement sets, the CNN-B and CNN-A networks obtained high accuracy (98.68% and 98.46%, 96.91% and 96.34%), surpassing all other methods. The application of the paired t-test was used between the CNN-B network and the SOM method to verify the existence of a statistical difference between the presented concept and the best-evaluated algorithm. The result of the evaluation guarantees the validity of the hypothesis because there was a statistical difference between the two results.

Regarding the training time, the CNN-A and CNN-B networks took 1.86 and 2.76 seconds, considering the first set and 1.75 and 2.63 seconds, concerning the second sample collection. Despite not showing the best results of the investigation, the overall performance was good and is sufficient to be shipped on a device with limited hardware features. The MLP network had the worst result in this test battery, and its average training time was above 17 seconds in both collections. In this case, the paired t-test was left aside, as the difference in values was substantial, it is possible to intuitively infer that the hypothesis of statistical difference exists. The comparative results are displayed graphically in Figure 4.5.

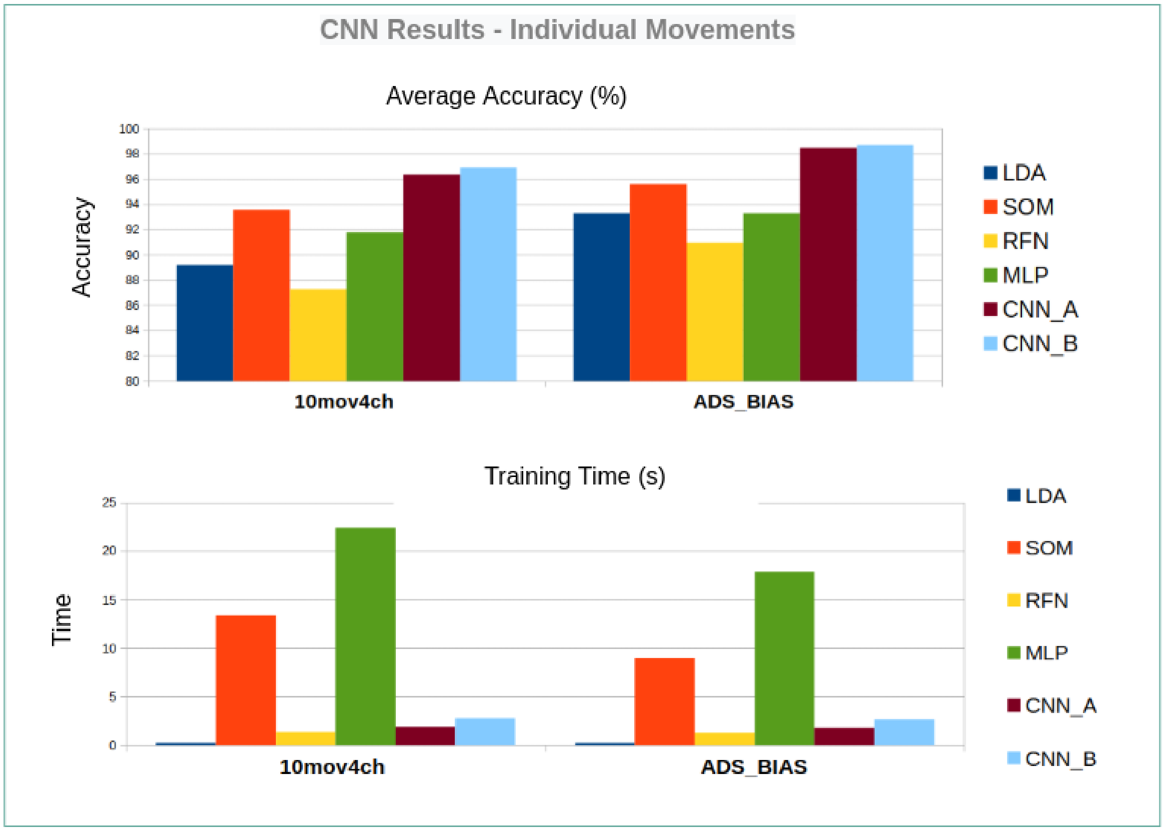

Figure 4.5 – Comparative results between traditional networks and the CNN Deep Learning technique proposed by the investigation.

4.2.2 Simultaneous Movements

Evaluating the results representing movements with up to 3 degrees of freedom in their composition, both CNNs obtained high accuracy. The results were the best of the study in this regard. The CNN-B arrangement reached an average of 97.42 % between the population and the CNN-A network obtained 97.03 against 95.19 achieved by the SOM method. The investigation applied the paired t-test between the two samples, and again, there was a statistical difference *(p <0.05)*. The training times obtained by the CNN-A and CNN-B networks were 3.46 and 6.90 seconds and again not the best in the test battery. The LDA network achieved the shortest time and took only 0.32 seconds in its average training process. This result is because of its low mathematical complexity, which is not built on elements in parallel and would eventually not benefit from parallelized hardware. The worst time was obtained by the MLP network that spent more than 1 minute (average) for each training. Figure 4.6 allows the reader to ascertain the commented differences.

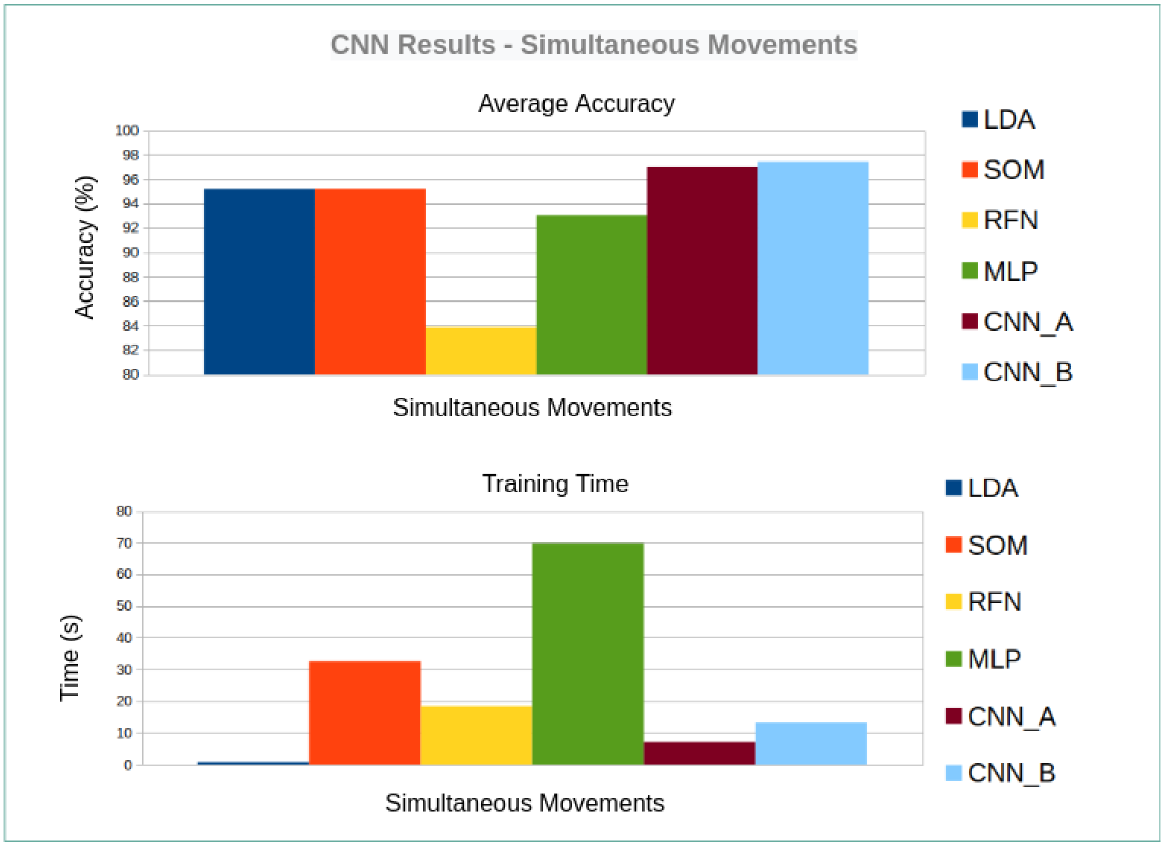

Figure 4.6 – Result of the experiment considering only the collection of individual movements.

4.2.3 Lower Limbs

The last section of the tests assessed the ability of the network considering three sets of signals, which represent movements of the lower limbs. The collections consider different arrangements regarding the configuration of the electrodes and can be used to understand the effect of the signal capture on the model. In the first group (TMC), the networks CNN-A and CNN-B obtained the best results of the investigation, reaching 96.76% and 97.19% precision, respectively, followed by the LDA classifier with 95.49% precision. The work employed the t-test and found that the hypothesis of difference existed between them. When analyzing the time, both networks obtained satisfactory values, taking 2.83 and 5.29 seconds. The LDA network achieved the best result with only 0.31 seconds. The MLP algorithm took 31 seconds, on average, to perform a training operation, presenting the worst value among the classifiers employed. Analyzing the second set of signals (TBC), the hybrid CNN obtained the best accuracy, with average values of 97.40% and 97.91%, respectively configurations A and B. The LDA was responsible for the third position with 96.45% correctness between movement classes. The paired t-test guaranteed that there is a statistical difference between both algorithms. The average training time was 2.96 and 5.31 seconds for the classifiers introduced in this study, the best time was obtained by the Discriminant Analysis, with an average time of 0.30 seconds, and the SOM classifier was responsible for the longest training time, with 28 seconds on average. The CNN network did not present the best results in the UMC collection, yet it was able to distinguish the movements of the lower limbs with 96.56% and 96.42% correct, and the training process required less than 7 seconds in both cases. In this regard, the LDA surpassed all other methods and reached an accuracy of 96.76%, with an average training time of 0.32 seconds. The application of the t-test ensured that there is a statistical difference. The results are in Figure 4.7.

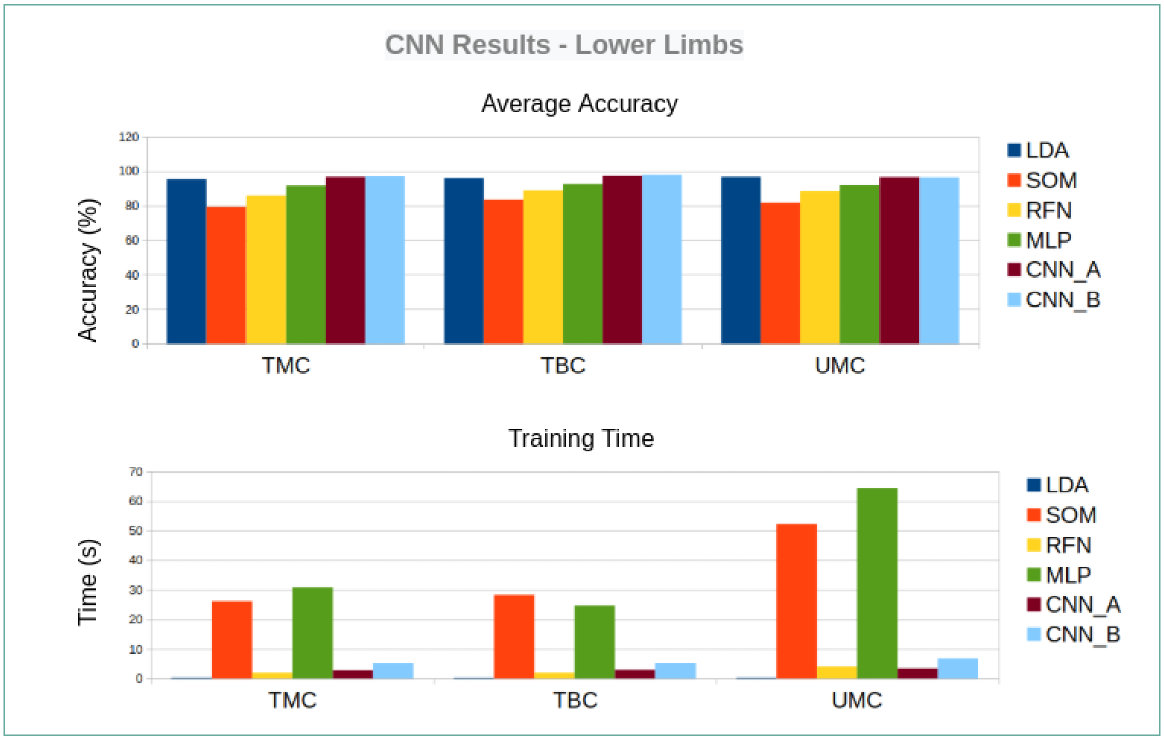

Figure 4.7 – List of results considering the last three sets, corresponding to individual movements of the lower limbs.

4.3 BIOPATREC-PY RESULTS

The results of the LSTM-Py network running in parallel through the 324 CUDA cores were satisfactory and met all expectations regarding training time, which was one of the initial assumptions. The time approached the LDA and did not reach 2 seconds in any case. This result shows that the parallelization allowed a considerable reduction in this metric and, even so, maintained the high accuracy rate.

Considering a network with 100 neurons and 100 epochs, the average accuracy was 97.62%, and the training time was 0.81 seconds. In the beginning, the objective was to perform the complete grid search. However, it was noticed that the results did not vary much between one configuration and another. The immediate response to this behavior is related to the two methods employed: normalization Quantil and Kaufman's adaptive filtration. Therefore, it was decided to carry out a leaner experiment. Table ref LSTM-Py presents the quantitative results [Souza, Moreno e Pimenta 2020].

Table 4.5 – General Result of the Investigation of the LSTM Network - Parallelized

Network	**Accuracy**	STD	**Time**	**STD**
50 Neurons 50 Epochs	97.15%	0.122	0.44s	0.007
100 Neurons 100 Epochs	97.62%	0.08	0.81s	0.028
100 Neurons 200 Epochs	97.64%	0.07	1.34s	0.013

This work made a last effort to test another recurrent neural network and created a similar approach to evaluate a GRU-type network. In this case, the algorithm had only 50 units, and the other configurations were the same as the LSTM classifier. This configuration was chosen after some tests based on heuristics, aiming to reduce time and increase accuracy. The GRU network achieved 94.56% accuracy and training time equal to 2 seconds.

4.3.1 1D Convulutional Neural Network

The convolution operation has a nature that allows the algebraic parallelization of its iterations and, this fact was evidenced in the results. The networks obtained high accuracy with an average training time of less than 1 second in all configurations. Figure 4.8 shows the qualitative results of this classifier, and Table 4.6 shows the quantitative results.

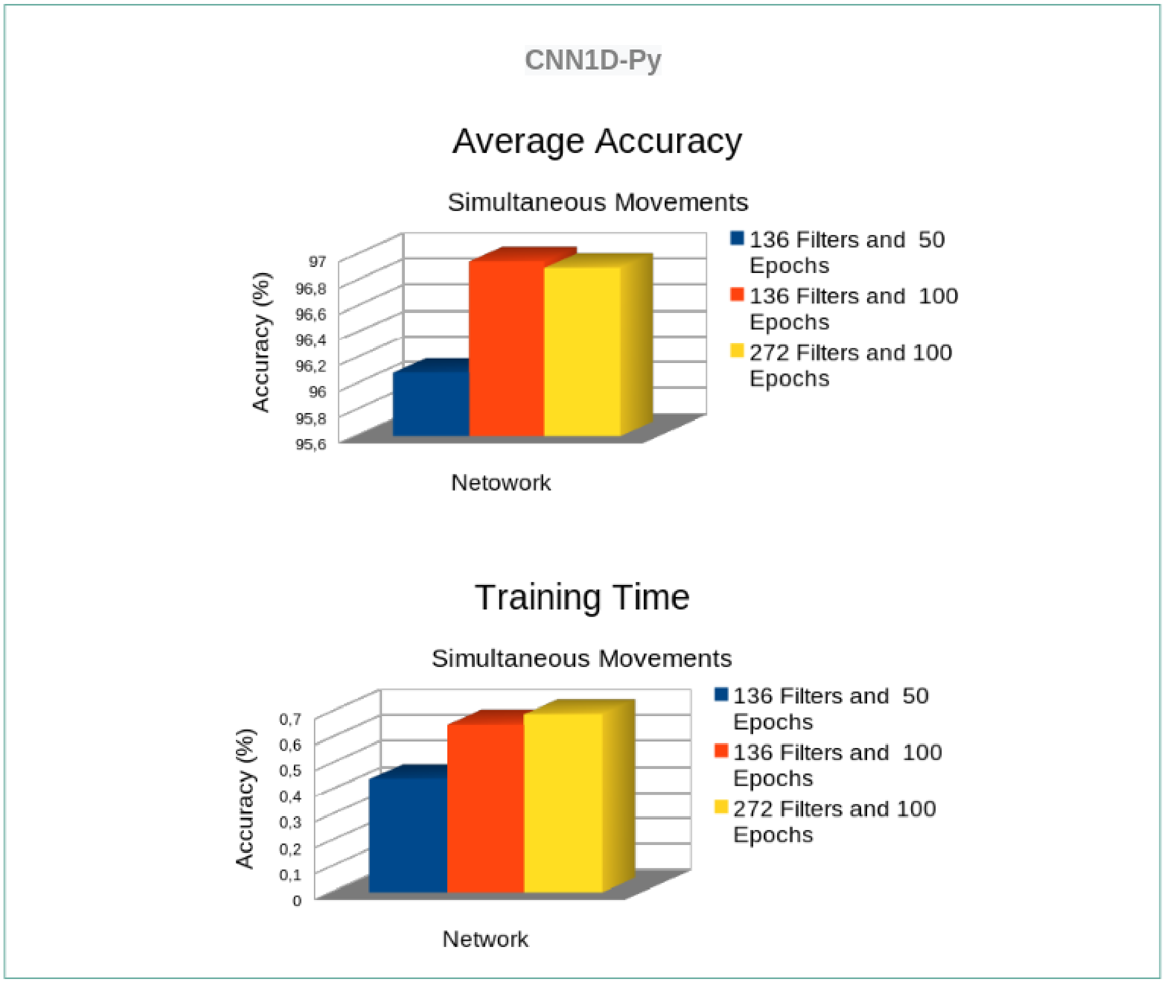

Figure 4.8 – Results presented by the one dimension Convolutional Neural Network.

The network with 136 filters and 100 times obtained an average accuracy of 96.93% among the population analyzed the training time was 0.65 seconds [Souza, Moreno e Pimenta 2020]. Analogously to the previous case, there was not much variation between one configuration and another, which again suggests that the adopted features engineering can present data more homogeneous to the algorithm. Another important fact is that the algorithms implemented in Python using GPU were not rigorously evaluated concerning their hyper-parameters. The research did not fine-tune this step and considered arrangements based on the heuristic of previous experiments.

4.3.2 2D Convolutional Neural Network

The two-dimensional convolutional neural network also showed interesting results, although slightly inferior to the previous classifier. The behavior concerning the number of filters was relevant since the network did not need many filters to solve the problem and,

Table 4.6 – General Result of the CNN1D Network - Parallelized

Network	Accuracy	STD	Time	STD
136 Filters 50 Epochs	96.44%	0.09	0.44s	0.019
136 Filters 100 Epochs	96.93%	0.06	0.65s	0.019
272 Filters 200 Epochs	91.10%	0.06	0.69s	0.02

therefore, the study used only 17 or 34. Again, the training time did not exceed 1 second in the parallelized approach. The graphs with the results are in Figure 4.9.

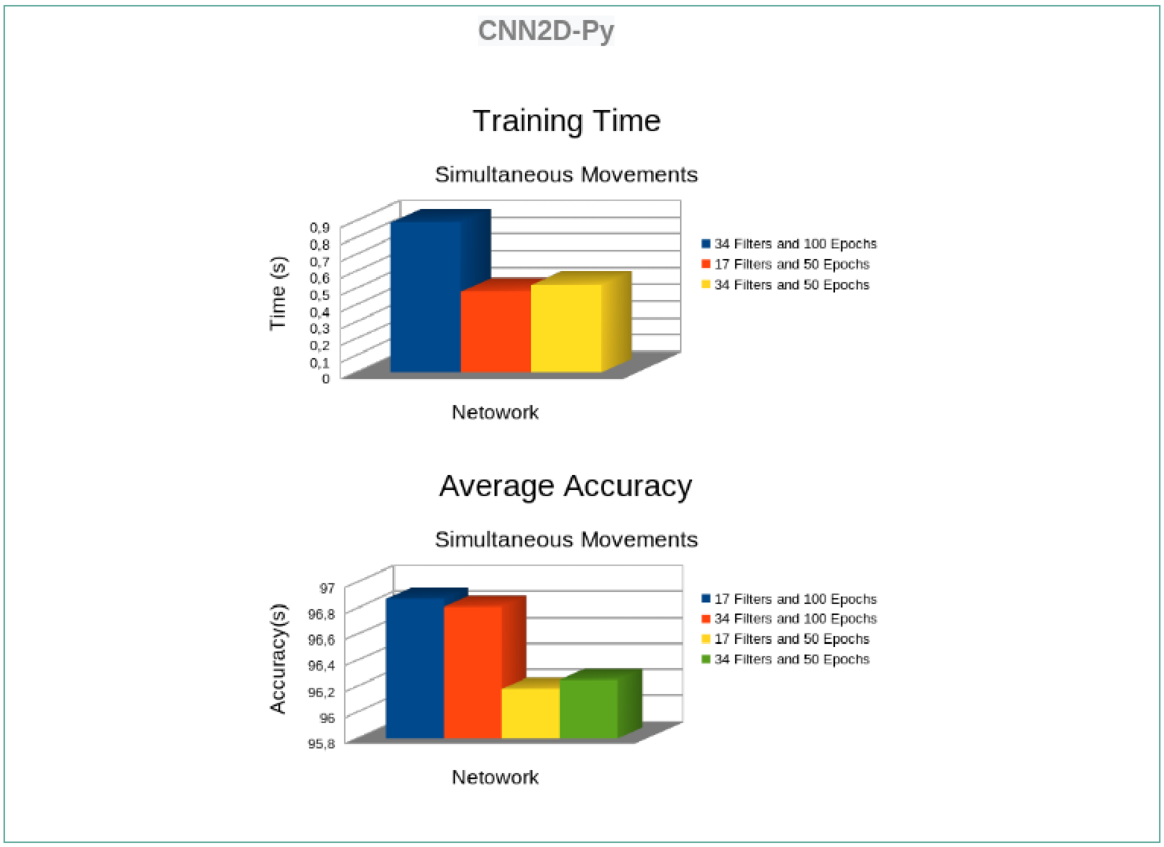

Figure 4.9 – Results presented by the two dimension Convolutional Neural Network.

A network with 17 filters and being trained over 100 times obtained an accuracy of 96.87 % in the distinction between 27 simultaneous movements (1200 samples), with a training time of 0.78 seconds [Souza, Moreno e Pimenta 2020]. The complete list of results is in Table 4.7.

Table 4.7 – General Result of the 2D-CNN Network - Parallelized

Network	Accuracy	STD	Time	STD
17 Filters 100 Epochs	96.87%	0.062	0.78s	0.020
34 Filters 100 Epochs	96.76%	0.088	0.89s	0.017
17 Filters 50 Epochs	96.82%	0.13	0.41s	0.015
34 Filters 50 Epochs	96.85%	0.10	0.62s	0.014

4.3.3 Global Approach

As each individual has its peculiarities regarding EMG, it was decided to create a generic approach, which is not usual, since this kind of study always consider the individual approach. The goal is to train a network that can work for anyone. For this, the study collected all the signals from the 17 participants and generated a single spreadsheet. The algorithm used a large amount of information in the training process and, hypothetically, it must be able to discriminate movements regardless of the person who will use the prosthesis. The Thesis used the LSTM network (100 neurons) to perform the classification. The size of the batch was equivalent to the entry divided by 128, and the maximum number of epochs was 20 [Souza, Moreno e Pimenta 2020].

First, the research evaluated the algorithm without the filtering method and without Quantil normalization (normalization *MinMax*). The algorithm took an average of 3.6 seconds in the training phase, considering the entire population, and the average accuracy was 46.38%. Then, we introduced Quantil normalization, which resulted in a 3.89-second workout and a 63.7% accuracy. In the next step, the study added the filter, and the accuracy reached 77.9 %, the training time was 3.95 seconds [Souza, Moreno e Pimenta 2020].

As the idea of creating a general approach is tempting, the study looked for ways to compensate for unsatisfactory accuracy and used the PCA to do so. The algorithm reduces the dimensionality and presents the network with the most relevant independent n-components (the study employed 130). The methodology configured the PCA with the option *whiten*, which transforms the input variables into a new set, where the covariance is the identity matrix. Therefore, the PCA ensures that the input is not correlated, with a variance equal to 1 [Souza, Moreno e Pimenta 2020].

Adopting the PCA and the other proposed concepts, the average accuracy was **97.83**% performing generic training in the entire population. The average time reached was 4.01 seconds. When repeating the experiment without the filter and the normalization step, the accuracy drops to 74.4%. The results show the effectiveness of the introduced concept, which can be employed to create a generic approach. In addition, global training suggests the network's ability to better abstract the relationship between features, which, ultimately, can provide a more robust model [Souza, Moreno e Pimenta 2020].

4.3.4 Robustness Analysis

As mentioned in Chapter Two, the robustness of a classification system is not limited to accuracy. In this case, we must take into account the natural variability of the system. Aiming to evaluate the approach in a scenario closer to reality, the research progressively added noise to the test set. The idea is to emulate everyday use by contaminating the signal with normally distributed random data since such distributions happen frequently in nature.

The function that generates the noise receives three parameters: the mean of the distribution (center), the standard deviation, and the size of the sample that should be generated. The first parameter was left at 0, as we don't want to shift the Gaussian, which would result in values that don't match the actual use case, as this shift would be too much. The second parameter was progressively tested in 5 configurations (0.1, 0.2, 0.3, 0.4, 0.5). The idea is to increase the noise and visualize the difference between the clean and the contaminated signal. Finally, we performed the network evaluation with the modified signal to verify the robustness. The Figures 4.10, 4.11, 4.12, 4.13 and 4.15. Noise is proportional to σ.

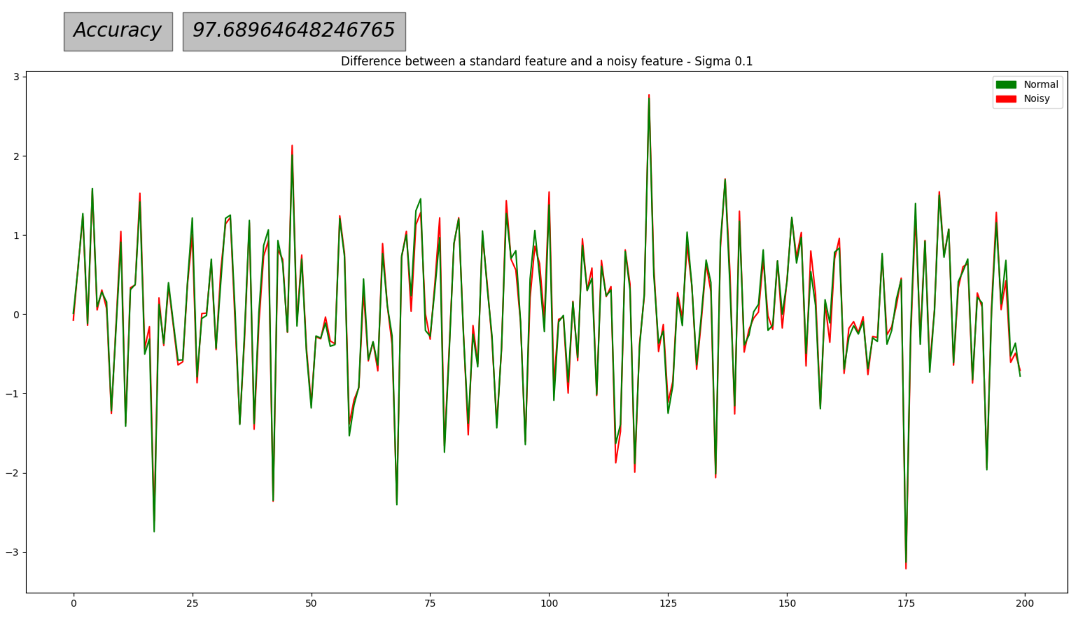

Figure 4.10 – Result of the robustness evaluation with $\sigma = 0.1$. Difference between two equivalent features, except for added random noise.

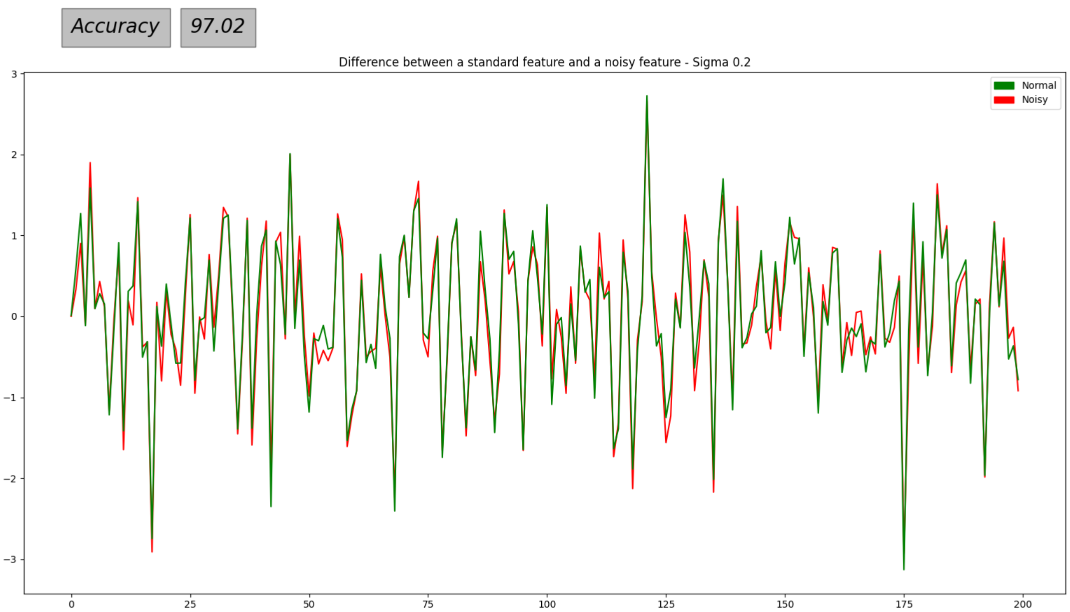

Figure 4.11 – Result of the robustness evaluation with $\sigma = 0.2$.

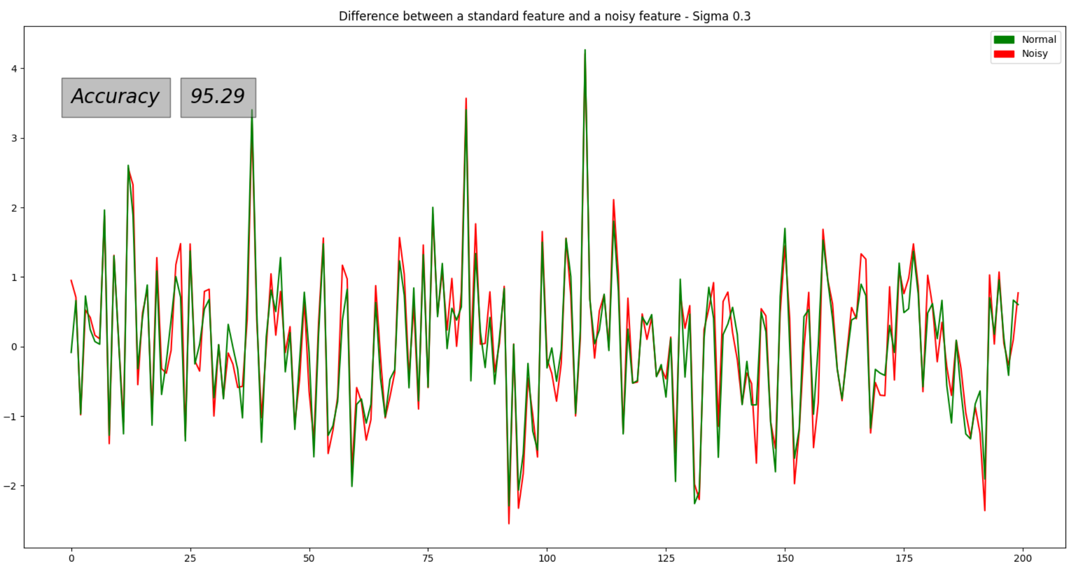

Figure 4.12 – Result of the robustness evaluation with $\sigma = 0.3$.

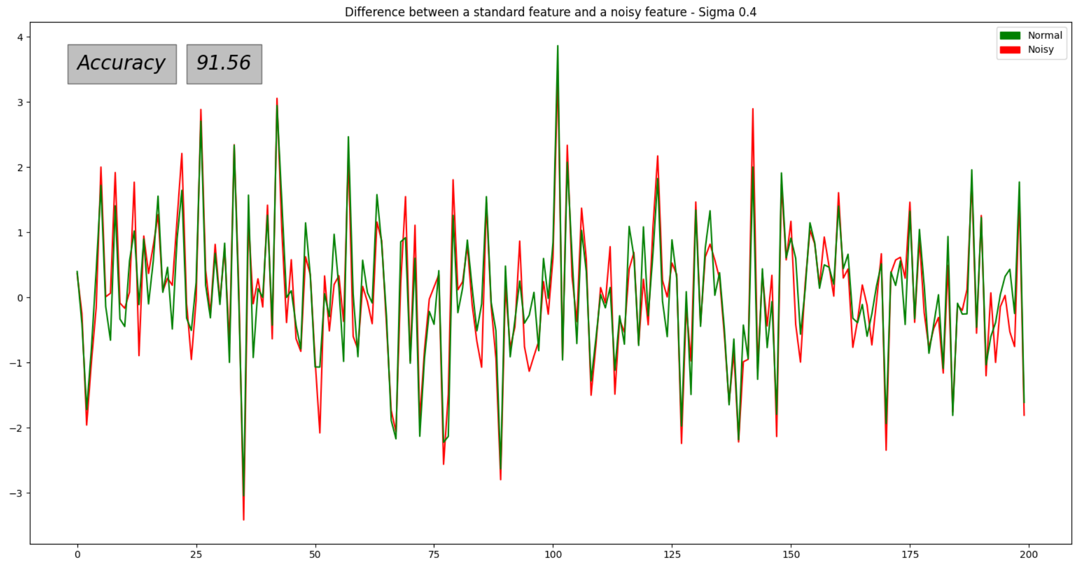

Figure 4.13 – Result of the robustness evaluation with $\sigma = 0.4$.

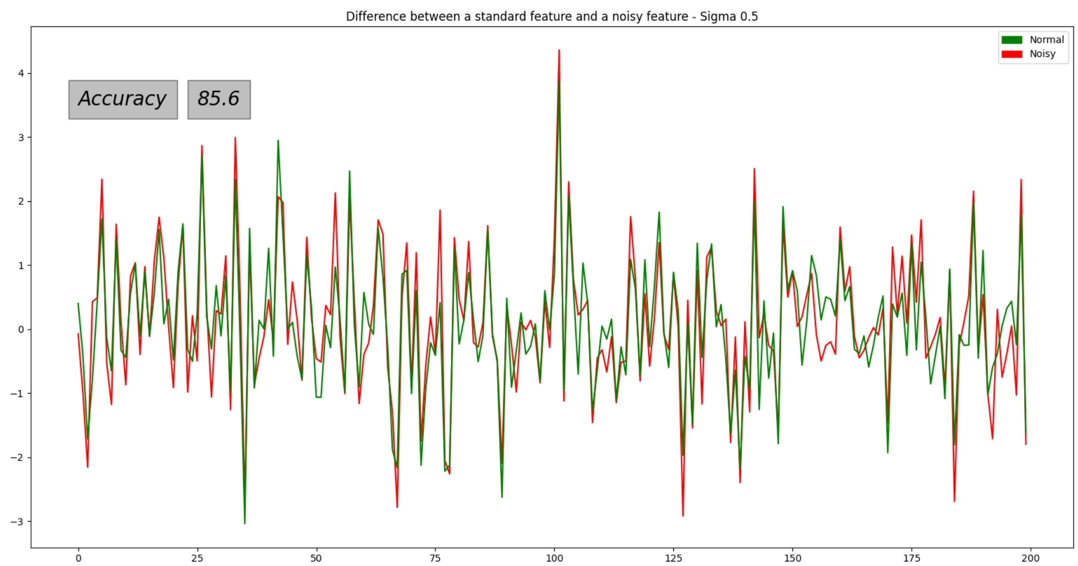

Figure 4.14 – Result of the robustness evaluation with $\sigma = 0.5$.

As the reader can verify, the network still got high accuracy, except in the case where $\sigma = 0.5$, where the noise is high. This test suggests good system robustness since the induced artificial variability did not affect the accuracy so much. This observation ultimately indicates that the system could mitigate some of the limitations of everyday use.

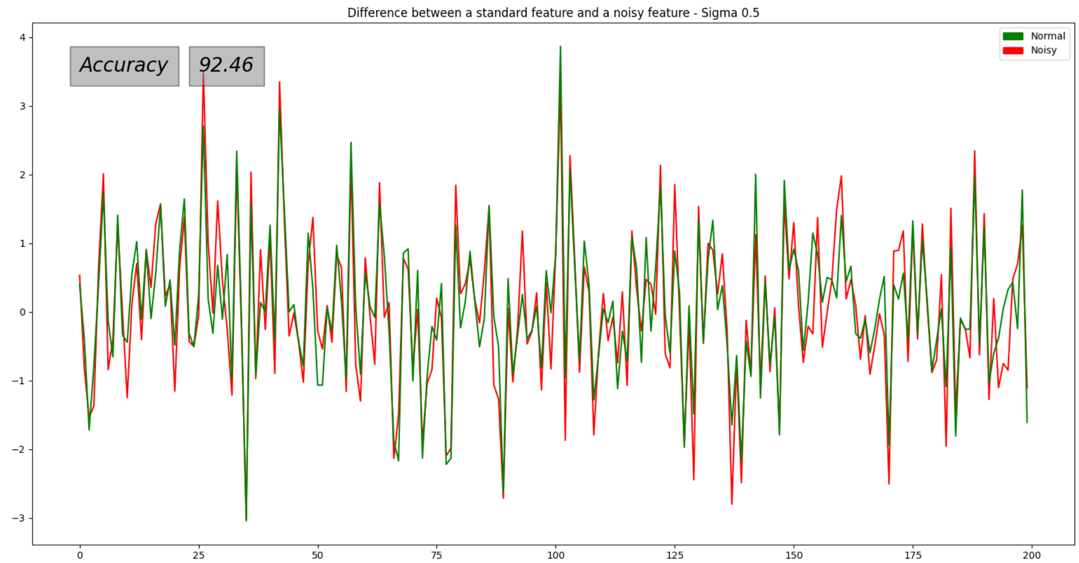

Figure 4.15 – Result of the robustness evaluation with $\sigma = 0.5$.

Aiming to overcome the accuracy drop, the author modified the architecture of the network, which now has 350 neurons and trains in 100 epochs. Furthermore, the batch was divided by 4. The modifications resulted in an accuracy of 92.46%, but the training increased to 22 seconds. The results show that the approach was able to handle a distinctly different noise from the original series. It would be interesting, in future works, to quantify and better detail the real noise so the robustness could be quantified adequately. BioPatRec-Py software is available at https://github.com/gabrielcirac/BioPatRec-Py.

5 ANALYSIS AND DISCUSSION OF RESULTS

In a discussion, it is necessary to diminish the passion.

Leandro Karnal

At first, the investigation was able to integrate BioPatRec in a hybrid manner with two recognized Deep Learning techniques, and the software used was Matlab. The networks showed good resolution skills in the classification process of the myoelectric signal, surpassing the traditional networks when the target metric was the accuracy of distinction between movements. In addition, the method of extracting features a priori proposed by the investigation proved to be feasible since it was able to increase the quality of the results and substantially decrease the training time. This fact implies two things: the first is the feasibility of embedding solutions in a parallelized device, the second shows the relevance of resource engineering in the broader context of AI. The work employed a solid statistical base in the conduct of the experiment, and the networks tested comparatively represent the state-of-the-art in the myography environment. Therefore, the proposed approach is a candidate to be used in an intelligent prosthesis.

In the second stage of the study, the premise of using a parallelized device was tested, and for that, the methodology developed a platform called BioPatRec-Py. The introduced system adopted a unique filtering system (Kaufman), and this ensured that the entry would become homogeneous among the evaluated individuals [Souza, Moreno e Pimenta 2020]. This perception emerged during the initial tests, where the variability of the results was less accurate. In addition, the filtering process benefited the training convergence. The initial idea of the research was to employ an FPGA dedicated to AI that adopts a high-level framework (Python, Keras, Tensor Flow.) to perform the HDL synthesis. However, such devices are still expensive, and that is why were left for future research. Given the impossibility of obtaining the referred hardware, the study acquired a GPU to parallelize the system. The methodology employed the same software technology that smart FPGAs adopt, and the results can be replicated for that hardware. The study validated the initial proposal to use parallelized devices in consortia with DL techniques and applied it to the case of the myoelectric signal. The results achieved by the research met the requirements of time and precision with elegance and robustness and were able to overcome all the approaches currently available. Its possible limitation arises about the real test of the application, as the method was not used in an embedded prototype, it is impossible to describe its exact behavior under such circumstances. However, as it was possible to train a global model, it can be said that the concept is robust and meets all requirements.

5.1 LONG SHORT-TERM MEMORY NETWORK

LSTM networks have persistence and can retain information in a chain for long periods this was evidenced by how they dealt with the sets of characteristics since it obtained the best results when the experiment used all the information in the tests. The classifier does not need to perform the previous features extraction (as long as the inputs are correctly sized, as in the case of CNN) and can manipulate the characteristics in its training process. However, obtained promising results using the methodology suggested. It is possible to observe the ability to capture information about another perspective: comparing the individual and compound movements results. Intuition leads us to think that the network would reach lower values in the classification of simultaneous samples since there is a natural tendency for overlap between classes. However, what is observed is the opposite. The network is better when the dataset is large (regardless of the number of moves). In this case, the signals representing movements with up to 3-DoF had 30MB per individual, against 3MB in the case with only 1-DoF. This difference demonstrates one of the main characteristics of deep neural networks which is to obtain a better solution in larger datasets.

Regarding population variance, the study confirmed that the extraction process with homogeneous characteristics, individual brain plasticity, and the person's willingness to perform the activity impact the quality of the results. Thus, the step that precedes the classification directly affects its processes and shows that it is still necessary to study more robust ways of acquiring information. This statement originates from experimental observation, where the participants who obtained the worst results in this investigation were the same as in other similar work [Souza, Pimenta e Moreno 2020]. Therefore, several factors and not just an algorithm limit the classification.

Another point assessed was the cost-benefit ratio between the number of neurons and the maximum times allowed for the network. A complex arrangement is not strictly necessary to resolve this type of classification. Using 500 epochs and 100 neurons, the algorithm achieved 96.32% accuracy with a training time of 11.51 seconds for individual movements. Using 100 epochs and 100 neurons, the network was able to predict with an accuracy of 96.14%, and it took 2.81 seconds to train the model (adjusted network). These results suggest that the equilibrium ratio is fundamental to preserve the consumption of hardware resources since they are limited to a mechanical arm.

5.1.1 Network Architecture

Due to the large number of variables that the experiment tested the choice of hyper-parameters was made based on only three individuals, and despite the triviality employed, the effects were quantitatively good and presented statistical relevance. The results spread to other people, drastically reduced the experiments, and the hypothesis was confirmed when repeated in the entire population. In individual actions, the adjusted configuration obtained an accuracy of 96.14% with a training time of 2.81 seconds against 88.78% and 2.63, respectively. For movements with more than one degree of freedom, each set of tests was performed in two stages, without the parameters adjusted and with the new values. In all cases, the accuracy presented by the refined model was better. Under 100 neurons and 100 times, the adjusted network obtained 97.42 % of correctness and training time of 6.77 seconds, against 89.02 % and 5.43 seconds for a primary network. The only downside was time; When a very complex network is employed by methodology, the metric tends to grow exponentially (probably due to the lower learning rate). Finally, it is possible to mention that the field of hyperparameter adjustments is potentially relevant, and the investigation beyond the scope of this work would be interesting.

Table 5.1 shows the comparison of the LSTM algorithm with other studies. The work developed the investigation considering similar test protocols (the same set of signals in most cases). The LSTM network managed to surpass all other classifiers, except for the Netlab MLP method, which presented an accuracy of 0.9% greater than the proposed DL technique. Regarding the training time, the LDA classifier is a reference and carried out the process in less than 1 second. However, when the metric is the accuracy, the linear discriminant analysis is overcome by several algorithms. Also, both LDA and Netlab MLP would not benefit from the use of an FPGA/GPU. In the case of LDA, there is no reason for parallelization since Netlab MLP is not maintained by the Frameworks that this technology employs, which makes its use obsolete (considering this type of hardware).

Table 5.1 – Comparison Table

Individual Movements			
Network	**Accuracy. (%)**	**Time (s)**	**Article**
LSTM	96,14	2,81	-
LDA	92,10	0,15	[Prahm et al. 2016]
MLP	91,20	164,1	[Prahm et al. 2016]
RFN	83,8	0,55	[Prahm et al. 2016]
GLM	94,58	1,15	[Souza, Moreno e Pimenta 2018]
Netlab MLP	96,9	3,26	[Souza e Moreno 2018]
Simultaneous Movements			
LSTM	97,42	6,77	-
LDA	93,8	0,35	[Prahm et al. 2016]
MLP	94,1	172	[Prahm et al. 2016]
Netlab MLP	98,3	1,88	[Prahm et al. 2016]

5.2 CONVOLUTIONAL NEURAL NETWORK

The idea of automating the process of extracting characteristics is tempting since this step requires time, statistical knowledge, computational resources and implies the constant manipulation of data. One of the inherent qualities of the CNN network is precisely to provide the programmer with a simple and abstract way to eliminate the mentioned step, especially when the information is in a matrix rather than a linear way.

Some researchers (and this one too) used the convolutional method in the problem of the myoelectric signal to obtain a model capable of classifying the movements as accurately as possible. However, the researchers face two drawbacks, the inferior quality of the results and excessive training time [Laezza 2018]. At the beginning of this research, the methodology implemented the Deep Learning technique in the same way. The electrical signal was tracked as an image, and the results were discouraging since the accuracy was poor and the time was too long. Then, it was possible to perceive that two problems hindered the classification process. First, the images of the electrical signals corresponding to each movement are similar and do not have an obvious distinction (different from an image of a pedestrian and a car, for example). Second, there was an irrelevant amount of pixels for the problem, causing the time to increase. Figure 5.1 shows this similarity. The differences between one movement and another are subtle, and this may have made the classification process more difficult. That is why the resource extraction process makes sense in this context. In addition to reducing variability and compressing input, the method provides information to the network that increases the separation limit between classes and speeds up training.

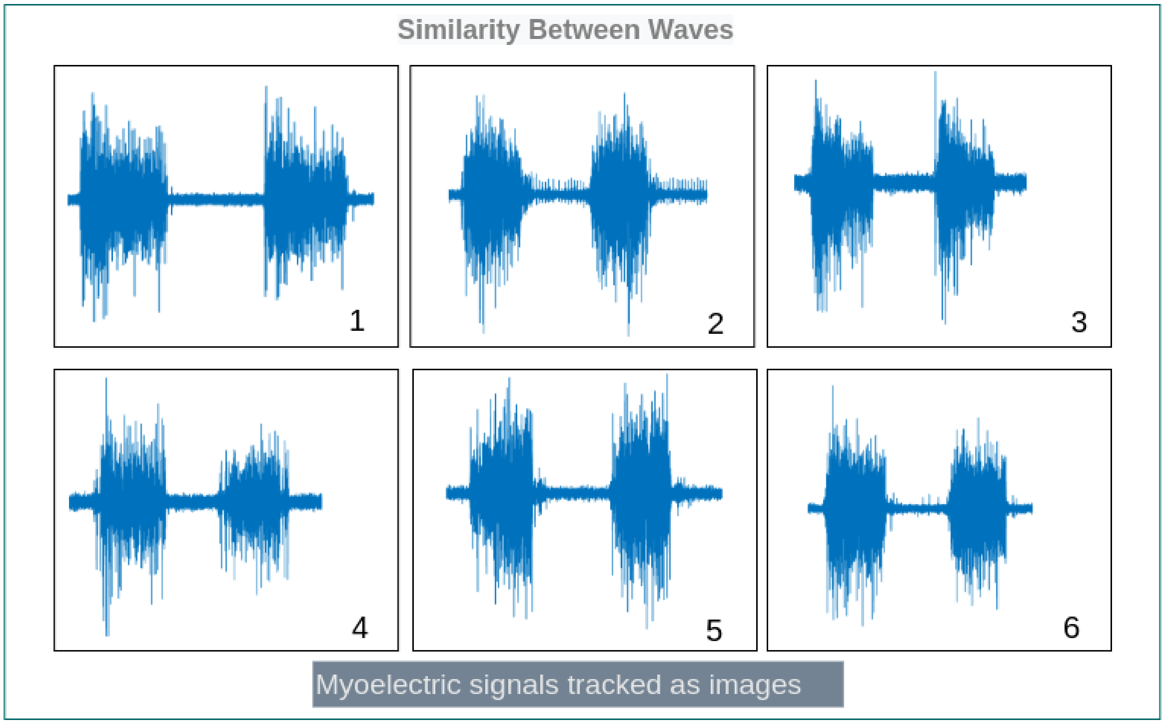

Figure 5.1 – Images representing six distinct movements and their similarity.

At this point, we chose to track the signal in the usual way, that is, linearly extract the features and then assemble a two-dimensional matrix containing only expressive data, such as complexity, magnitude, and activation level muscular. With the feature matrix, it was possible to feed the CNN network and continue with its standard mapping of features followed by the classification, and this resulted in a robust model capable of combining distinction capacity and training time.

Table 5.2 shows the comparison of the proposed model with other approaches that use artificial intelligence to recognize patterns in the myoelectric signal. The method introduced in this research showed better performance rates than all algorithms and low training time. Studies using Deep Learning obtained a high training time (greater than 13 minutes in some cases), which indicates that the hybrid model developed is more feasible for implementation in an embedded real-time system, as it requires fewer resources of hardware and software, thus requiring little battery power.

5.2.1 CNN Network Architecture

Before carrying out the network evaluation, the study tested many arrangements, varying the number of layers, the configuration of the input matrix, size of the filter kernel, type of optimization function, and others. The addition of new convolutional layers increased the training time exponentially. The mini-batch size is directly proportional to precision and

Table 5.2 – Comparison Table - CNN

Individual Movements			
network	**Accuracy (%)**	**Time (s)**	**Study**
CNN-B	**98.68**	2.71	-
CWT	98.31	-	[Cote-Allard et al. 2019]
LDA	92.10	0.15	[Catalan, Brånemark e Håkansson 2013]
MLP	91.20	164.1	[Catalan, Brånemark e Håkansson 2013]
RFN	83.8	0.55	[Catalan, Brånemark e Håkansson 2013]
GLM	94.58	1.15	[Souza, Moreno e Pimenta 2018]
CNN	85.69	$\geq$ 13 minutes	[Laezza 2018]
RNN	91.38	$\geq$ 24 minutes	[Laezza 2018]
Combined Movements			
CNN-B	**97.42**	6.9	-
CWT	68.98	-	[Cote-Allard et al. 2019]
LDA	93.8	0.35	[Prahm et al. 2016]
MLP	94.1	172	[Prahm et al. 2016]
CNN	83.56	$\geq$ 13 minutes	[Laezza 2018]
RNN	87.47	$\geq$ 24 minutes	[Laezza 2018]

time: for low values, training tends to grow, and for high quantities, the ability to distinguish between classes decreases. The learning rate follows a similar rule.

The lower the ratio, the longer it takes to train the algorithm, and higher values cause the algorithm to lose accuracy. Figure 5.2 illustrates the evolution during the network training and its respective confusion matrix. Note that the convergence happens in the initial stages, and this suggests that not so many times may be necessary, which can reduce the time of training.

CNN with pre-processing of features managed to get the best results out of 5 out of 6 data sets. The work demonstrates the superior performance of the deep learning technique when fed by a pre-extracted set of features obtained from EMG signals. The classification performance was better because the information presented as input to the algorithm already had relevant and formatted characteristics, which highlighted the differences between the classes of movements. In addition, there was a reduction in inherent dimensionality. As the input is representative, the filtering process of the algorithm was able to obtain a more objective resolution space.

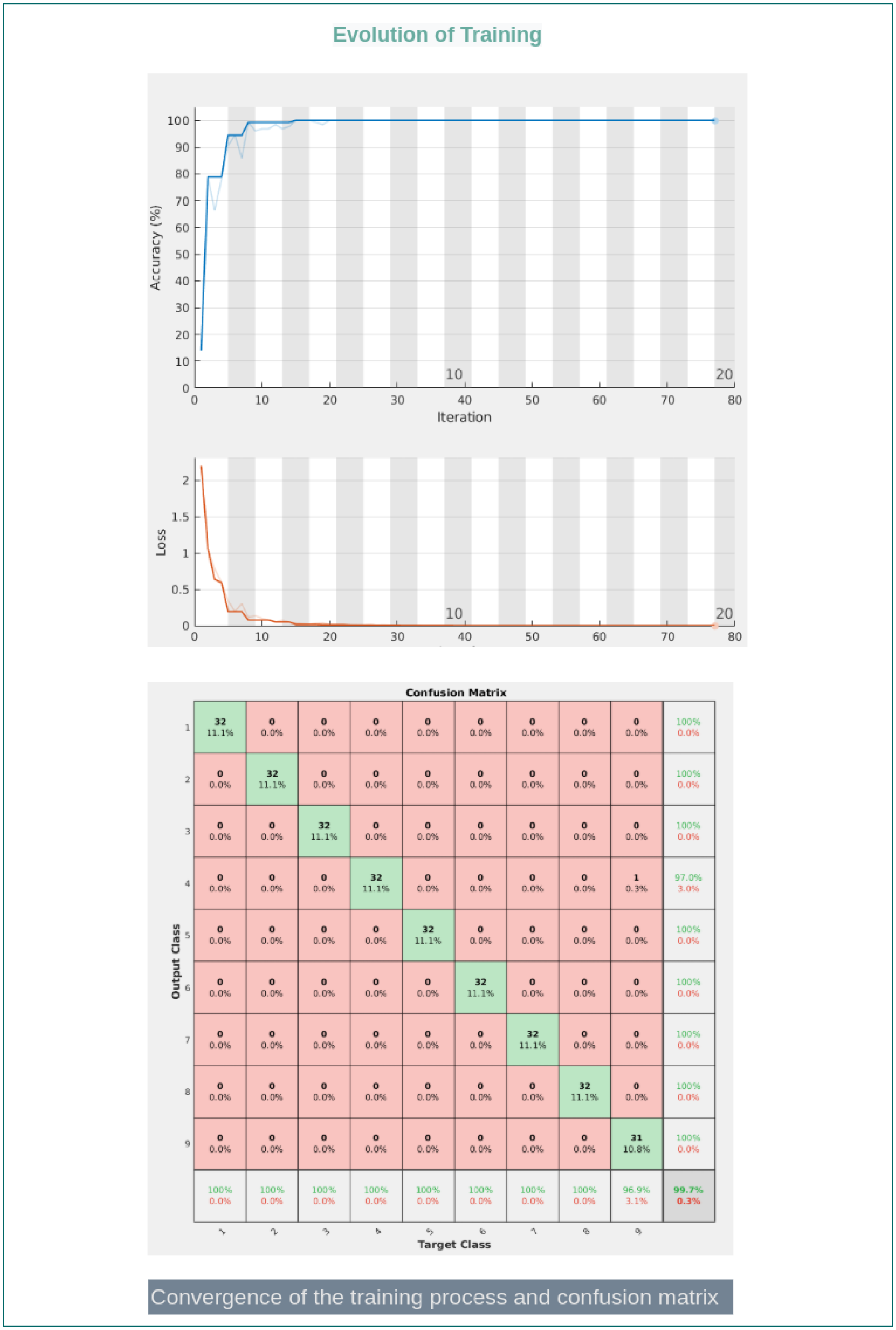

Figure 5.2 – Evolution of training and confusion matrix

5.3 BIOPATREC-PY AND PARALLELIZATION

The BioPatRec-Py software developed by the research guaranteed the execution of the EMG classification system in parallel. Besides, it introduced a new filtering and standardization system. The high-level language used in the program construction has native AI libraries and is directed to devices such as GPU / FPGA. This step allowed the use of DL techniques at a lower computational cost and without degrading the accuracy of movement prediction. The study adopted the most simplistic Nvidia card, and even so, the results were better than a high-end CPU. This result suggests that in the case of the FPGA, the values obtained would be even better since it is also parallelized and has hardware dedicated to the implementation of the deep neural networks that the study adopted.

The principal metric of this stage was the training time, which confirmed the initial proposition: it is possible to embark DL techniques robustly through its parallelization. In general, the comparative analysis between the serial and parallelized approach showed that the second is faster, with values below 1.5 seconds in all cases, against results of up to 30 seconds from its counterpart. In addition, the Kaufman filter and the Quantil standardization employed by the methodology presented a more uniform input to the network, which allowed that there was not much variation between the results achieved by each configuration so that the convergence was homogeneous between the different arrangements. The two techniques enabled the number of experiments to be reduced since there were no relevant changes between the initial results between distinct architectures.

The following comparison can be made, consider the CNN-A that obtained 97.03% of correct answers with a training time of 3.46 and a CNN-GPU with 96.44% accuracy and a time of 0.44s. The parallelized configuration is 7.8 times faster and with no relevant statistical variation concerning its accuracy. If we consider LSTM networks, the differences are even more striking.

Table 5.3 shows the comparison between the introduced concept with other techniques that use DL to recognize patterns in the EMG. The algorithm used in this Thesis achieved better performance values than all the others, and with low training time, [Souza, Moreno e Pimenta 2020]. Yu *et al.* [Hu et al. 2018] adopted a similar model and employed the same set. However, the research did not evaluate the time, and the technique used was the individual, without considering the global training. The article [Du et al. 2017] used a methodology inspired by an adaptive framework, based on DL, to increase the recognition of gestures between sessions. The [Srinivasan et al. 2018] survey used CNN to recognize finger movements. The research [Cote-Allard et al. 2019] classified 18 gestures and used the raw signal to feed the DL technique. The text [Wang et al. 2018] developed a model based on CNN's.

Despite the variability of signals between people, DL networks can create a high-level

Table 5.3 – Comparison Table

Combined Movements			
LSTM-Py	97.44%	0.69s	-
CNN-1D	97.10%	1.12s	-
CWT	68.98%	-	[Cote-Allard et al. 2019]
CNN	83.56%	13 minutes	[Laezza 2018]
CNN	91.61%	100s	[Ameri et al. 2018]
CNN-RNN	94.10%	-	[Hu et al. 2018]
DL-based	82.30%	-	[Du et al. 2017]
CNN	72.50%	-	[Srinivasan et al. 2018]
RCNN	87.3%	-	[Wang et al. 2018]

representation so that the input is mapped abstractly. As the standard deviation was low among individuals, it is possible to say that the characteristics learned by the algorithms were less susceptible to population variation. In addition, in the last stage of the trial, it was possible to train a network with data from all participants and still obtain high accuracy. This result is promising and indicates that it is feasible to develop a robust and comprehensive concept, where the training process is carried out with a collection of signals. This procedure would intuitively allow DL techniques to work under adverse and different conditions, as in the case of everyday use [Souza, Moreno e Pimenta 2020].

Finally, to simulate adverse conditions, the work contaminated the test matrix with random noise and made a progressive analysis of the impact caused by the daily variability of use. It was observed that if the noise is not exaggerated to changing the wave characteristics, the network still obtains excellent results. These observations support the hypothesis that the approach would be robust in an embedded system.

6 CONCLUSION AND SUGGESTIONS FOR FUTURE WORK

Vim, vi, concluí!

Isaac Asimov

This thesis introduced a method for performing pattern recognition in EMG in offline mode, using Deep Learning techniques in consortium with features extraction in the domain of time and frequency. Subsequently, the study paralleled the approach introduced using an Nvidia GPU and adopted its filtering and normalization method. The results will allow Rehabilitation Engineering to increase its literature and, thus, can positively impact the lives of people who have lost a limb or were born with some type of congenital malformation.

The study was done incrementally, wherein summary, it was satisfactory and met all its functional requirements. The first result obtained was the hybrid combination between two DL techniques and BioPatRec, in Matlab software. At this point, the work contributed simultaneously with Rehabilitation Engineering and Features Engineering. In the first case, there was a drastic reduction in training time compared to similar jobs. Tracking the myoelectric signal as a time series, instead of using its raw data in the form of an image, accelerated the convergence process of the CNN and LSTM algorithms. In the second case, the hybrid approach will provide a method capable of allowing accurate classification in systems with hardware limitations, which employ DL techniques. The concept is capable of substantially relieving the load on the network, thus excluding irrelevant geometric information. It can be concluded that the a priori extraction of characteristics is functional in the case of EMG and can be adopted in any model that uses signs and pattern recognition in its construction.

The second increment of the work was the statistical evaluation of the networks. For each algorithm, the study selected an evaluation scenario, which had different arrangements and a search grid to adjust the hyperparameters. The methodology considered several sets of movements, evaluated different perspectives and concepts of Feature Engineering. The results were satisfactory, and the networks achieved high precision and training times in seconds. It is possible to say that the implemented models were able to deliver all the robustness that an offline system requires, both for lower and upper limb movements, before being shipped in a real prosthesis. The operation manual of such classifiers is also another result of this investigation and can be used for another research, considering the conditions previously evaluated. Thus, it is concluded that the DL algorithms are efficient in solving the EMG classification problem and were superior to the others in most of the scenarios set up.

Individually, the LSTM network allows us to conclude some points. Due to the way it manipulates data that have a long-term dependency, the classifier works best on larger data sets, regardless of the number of classes, as the algorithm obtained better results in the set of movements with up to 3-DoF. During the population variance assessment stage, the need to develop more efficient signal extraction techniques became evident. Finally, it's possible to conclude that the LSTM network does not need to be very complex to address the EMG problem and that the adjustment of hyper-parameters is a fundamental step in its configuration.

The researchers who used the CNN network in the case of EMG did not take into account that the signals are symmetrical and adopted the mapping of pixels. This method increased the training time and did not guarantee high accuracy. The research then induces that the features extracted in the domain of time and frequency had information capable of mitigating the stochastic nature of the signal, such as complexity and magnitude, RMS value, peak wave value, and others. In this way, with expressive and compressed data, it was possible to organize the entry in a matrix way, accelerating the filtering and increasing the ability to distinguish between movements. From the comparative study done, it can be said that the concept is robust and allows the use of the technique without concern for the computational resources reported.

The parallel implementation step was, in fact, the most satisfactory, as it guaranteed the practical requirements necessary to embark the solution on a prosthesis, as it delivered the performance metrics crucial to the functioning of the apparatus, without worrying about the natural limitations of such a system. During the development of BioPatRec-Py, it was possible to view the data and, it was noticed that they did not follow a uniform distribution, so the study adopted an efficient standardization method that improved accuracy. Based on a Quantil standardization, the methodology ensured that every characteristic follows a standard distribution. Another development of this research was the adoption of a unique filtering system through the Kaufman Moving Average as the signal was treated as a series instead of a set of pixels, it was possible to adopt a filter that is common in finance and thus attenuating the noise, decreasing the volatility of the series and eliminating several irrelevant points of the wave. This step reduced the uncertainty and immediately impacted the training time. It accelerated the rate of convergence of the objective function of the studied ANNs. The BioPatRec-Py executed by a GPU has satisfactorily met the real-time requirements, since most of the time it was able to perform the training in less than 1 seconds, this ensures that the process of calibrating a prosthesis is not limited to an environment dedicated. This possibility is directly linked to the individual's brain plasticity since, from time to time, the characteristics of the EMG undergo short variations, which need to be addressed to the network. Hypothetically and intuitively, a closed system will benefit from the possibility of executing a new training every time the results start to move away from an acceptable level.

Finally, the differential delivered by the research was its ability to train a generic model, capable of operating under different conditions, considering a heterogeneous set of signals. The global training reached 97.83% accuracy in a data set that represented the entire population. After the training process, it is possible to use the model generated uniformly among the participants without worrying about the need to create an individual protocol. In summary, the application of the techniques introduced by the research allows formulating devices whose operation is more comprehensive. The results suggest the ability of the ANN to identify the global patterns of each movement and not just individual patterns. In addition, the work carried out a study on variability and simulated scenarios where the network operated under adverse conditions. A uniformly distributed noise was progressively added to the test set to confuse the classifier. The results showed that despite the noise, the network obtained satisfactory results.

6.1 FUTURE WORK

Considering a possible future work (an extension or a postdoc), or as a suggestion for other authors, the following topics with potential for exploration can be mentioned:

- Evaluate the network in adverse situations, such as in the case of noise. Missing values, intentional peaks and bottoms, and noise can be added to the test suite. These steps would increase the system's reliability and allow to simulate everyday use situations without the need for a prosthesis.

- Creation of a methodology to test the global approach in loco with new participants and evaluate the results in an embedded device. For this, it would be necessary to acquire new signals and perform the same treatment used in this work. Thus, networks could be trained with a new set of information and tested on the hardware described in the previous item;

- To make the use of the prosthesis more natural it is possible to think of a regression system instead of classification. This way, the signal would not be segmented instead, the prediction would be constant over time, and instead of using 200-millisecond intervals, the network would operate continuously, which would intuitively make the operation more robust. Of course, this approach has several practical limitations, as labeling data continuously would be a challenging task.

- As BioPatRec-Py was written using high-level frameworks aimed at AI and parallelization, the next step of the research (in) would be to acquire a dedicated machine learning FPGA (with specialized hardware) to test the presented software. Thus, it would be possible to generate all the HDL code necessary for the logic circuit. Also, development kits for such devices would allow you to build and test the application before printing the board;

BIBLIOGRAPHY

Abiodun et al. 2018 ABIODUN, O. I. et al. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, v. 4, n. 11, p. e00938, nov. 2018. ISSN 24058440. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S2405844018332067>.

Agarap 2018 AGARAP, A. F. Deep learning using rectified linear units (relu). *ArXiv e-prints*, 03 2018. Disponível em: <https://arxiv.org/pdf/1803.08375.pdf>.

Alaeldin Suliman e Yun Zhang 2015 Alaeldin Suliman; Yun Zhang. A Review on Back-Propagation Neural Networks in the Application of Remote Sensing Image Classification. *Journal of Earth Science and Engineering*, v. 5, n. 1, jan. 2015. ISSN 2159581X, 2159581X. Disponível em: <http://www.davidpublisher.org/index.php/Home/Article/index?id=6142.html>.

Alsmadi, Omar e Noah 2009 ALSMADI, M.; OMAR, K.; NOAH, S. A. M. Back propagation algorithm : The best algorithm among the multi-layer perceptron algorithm. *International Journal of Computer Science and Network Security*, v. 9, p. 378–383, 01 2009.

Altman e Krzywinski 2018 ALTMAN, N.; KRZYWINSKI, M. The curse(s) of dimensionality. *Nature Methods*, v. 15, n. 6, p. 399–400, jun. 2018. ISSN 1548-7091, 1548-7105. Disponível em: <http://www.nature.com/articles/s41592-018-0019-x>.

Ameri et al. 2018 AMERI, A. et al. Real-time, simultaneous myoelectric control using a convolutional neural network. *PLOS ONE*, v. 13, n. 9, p. e0203835, set. 2018. ISSN 1932-6203. Disponível em: <https://dx.plos.org/10.1371/journal.pone.0203835>.

Andrearczyk e Whelan 2017 ANDREARCZYK, V.; WHELAN, P. F. Deep Learning in Texture Analysis and Its Application to Tissue Image Classification. In: *Biomedical Texture Analysis*. Elsevier, 2017. p. 95–129. ISBN 9780128121337. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/B9780128121337000041>.

Andresen 2002 Andresen, S. L. John mccarthy: father of ai. *IEEE Intelligent Systems*, v. 17, n. 5, p. 84–85, 2002.

Arjunan e Kumar 2010 ARJUNAN, S.; KUMAR, D. Decoding subtle forearm flexions using fractal features of surface electromyogram from single and multiple sensors. *Journal of NeuroEngineering and Rehabilitation*, v. 7, n. 1, p. 53, 2010. ISSN 1743-0003. Disponível em: <http://jneuroengrehab.biomedcentral.com/articles/10.1186/1743-0003-7-53>.

Asht e Dass 2012 ASHT, S.; DASS, R. Pattern recognition techniques: A review. *International Journal of Computer Science and Telecommunications*, v. 3, 09 2012.

Atzori et al. 2014 ATZORI, M. et al. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. *Scientific Data*, v. 1, n. 1, p. 140053, dez. 2014. ISSN 2052-4463. Disponível em: <http://www.nature.com/articles/sdata201453>.

Atzori e Muller 2015 ATZORI, M.; MULLER, H. The Ninapro database: A resource for sEMG naturally controlled robotic hand prosthetics. In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*

(EMBC). Milan: IEEE, 2015. p. 7151–7154. ISBN 9781424492718. Disponível em: <http://ieeexplore.ieee.org/document/7320041/>.

Bao et al. 2021 BAO, T. et al. A CNN-LSTM Hybrid Model for Wrist Kinematics Estimation Using Surface Electromyography. *IEEE Transactions on Instrumentation and Measurement*, v. 70, p. 1–9, 2021. ISSN 0018-9456, 1557-9662. Disponível em: <https://ieeexplore.ieee.org/document/9252155/>.

Barik e Sinha 2016 BARIK, R.; SINHA, A. Taxonomy of reconfigurable computing and operating system. In: . [S.l.: s.n.], 2016.

Basart, Farrus e Serra 2019 Basart, J. M.; Farrus, M.; Serra, M. Mindful engineers in sustainable engineering. *IEEE Technology and Society Magazine*, v. 38, n. 3, p. 68–73, 2019.

Bhamare e Suryawanshi 2018 BHAMARE, D.; SURYAWANSHI, P. Review on Reliable Pattern Recognition with Machine Learning Techniques. *Fuzzy Information and Engineering*, v. 10, n. 3, p. 362–377, jul. 2018. ISSN 1616-8658, 1616-8666. Disponível em: <https://www.tandfonline.com/doi/full/10.1080/16168658.2019.1611030>.

Blessie e Karthikeyan 2012 BLESSIE, E. C.; KARTHIKEYAN, E. Sigmis: A Feature Selection Algorithm Using Correlation Based Method. *Journal of Algorithms & Computational Technology*, v. 6, n. 3, p. 385–394, set. 2012. ISSN 1748-3018, 1748-3026. Disponível em: <http://journals.sagepub.com/doi/10.1260/1748-3018.6.3.385>.

Boccia et al. 2016 BOCCIA, G. et al. Muscle fiber conduction velocity and fractal dimension of EMG during fatiguing contraction of young and elderly active men. *Physiological Measurement*, v. 37, n. 1, p. 162–174, jan. 2016. ISSN 0967-3334, 1361-6579. Disponível em: <https://iopscience.iop.org/article/10.1088/0967-3334/37/1/162>.

Borges 2014 BORGES, A. *Diferença entre placa motora e junção neuromuscular*. 2014. Https://questoesdefisiocomentadas.wordpress.com/tag/diferenca-entre-placa-motora-e-juncao-neuromuscular/.

Cai et al. 2018 CAI, W. et al. Network linear discriminant analysis. *Computational Statistics & Data Analysis*, v. 117, p. 32–44, jan. 2018. ISSN 01679473. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S016794731730155X>.

Campbell, Phinyomark e Scheme 2019 CAMPBELL, E.; PHINYOMARK, A.; SCHEME, E. Linear Discriminant Analysis with Bayesian Risk Parameters for Myoelectric Control. In: *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. Ottawa, ON, Canada: IEEE, 2019. p. 1–5. ISBN 9781728127231. Disponível em: <https://ieeexplore.ieee.org/document/8969237/>.

Carney e Cunningham 1998 CARNEY, J. G.; CUNNINGHAM, P. *The Epoch Interpretation of Learning*. 1998. Disponível em: <https://pdfs.semanticscholar.org/e512/0cf22bda8e5f3a8b4456e7422903122b15dc.pdf>.

Carrizosa, Martín-Barragán e Morales 2011 CARRIZOSA, E.; MARTíN-BARRAGáN, B.; MORALES, D. R. Detecting relevant variables and interactions in supervised classification. *European Journal of Operational Research*, v. 213, n. 1, p. 260–269, ago. 2011. ISSN 03772217. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0377221710002195>.

Carvalho 2017 CARVALHO, A. C. P. de Leon Ferreira de. *Redes Neurais Artificias*. 2017. Http://conteudo.icmc.usp.br/pessoas/andre/research/neural/.

Catalan, Brånemark e Håkansson 2013 CATALAN, M. O.; BRåNEMARK, R.; HåKANSSON, B. BioPatRec: A modular research platform for the control of artificial limbs based on pattern recognition algorithms. *Source Code for Biology and Medicine*, v. 8, n. 1, p. 11, dez. 2013. ISSN 1751-0473. Disponível em: <https://scfbm.biomedcentral.com/articles/10.1186/1751-0473-8-11>.

Chadwell et al. 2020 CHADWELL, A. et al. Technology for monitoring everyday prosthesis use: a systematic review. *Journal of NeuroEngineering and Rehabilitation*, v. 17, n. 1, p. 93, dez. 2020. ISSN 1743-0003. Disponível em: <https://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-020-00711-4>.

Chaudhary, Bhatia e Ahlawat 2014 CHAUDHARY, V.; BHATIA, R.; AHLAWAT, A. K. A novel Self-Organizing Map (SOM) learning algorithm with nearest and farthest neurons. *Alexandria Engineering Journal*, v. 53, n. 4, p. 827–831, dez. 2014. ISSN 11100168. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1110016814000970>.

Cortes e Vapnik 1995 CORTES, C.; VAPNIK, V. Support-vector networks. *Machine Learning*, v. 20, n. 3, p. 273–297, set. 1995. ISSN 0885-6125, 1573-0565. Disponível em: <http://link.springer.com/10.1007/BF00994018>.

Cote-Allard et al. 2019 COTE-ALLARD, U. et al. Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 27, n. 4, p. 760–771, abr. 2019. ISSN 1534-4320, 1558-0210. Disponível em: <https://ieeexplore.ieee.org/document/8630679/>.

Cui e Ji 2009 CUI, Z.-f.; JI, X.-h. Feature selection based on linear discriminant analysis. *Journal of Computer Applications*, v. 29, p. 2781–2785, 12 2009.

Dominik, Andreas e Behnke 2010 DOMINIK, S.; ANDREAS, M.; BEHNKE, S. Evaluation of pooling operations in convolutional architectures for object recognition. In: *Artificial Neural Networks – ICANN 2010*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. v. 6354, p. 92–101. ISBN 9783642158247 9783642158254. Disponível em: <http://link.springer.com/10.1007/978-3-642-15825-4_10>.

Dormann et al. 2013 DORMANN, C. F. et al. Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. *Ecography*, v. 36, n. 1, p. 27–46, jan. 2013. ISSN 09067590. Disponível em: <http://doi.wiley.com/10.1111/j.1600-0587.2012.07348.x>.

Du et al. 2017 DU, Y. et al. Surface EMG-Based Inter-Session Gesture Recognition Enhanced by Deep Domain Adaptation. *Sensors*, v. 17, n. 3, p. 458, fev. 2017. ISSN 1424-8220. Disponível em: <http://www.mdpi.com/1424-8220/17/3/458>.

Emmert-Streib et al. 2020 EMMERT-STREIB, F. et al. An introductory review of deep learning for prediction models with big data. *Frontiers in Artificial Intelligence*, v. 3, p. 4, fev. 2020. ISSN 2624-8212. Disponível em: <https://www.frontiersin.org/article/10.3389/frai.2020.00004/full>.

Escalante 2005 ESCALANTE, H. J. A comparison of outlier detection algorithms for machine learning. *Programming and Computer Software*, 01 2005.

Esteva et al. 2019 ESTEVA, A. et al. A guide to deep learning in healthcare. *Nature Medicine*, v. 25, n. 1, p. 24–29, jan. 2019. ISSN 1078-8956, 1546-170X. Disponível em: <http://www.nature.com/articles/s41591-018-0316-z>.

Fabris, Magalhães e Freitas 2017 FABRIS, F.; MAGALHãES, J. P. d.; FREITAS, A. A. A review of supervised machine learning applied to ageing research. *Biogerontology*, v. 18, n. 2, p. 171–188, abr. 2017. ISSN 1389-5729, 1573-6768. Disponível em: <http://link.springer.com/10.1007/s10522-017-9683-y>.

Farina e Aszmann 2014 FARINA, D.; ASZMANN, O. Bionic limbs: clinical reality and academic promises. *Science Translational Medicine*, v. 6, n. 257, p. 257ps12–257ps12, out. 2014. ISSN 1946-6234, 1946-6242. Disponível em: <https://stm.sciencemag.org/lookup/doi/10.1126/scitranslmed.3010453>.

Farina et al. 2010 FARINA, D. et al. Decoding the neural drive to muscles from the surface electromyogram. Clinical Neurophysiology, Elsevier - Amsterdam, 2010.

Farrell e Weir 2007 FARRELL, T. R.; WEIR, R. F. The Optimal Controller Delay for Myoelectric Prostheses. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 15, n. 1, p. 111–118, mar. 2007. ISSN 1534-4320, 1558-0210. Disponível em: <https://ieeexplore.ieee.org/document/4126535/>.

Feng et al. 2014 FENG, C. et al. Log-transformation and its implications for data analysis. *Shanghai archives of psychiatry*, v. 26, p. 105–9, 04 2014.

Garavito et al. 2016 GARAVITO, F. et al. EMG signal analysis based on fractal dimension for muscle activation detection under exercice protocol. In: *2016 XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA)*. Bucaramanga, Colombia: IEEE, 2016. p. 1–5. ISBN 9781509037971. Disponível em: <http://ieeexplore.ieee.org/document/7743365/>.

Gers, Schmidhuber e Cummins 2000 GERS, F. A.; SCHMIDHUBER, J.; CUMMINS, F. Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, v. 12, n. 10, p. 2451–2471, out. 2000. ISSN 0899-7667, 1530-888X. Disponível em: <http://www.mitpressjournals.org/doi/10.1162/089976600300015015>.

Ghallab, Fahmy e Nasr 2019 GHALLAB, H.; FAHMY, H.; NASR, M. Detection outliers on internet of things using big data technology. *Egyptian Informatics Journal*, p. S1110866519301616, dez. 2019. ISSN 11108665. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1110866519301616>.

Ghasemian, Hosseinmardi e Clauset 2019 GHASEMIAN, A.; HOSSEINMARDI, H.; CLAUSET, A. Evaluating Overfit and Underfit in Models of Network Community Structure. *IEEE Transactions on Knowledge and Data Engineering*, p. 1–1, 2019. ISSN 1041-4347, 1558-2191, 2326-3865. Disponível em: <https://ieeexplore.ieee.org/document/8692626/>.

Ghojogh et al. 2019 GHOJOGH, B. et al. Feature Selection and Feature Extraction in Pattern Analysis: A Literature Review. *arXiv:1905.02845 [cs, stat]*, maio 2019. ArXiv: 1905.02845. Disponível em: <http://arxiv.org/abs/1905.02845>.

Greff et al. 2017 GREFF, K. et al. LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, v. 28, n. 10, p. 2222–2232, out. 2017. ISSN 2162-237X, 2162-2388. Disponível em: <http://ieeexplore.ieee.org/document/7508408/>.

Guresen e Kayakutlu 2011 GURESEN, E.; KAYAKUTLU, G. Definition of artificial neural networks with comparison to other networks. *Procedia Computer Science*, v. 3, p. 426–433, 2011. ISSN 18770509. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1877050910004461>.

Hassabis et al. 2017 HASSABIS, D. et al. Neuroscience-inspired artificial intelligence. *Neuron*, v. 95, n. 2, p. 245–258, jul. 2017. ISSN 08966273. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0896627317305093>.

He et al. 2018 HE, Y. et al. Surface EMG Pattern Recognition Using Long Short-Term Memory Combined with Multilayer Perceptron. In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Honolulu, HI: IEEE, 2018. p. 5636–5639. ISBN 9781538636466. Disponível em: <https://ieeexplore.ieee.org/document/8513595/>.

Hengstler et al. 2016 HENGSTLER, M. et al. Applied artificial intelligence and trust—The case of autonomous vehicles and medical assistance devices. *Technological Forecasting and Social Change*, v. 105, p. 105–120, abr. 2016. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0040162515004187>.

Hira e Gillies 2015 HIRA, Z. M.; GILLIES, D. F. A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data. *Advances in Bioinformatics*, v. 2015, p. 1–13, 2015. ISSN 1687-8027, 1687-8035. Disponível em: <http://www.hindawi.com/journals/abi/2015/198363/>.

Hochreiter 1998 HOCHREITER, S. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, v. 06, n. 02, p. 107–116, abr. 1998. ISSN 0218-4885, 1793-6411. Disponível em: <https://www.worldscientific.com/doi/abs/10.1142/S0218488598000094>.

Hochreiter e Schmidhuber 1997 HOCHREITER, S.; SCHMIDHUBER, J. Long Short-Term Memory. *Neural Computation*, v. 9, n. 8, p. 1735–1780, nov. 1997. ISSN 0899-7667, 1530-888X. Disponível em: <http://www.mitpressjournals.org/doi/10.1162/neco.1997.9.8.1735>.

Howard 2016 HOWARD, R. M. *The application of data analysis methods for surface electromyography in shot putting and sprinting.* https://ulir.ul.ie/bitstream/handle/10344/5584/Howard$_2$016$_a$$pplication.pdf?sequence = 6 : [s.n.], 2016$

Hu et al. 2018 HU, Y. et al. A novel attention-based hybrid CNN-RNN architecture for sEMG-based gesture recognition. *PLOS ONE*, v. 13, n. 10, p. e0206049, out. 2018. ISSN 1932-6203. Disponível em: <https://dx.plos.org/10.1371/journal.pone.0206049>.

Hudgins, Parker e Scott 1993 HUDGINS, B.; PARKER, P.; SCOTT, R. A new strategy for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering*, v. 40, n. 1, p. 82–94, jan. 1993. ISSN 00189294. Disponível em: <http://ieeexplore.ieee.org/document/204774/>.

Ibrahim et al. 2018 IBRAHIM, N. et al. Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy. *Pertanika Journal of Science and Technology*, v. 26, p. 329–340, 01 2018.

Ide e Kurita 2017 IDE, H.; KURITA, T. Improvement of learning for CNN with ReLU activation by sparse regularization. In: *2017 International Joint Conference on Neural Networks (IJCNN)*. Anchorage, AK, USA: IEEE, 2017. p. 2684–2691. ISBN 9781509061822. Disponível em: <http://ieeexplore.ieee.org/document/7966185/>.

Igual et al. 2019 IGUAL, C. et al. Myoelectric Control for Upper Limb Prostheses. *Electronics*, v. 8, n. 11, p. 1244, out. 2019. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/8/11/1244>.

Indolia et al. 2018 INDOLIA, S. et al. Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. *Procedia Computer Science*, v. 132, p. 679–688, 2018. ISSN 18770509. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1877050918308019>.

Indolia et al. 2018 INDOLIA, S. et al. Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. *Procedia Computer Science*, v. 132, p. 679–688, 2018. ISSN 18770509. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1877050918308019>.

Iqbal, Subramaniam e Shaniba 2018 IQBAL, N. V.; SUBRAMANIAM, K.; SHANIBA, A. A Review on Upper-Limb Myoelectric Prosthetic Control. *IETE Journal of Research*, v. 64, n. 6, p. 740–752, nov. 2018. ISSN 0377-2063, 0974-780X. Disponível em: <https://www.tandfonline.com/doi/full/10.1080/03772063.2017.1381047>.

Ishola, Nawawi e Abdullah 2015 ISHOLA, K. S.; NAWAWI, M. N. M.; ABDULLAH, K. Combining Multiple Electrode Arrays for Two-Dimensional Electrical Resistivity Imaging Using the Unsupervised Classification Technique. *Pure and Applied Geophysics*, v. 172, n. 6, p. 1615–1642, jun. 2015. ISSN 0033-4553, 1420-9136. Disponível em: <http://link.springer.com/10.1007/s00024-014-1007-4>.

Ison et al. 2016 ISON, M. et al. High-Density Electromyography and Motor Skill Learning for Robust Long-Term Control of a 7-DoF Robot Arm. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 24, n. 4, p. 424–433, abr. 2016. ISSN 1534-4320, 1558-0210. Disponível em: <https://ieeexplore.ieee.org/document/7073629/>.

Jabbar e Khan 2014 JABBAR, H. K.; KHAN, R. Z. Methods to Avoid Over-Fitting and Under-Fitting in Supervised Machine Learning (Comparative Study). In: *Computer Science, Communication and Instrumentation Devices*. Research Publishing Services, 2014. p. 163–172. ISBN 9789810952471. Disponível em: <http://rpsonline.com.sg/proceedings/9789810952471/html/017.xml>.

Jia et al. 2019 JIA, W. et al. Hyperparameter optimization for machine learning models based on bayesian optimizationb. *Journal of Electronic Science and Technology*, 2019.

Kandel e Castelli 2020 KANDEL, I.; CASTELLI, M. The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express*, p. S2405959519303455, maio 2020. ISSN 24059595. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S2405959519303455>.

Kim, Schug e Kim 2015 KIM, Y.; SCHUG, K. A.; KIM, S. B. An ensemble regularization method for feature selection in mass spectral fingerprints. *Chemometrics and Intelligent Laboratory Systems*, v. 146, p. 322–328, ago. 2015. ISSN 01697439. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0169743915001227>.

Kohonen 1982 KOHONEN, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, v. 43, n. 1, p. 59–69, 1982. ISSN 0340-1200, 1432-0770. Disponível em: <http://link.springer.com/10.1007/BF00337288>.

Kohonen 2013 KOHONEN, T. Essentials of the self-organizing map. *Neural Networks*, v. 37, p. 52–65, jan. 2013. ISSN 08936080. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0893608012002596>.

Krueger-Beck et al. 2011 KRUEGER-BECK, E. et al. Potencial de ação: do estímulo à adaptação neural. *Fisioterapia em Movimento*, v. 24, n. 3, p. 535–547, set. 2011. ISSN 0103-5150. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-51502011000300018&lng=pt&tlng=pt>.

Kulkarni et al. 2020 KULKARNI, S. et al. Artificial Intelligence in Medicine: Where Are We Now? *Academic Radiology*, v. 27, n. 1, p. 62–70, jan. 2020. ISSN 10766332. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1076633219304581>.

Kyranou, Vijayakumar e Erden 2018 KYRANOU, I.; VIJAYAKUMAR, S.; ERDEN, M. S. Causes of Performance Degradation in Non-invasive Electromyographic Pattern Recognition in Upper Limb Prostheses. *Frontiers in Neurorobotics*, v. 12, p. 58, set. 2018. ISSN 1662-5218. Disponível em: <https://www.frontiersin.org/article/10.3389/fnbot.2018.00058/full>.

Lacey et al. 2016 LACEY, G. et al. Deep Learning on FPGAs: Past, Present, and Future. *arXiv:1602.04283 [cs, stat]*, fev. 2016. ArXiv: 1602.04283. Disponível em: <http://arxiv.org/abs/1602.04283>.

Laezza 2018 LAEZZA, R. *Deep Neural Networks for Myoelectric Pattern Recognition*. Tese (Doutorado) — CHALMERS UNIVERSITY OF TECHNOLOGY, 2018. Disponível em: <http://publications.lib.chalmers.se/records/fulltext/254980/254980.pdf>.

LeCun et al. 1999 LECUN, Y. et al. Object Recognition with Gradient-Based Learning. In: *Shape, Contour and Grouping in Computer Vision*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999. v. 1681, p. 319–345. ISBN 9783540667223 9783540468059. Disponível em: <http://link.springer.com/10.1007/3-540-46805-6_19>.

Li et al. 2018 LI, C. et al. LightenNet: A Convolutional Neural Network for weakly illuminated image enhancement. *Pattern Recognition Letters*, v. 104, p. 15–22, mar. 2018. ISSN 01678655. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0167865518300163>.

Li et al. 2018 LI, C. et al. PCA and deep learning based myoelectric grasping control of a prosthetic hand. *BioMedical Engineering OnLine*, v. 17, n. 1, p. 107, dez. 2018. ISSN 1475-925X. Disponível em: <https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/s12938-018-0539-8>.

Li-Chiu Chang et al. 2012 Li-Chiu Chang et al. Reinforced Two-Step-Ahead Weight Adjustment Technique for Online Training of Recurrent Neural Networks. *IEEE Transactions* *on Neural Networks and Learning Systems*, v. 23, n. 8, p. 1269–1278, ago. 2012. ISSN 2162-237X, 2162-2388. Disponível em: <http://ieeexplore.ieee.org/document/6218199/>.

Li et al. 2015 LI, J. et al. LSTM time and frequency recurrence for automatic speech recognition. In: *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. Scottsdale, AZ, USA: IEEE, 2015. p. 187–191. ISBN 9781479972913. Disponível em: <http://ieeexplore.ieee.org/document/7404793/>.

Li et al. 2016 LI, X. et al. Deep learning and its parallelization. In: *Big Data*. Elsevier, 2016. p. 95–118. ISBN 9780128053942. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/B9780128053942000040>.

Liashchynskyi e Liashchynskyi 2019 LIASHCHYNSKYI, P.; LIASHCHYNSKYI, P. Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS. *arXiv:1912.06059 [cs, stat]*, dez. 2019. ArXiv: 1912.06059. Disponível em: <http://arxiv.org/abs/1912.06059>.

Lin et al. 2019 LIN, Z. et al. Benchmarking Deep Learning Frameworks and Investigating FPGA Deployment for Traffic Sign Classification and Detection. *IEEE Transactions on Intelligent Vehicles*, v. 4, n. 3, p. 385–395, set. 2019. ISSN 2379-8904, 2379-8858. Disponível em: <https://ieeexplore.ieee.org/document/8723505/>.

Lipton, Berkowitz e Elkan 2019 LIPTON, Z. C.; BERKOWITZ, J.; ELKAN, C. A critical review of recurrent neural networks for sequence learning. *arXiv*, p. 1235–1270, jul. 2019. Disponível em: <https://arxiv.org/pdf/1506.00019.pdf>.

Liu et al. 2015 LIU, T. et al. Implementation of training convolutional neural networks. *arXiv*, 06 2015. Disponível em: <https://arxiv.org/pdf/1506.01195.pdf>.

Liu 2018 LIU, Y. H. Feature Extraction and Image Recognition with Convolutional Neural Networks. *Journal of Physics: Conference Series*, v. 1087, p. 062032, set. 2018. ISSN 1742-6588, 1742-6596. Disponível em: <https://iopscience.iop.org/article/10.1088/1742-6596/1087/6/062032>.

Luca et al. 1982 LUCA, C. J. de et al. Behaviour of human motor units in different muscles during linearly varying contractions. The Journal of Physiology, Malden MA - USA, 1982.

Madiajagan e Raj 2019 MADIAJAGAN, M.; RAJ, S. S. Parallel machine learning and deep learning approaches for bioinformatics. In: *Deep Learning and Parallel Computing Environment for Bioengineering Systems*. Elsevier, 2019. p. 245–255. ISBN 9780128167182. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/B9780128167182000221>.

Maesschalck, Jouan-Rimbaud e Massart 2000 MAESSCHALCK, R. D.; JOUAN-RIMBAUD, D.; MASSART, D. The Mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, v. 50, n. 1, p. 1–18, jan. 2000. ISSN 01697439. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0169743999000477>.

Marius et al. 2009 MARIUS, P. et al. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, v. 8, 07 2009.

McCulloch e Pitts 1943 MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of mathematical biophysics*, v. 5, n. 4, p. 115–133, 1943. Disponível em: <https://doi.org/10.1007/BF02478259>.

McMullen et al. 2014 MCMULLEN, D. P. et al. Demonstration of a semi-autonomous hybrid brain–machine interface using human intracranial eeg, eye tracking, and computer vision to control a robotic upper limb prosthetic. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 22, n. 4, p. 784–796, 2014.

Medus et al. 2019 MEDUS, L. D. et al. A Novel Systolic Parallel Hardware Architecture for the FPGA Acceleration of Feedforward Neural Networks. *IEEE Access*, v. 7, p. 76084–76103, 2019. ISSN 2169-3536. Disponível em: <https://ieeexplore.ieee.org/document/8731886/>.

Mereu et al. 2021 MEREU, F. et al. Control Strategies and Performance Assessment of Upper-Limb TMR Prostheses: A Review. *Sensors*, v. 21, n. 6, p. 1953, mar. 2021. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/21/6/1953>.

Merletti e Knaflitz 1992 MERLETTI, R.; KNAFLITZ, M. Electrically evoked myoelectric signals. Critical Reviews in Biomedical Engineering, Boston, MA, 1992.

Miljkovic 2017 MILJKOVIC, D. Brief review of self-organizing maps. In: *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. Opatija, Croatia: IEEE, 2017. p. 1061–1066. ISBN 9789532330908. Disponível em: <http://ieeexplore.ieee.org/document/7973581/>.

Mishra et al. 2017 MISHRA, S. et al. Principal Component Analysis. *International Journal of Livestock Research*, p. 1, 2017. ISSN 2277-1964. Disponível em: <http://www.ejmanager.com/fulltextpdf.php?mno=261590>.

Morais 2010 MORAIS, E. C. *Reconhecimento de Padrões e Redes Neurais Artificiais em Predicão de Estruturas Secundárias de Proteínas*. Dissertação (Mestrado) — COPPE/UFRJ, http://www.cos.ufrj.br/uploadfile/1277729485.pdf, 3 2010.

Morchid et al. 2014 MORCHID, M. et al. Feature selection using Principal Component Analysis for massive retweet detection. *Pattern Recognition Letters*, v. 49, p. 33–39, nov. 2014. ISSN 01678655. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0167865514001767>.

Müller 2012 MüLLER, M. Generalized Linear Models. In: GENTLE, J. E.; HäRDLE, W. K.; MORI, Y. (Ed.). *Handbook of Computational Statistics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 681–709. ISBN 9783642215506 9783642215513. Disponível em: <http://link.springer.com/10.1007/978-3-642-21551-3_24>.

Nelder e Wedderburn 1972 NELDER, J. A.; WEDDERBURN, R. W. M. Generalized Linear Models. *Journal of the Royal Statistical Society. Series A (General)*, v. 135, n. 3, p. 370, 1972. ISSN 00359238. Disponível em: <https://www.jstor.org/stable/10.2307/2344614?origin=crossref>.

Nielsen et al. 2011 NIELSEN, J. L. G. et al. Simultaneous and Proportional Force Estimation for Multifunction Myoelectric Prostheses Using Mirrored Bilateral Training. *IEEE Transactions on Biomedical Engineering*, v. 58, n. 3, p. 681–688, mar. 2011. ISSN 0018-9294, 1558-2531. Disponível em: <http://ieeexplore.ieee.org/document/5551179/>.

Nurvitadhi et al. 2017 NURVITADHI, E. et al. Can FPGAs Beat GPUs in Accelerating Next-Generation Deep Neural Networks? In: *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA '17*. Monterey, California, USA: ACM Press, 2017. p. 5–14. ISBN 9781450343541. Disponível em: <http://dl.acm.org/citation.cfm?doid=3020078.3021740>.

Nwankpa et al. 2018 NWANKPA, C. et al. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv*, 11 2018. Disponível em: <https://arxiv.org/pdf/1811.03378.pdf>.

Ortiz-Catalan 2015 ORTIZ-CATALAN, M. Cardinality as a highly descriptive feature in myoelectric pattern recognition for decoding motor volition. *Frontiers in Neuroscience*, v. 9, out. 2015. ISSN 1662-453X. Disponível em: <http://journal.frontiersin.org/Article/10.3389/fnins.2015.00416/abstract>.

Ortolan 2002 ORTOLAN, R. L. *Estudo E Avaliacão de Técnicas de processamento do Sinal Mioelétrico para controle de sistemas de reabiliatacão*. Dissertação (Mestrado) — Universidade de São Paulo, http://www.teses.usp.br/teses/disponiveis/18/18133/tde-19112002-153337/pt-br.php, 4 2002.

O'Shea e Nash 2015 O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. *ArXiv e-prints*, 11 2015. Disponível em: <https://www.researchgate.net/publication/285164623_An_Introduction_to_Convolutional_Neural_Networks>.

Oskoei e Hu 2007 OSKOEI, M. A.; HU, H. Myoelectric control systems—A survey. *Biomedical Signal Processing and Control*, v. 2, n. 4, p. 275–294, out. 2007. ISSN 17468094. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1746809407000547>.

Parsaei e Stashuk 2011 PARSAEI, H.; STASHUK, D. W. An svm classifier for detecting merged motor unit potential trains extracted by emg signal decomposition using their mup shape information. In: *2011 24th Canadian Conference on Electrical and Computer Engineering(CCECE)*. [S.l.: s.n.], 2011. p. 000795–000798.

Patel 2016 PATEL, K. A review on feature extraction methods. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, v. 27, 2016. ISSN 2320 – 3765.

Paulheim e Meusel 2015 PAULHEIM, H.; MEUSEL, R. A decomposition of the outlier detection problem into a set of supervised learning problems. *Machine Learning*, v. 100, n. 2-3, p. 509–531, set. 2015. ISSN 0885-6125, 1573-0565. Disponível em: <http://link.springer.com/10.1007/s10994-015-5507-y>.

Phinyomark, Khushaba e Scheme 2018 PHINYOMARK, A.; KHUSHABA, R. N.; SCHEME, E. Feature Extraction and Selection for Myoelectric Control Based on Wearable EMG Sensors. *Sensors*, v. 18, n. 5, p. 1615, maio 2018. ISSN 1424-8220. Disponível em: <http://www.mdpi.com/1424-8220/18/5/1615>.

Phinyomark, Khushaba e Scheme 2018 PHINYOMARK, A.; KHUSHABA, R. N.; SCHEME, E. Feature Extraction and Selection for Myoelectric Control Based on Wearable EMG Sensors. *Sensors*, v. 18, n. 5, p. 1615, maio 2018. ISSN 1424-8220. Disponível em: <http://www.mdpi.com/1424-8220/18/5/1615>.

Piccolino 1998 PICCOLINO, M. Animal electricity and the birth of electrophysiology: the legacy of luigi galvani. Brain Research Bulletin, Elsevier - Amsterdam, 1998.

Poel e Kroes 2014 POEL, I. van de; KROES, P. *The Moral Status of Technical Artefacts*. Can technology embody values? Springer, Dordrecht, 2014. v. 17. ISBN 978-94-007-7914-3. Disponível em: <https://doi.org/10.1007/978-94-007-7914-3_7>.

Prahm et al. 2016 PRAHM, C. et al. Combining two open source tools for neural computation (BioPatRec and Netlab) improves movement classification for prosthetic control. *BMC Research Notes*, v. 9, n. 1, p. 429, dez. 2016. ISSN 1756-0500. Disponível em: <http://bmcresnotes.biomedcentral.com/articles/10.1186/s13104-016-2232-y>.

Raheema, Hussein e Al-Khazzar 2020 RAHEEMA, M.; HUSSEIN, J.; AL-KHAZZAR, A. Design of an intelligent controller for myoelectric prostheses based on multilayer perceptron neural network. *IOP Conference Series: Materials Science and Engineering*, v. 671, p. 012064, 01 2020.

Rakitianskaia e Engelbrecht 2015 RAKITIANSKAIA, A.; ENGELBRECHT, A. Measuring Saturation in Neural Networks. In: *2015 IEEE Symposium Series on Computational Intelligence*. Cape Town, South Africa: IEEE, 2015. p. 1423–1430. ISBN 9781479975600. Disponível em: <http://ieeexplore.ieee.org/document/7376778/>.

Rawat e Khemchandani 2017 RAWAT, T.; KHEMCHANDANI, V. Feature Engineering (FE) Tools and Techniques for Better Classification Performance. *International Journal of Innovations in Engineering and Technology*, v. 8, n. 2, 2017. ISSN 23191058. Disponível em: <http://ijiet.com/wp-content/uploads/2017/05/24.pdf>.

Rumelhart, Hinton e Williams 1986 RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, v. 323, n. 6088, p. 533–536, out. 1986. ISSN 0028-0836, 1476-4687. Disponível em: <http://www.nature.com/articles/323533a0>.

Saxena et al. 2017 SAXENA, A. et al. A review of clustering techniques and developments. *Neurocomputing*, v. 267, p. 664–681, dez. 2017. ISSN 09252312. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0925231217311815>.

Senthilkumar e Metilda 2016 SENTHILKUMAR, A.; METILDA, M. A Survey on Cluster Based Outlier Detection Techniques in Data Stream. *International Journal of Data Mining Techniques and Applications*, v. 5, n. 1, p. 96–101, jun. 2016. ISSN 22782419. Disponível em: <http://www.ijdmta.com/abstract_temp.php?id=V5-I1-P23>.

Shashoa et al. 2016 SHASHOA, N. A. A. et al. Classification depend on linear discriminant analysis using desired outputs. In: *2016 17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*. Sousse, Tunisia: IEEE, 2016. p. 328–332. ISBN 9781509034079. Disponível em: <http://ieeexplore.ieee.org/document/7952041/>.

Shawahna et al. 2019 SHAWAHNA, A. et al. FPGA-Based Accelerators of Deep Learning Networks for Learning and Classification: A Review. *IEEE Access*, v. 7, p. 7823–7859, 2019. ISSN 2169-3536. Disponível em: <https://ieeexplore.ieee.org/document/8594633/>.

She et al. 2010 SHE, Q. et al. Multiple kernel learning svm-based emg pattern classification for lower limb control. In: *2010 11th International Conference on Control Automation Robotics Vision*. [S.l.: s.n.], 2010. p. 2109–2113.

Singh 2020 SINGH, A. Demystifying the mathematics behind convolutional neural networks. Analytics Vidhya, 2020. Disponível em: <https://www.analyticsvidhya.com/blog/2020/02/mathematics-behind-convolutional-neural-network/>.

Singh e Singh 2019 SINGH, D.; SINGH, B. Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, p. 105524, maio 2019. ISSN 15684946. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1568494619302947>.

Song et al. 2020 SONG, J. et al. Effects of different feature parameters of semg on human motion pattern recognition using multilayer perceptrons and lstm neural networks. *Applied Sciences*, v. 10, n. 10, 2020. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/10/10/3358>.

Souza e Moreno 2018 SOUZA, G. C.; MORENO, R. L. Netlab MLP - Performance Evaluation for Pattern Recognition in Myoletric Signal. *Procedia Computer Science*, v. 130, p. 932–938, 2018. ISSN 18770509. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S187705091830454X>.

Souza, Pimenta e Moreno 2020 SOUZA, G. C.; PIMENTA, T.; MORENO, R. From AI and Electromyography to Financial Market: A Philosophical Perspective. *IOSR Journal of Humanities and Social Science*, v. 25, n. 12, p. 9, dez. 2020. ISSN 2279-0837. Disponível em: <https://www.iosrjournals.org/iosr-jhss/papers/Vol.25-Issue12/Series-11/E2512112735.pdf>.

Souza, Moreno e Pimenta 2018 SOUZA, G. C. M.; MORENO, R. L.; PIMENTA, T. C. Evaluation of Pattern Recognition in Myoelectric Signal Using Netlab GLM. In: *2018 25th International Conference "Mixed Design of Integrated Circuits and System" (MIXDES)*. Gdynia: IEEE, 2018. p. 436–440. ISBN 9788363578145. Disponível em: <https://ieeexplore.ieee.org/document/8436881/>.

Souza, Moreno e Pimenta 2020 Souza, G. C. M.; Moreno, R. L.; Pimenta, T. C. Pattern recognition in myoelectric signals using deep learning, features engineering, and a graphics processing unit. *IEEE Access*, v. 8, p. 208952–208960, 2020.

Srinivasan et al. 2018 SRINIVASAN, V. B. et al. Finger Movement Classification from Myoelectric Signals Using Convolutional Neural Networks. In: *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. Kuala Lumpur, Malaysia: IEEE, 2018. p. 1070–1075. ISBN 9781728103778. Disponível em: <https://ieeexplore.ieee.org/document/8664807/>.

Szegedy et al. 2015 SZEGEDY, C. et al. Going deeper with convolutions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, 2015. p. 1–9. ISBN 9781467369640. Disponível em: <http://ieeexplore.ieee.org/document/7298594/>.

Sánchez-Maroño, Alonso-Betanzos e Tombilla-Sanromán 2007 SáNCHEZ-MAROñO, N.; ALONSO-BETANZOS, A.; TOMBILLA-SANROMáN, M. Filter Methods for Feature Selection – A Comparative Study. In: YIN, H. et al. (Ed.). *Intelligent Data Engineering and Automated Learning - IDEAL 2007*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. v. 4881, p. 178–187. ISBN 9783540772255. Disponível em: <http://link.springer.com/10.1007/978-3-540-77226-2_19>.

Tam et al. 2021 TAM, S. et al. Intuitive real-time control strategy for high-density myoelectric hand prosthesis using deep and transfer learning. *Scientific Reports*, v. 11, n. 1, p. 11275, dez. 2021. ISSN 2045-2322. Disponível em: <http://www.nature.com/articles/s41598-021-90688-4>.

Teban et al. 2018 TEBAN, T.-A. et al. Recurrent Neural Network Models for Myoelectricbased Control of a Prosthetic Hand. In: *2018 22nd International Conference on System Theory, Control and Computing (ICSTCC)*. Sinaia: IEEE, 2018. p. 603–608. ISBN 9781538644447. Disponível em: <https://ieeexplore.ieee.org/document/8540720/>.

Tharwat et al. 2017 THARWAT, A. et al. Linear discriminant analysis: A detailed tutorial. *Ai Communications*, v. 30, p. 169–190,, 05 2017.

Toledo-Pérez et al. 2019 TOLEDO-PéREZ, D. C. et al. Support Vector Machine-Based EMG Signal Classification Techniques: A Review. *Applied Sciences*, v. 9, n. 20, p. 4402, out. 2019. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/9/20/4402>.

Too et al. 2018 TOO, J. et al. Application of spectrogram and discrete wavelet transform for emg pattern recognition. *Journal of Theoretical and Applied Information Technology*, v. 96, p. 3036–3047, 05 2018.

Trimberger e Stephen 2015 TRIMBERGER; STEPHEN, M. Three Ages of FPGAs: A Retrospective on the First Thirty Years of FPGA Technology. *Proceedings of the IEEE*, v. 103, n. 3, p. 318–331, mar. 2015. ISSN 0018-9219, 1558-2256. Disponível em: <http://ieeexplore.ieee.org/document/7086413/>.

Varella 2017 VARELLA, C. A. A. *Análise Multivariada Aplicada As Ciências Agrárias*. 2017. Http://www.ufrrj.br/institutos/it/deng/varella/.

Venkatesh e Anuradha 2019 VENKATESH, B.; ANURADHA, J. A Review of Feature Selection and Its Methods. *Cybernetics and Information Technologies*, v. 19, n. 1, p. 3–26, mar. 2019. ISSN 1314-4081. Disponível em: <https://content.sciendo.com/doi/10.2478/cait-2019-0001>.

Vinuesa et al. 2020 VINUESA, R. et al. The role of artificial intelligence in achieving the Sustainable Development Goals. *Nature Communications*, v. 11, n. 1, p. 233, dez. 2020. ISSN 2041-1723. Disponível em: <http://www.nature.com/articles/s41467-019-14108-y>.

Wan 2019 WAN, X. Influence of feature scaling on convergence of gradient iterative algorithm. *Journal of Physics: Conference Series*, v. 1213, p. 032021, jun. 2019. ISSN 1742-6588, 1742-6596. Disponível em: <https://iopscience.iop.org/article/10.1088/1742-6596/1213/3/032021>.

Wang 2019 WANG, P. On Defining Artificial Intelligence. *Journal of Artificial General Intelligence*, v. 10, n. 2, p. 1–37, jan. 2019. ISSN 1946-0163. Disponível em: <https://content.sciendo.com/doi/10.2478/jagi-2019-0002>.

Wang et al. 2018 WANG, W. et al. Sensor Fusion for Myoelectric Control Based on Deep Learning With Recurrent Convolutional Neural Networks: SENSOR FUSION FOR MYOELECTRIC CONTROL WITH RCNNs. *Artificial Organs*, v. 42, n. 9, p. E272–E282, set. 2018. ISSN 0160564X. Disponível em: <http://doi.wiley.com/10.1111/aor.13153>.

Wang et al. 2018 WANG, Z. et al. Auto-tuning of hyperparameters of machine learning models. *International Conference on High Performance Computing in Asia-Pacific Region*, 01 2018.

Waris et al. 2019 WARIS, A. et al. Multiday evaluation of techniques for emg-based classification of hand motions. *IEEE Journal of Biomedical and Health Informatics*, v. 23, n. 4, p. 1526–1534, jul. 2019. ISSN 2168-2194, 2168-2208. Disponível em: <https://ieeexplore.ieee.org/document/8429072/>.

Xu, Tao e He 2010 XU, S.; TAO, J.; HE, N. An improved extrapolation method and its application in ultrasound computerized tomography. *Procedia Engineering*, v. 7, p. 335–341, 2010. ISSN 18777058. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1877705810010477>.

Xu et al. 2018 XU, X. et al. A Comparison of Outlier Detection Techniques for High-Dimensional Data. *International Journal of Computational Intelligence Systems*, v. 11, n. 1, p. 652, 2018. ISSN 1875-6883. Disponível em: <http://www.atlantis-press.com/php/paper-details.php?id=25892518>.

Young et al. 2013 YOUNG, A. J. et al. Classification of Simultaneous Movements Using Surface EMG Pattern Recognition. *IEEE Transactions on Biomedical Engineering*, v. 60, n. 5, p. 1250–1258, maio 2013. ISSN 0018-9294, 1558-2531. Disponível em: <http://ieeexplore.ieee.org/document/6377275/>.

Yu e Zhu 2020 YU, T.; ZHU, H. Hyper-parameter optimization: A review of algorithms and applications. *ArXiv e-prints*, 03 2020. Disponível em: <https://arxiv.org/pdf/2003.05689.pdf>.

Yu et al. 2019 YU, Y. et al. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*, v. 31, n. 7, p. 1235–1270, jul. 2019. ISSN 0899-7667, 1530-888X. Disponível em: <https://www.mitpressjournals.org/doi/abs/10.1162/neco_a_01199>.

Yuan et al. 2019 YUAN, Y. et al. Quantitative research of convolutional neural network and FPGA deployment. IEEE, Chongqing, China, p. 1437–1440, maio 2019. Disponível em: <https://ieeexplore.ieee.org/document/8785497/>.

Zhang 2016 ZHANG, Z. Missing data imputation: Focusing on single imputation. *Annals of translational medicine*, v. 4, p. 9, 02 2016.

Zhong et al. 2011 ZHONG, J. et al. Recognition of hand motions via surface emg signal with rough entropy. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, v. 2011, p. 4100–3, 08 2011.

Zhou, Jin e Dong 2017 ZHOU, F.-Y.; JIN, L.; DONG, J. Review of convolutional neural network. *Jisuanji Xuebao/Chinese Journal of Computers*, v. 40, p. 1229–1251, 06 2017.

A APPENDIX

A.1 ALGORITHM WRITTEN IN MATLAB THAT IMPLEMENTS THE LSTM NETWORK ALONG WITH BIOPATREC.

```
function [LSTM, acc] = LSTM_Init(trSet, trOut, vSet, vOut, tType)
%% Initialize LSTM
% Selection of output function
if strcmp(tType, '10')
    numHiddenUnits = 10;
elseif strcmp(tType, '50')
    numHiddenUnits = 50;
elseif strcmp(tType, '100')
    numHiddenUnits = 100;
elseif strcmp(tType, '200')
    numHiddenUnits = 200;
else
    numHiddenUnits = 500;
end

[row, collum] = size(trOut);
[row, features] = size(trSet);

inputSize = features;
numClasses = collum;

aux = [];
for i = 1:row
    for j = 1:collum
        if (trOut(i,j) == 1)
            aux(i) = j;
        end
    end
end

layers = [...
```

```
    sequenceInputLayer(inputSize)
    lstmLayer(numHiddenUnits,'OutputMode','last')
    %lstmLayer(numHiddenUnits,'OutputMode','last')
    fullyConnectedLayer(numClasses)
    softmaxLayer
    classificationLayer]

%Pick the all datase as batch - number of rows
maxEpochs = 500;
miniBatchSize = row;


trOut = transpose(aux);
trOut = categorical(trOut);
trSet = transpose(trSet);
trSet = num2cell(trSet,1);

vSet = transpose(vSet);
vSet = num2cell(vSet,1);

[row, collum] = size(vOut);
aux2 = [];
for i = 1:row
    for j = 1:collum

        if (vOut(i,j) == 1)
            aux2(i) = j;
        end

    end
end

%pool = parpool
%pool.NumWorkers
vOut = transpose(aux2);
vOut = categorical(vOut);

%validationMatlab = {vSet,vOut}

```

```
options = trainingOptions('rmsprop', ...
    'ExecutionEnvironment','auto', ...
    'GradientThreshold',1, ...
    'MaxEpochs',maxEpochs, ...
    'MiniBatchSize',miniBatchSize, ...
    'SequenceLength','longest', ...
    'Shuffle','once', ...
    'Verbose',0, ...
    'InitialLearnRate',0.0005,...
    'SquaredGradientDecayFactor',0.9999,...
    'Plots','training-progress');

net = trainNetwork(trSet,trOut,layers,options)
YPred = classify(net,vSet,'MiniBatchSize',miniBatchSize);
acc = sum(YPred == vOut)./numel(vOut)
LSTM = net
end
```

A.2 ALGORITHM WRITTEN IN MATLAB THAT IMPLEMENTS THE CNN NETWORK ALONG WITH BIOPATREC.

```
%% Initialize CNN
function [CNN, acc] = CNN_Init(trSet, trOut, vSet, vOut, tType)
[row, collum] = size(trOut);
[row, features] = size(trSet);
inputSize = features
numClasses = collum
aux = [];
for i = 1:row
    for j = 1:collum
        if (trOut(i,j) == 1)
            aux(i) = j;
        end
    end
end
%Pick the all datase as batch - number of rows
miniBatchSize = row;
trOut = transpose(aux);
```

```
trOut = categorical(trOut);
trSet = transpose(trSet);
%trSet = num2cell(trSet,[1 136 1296]);
vSet = transpose(vSet);
%vSet = num2cell(vSet,1);

[row, collum] = size(vOut);
aux2 = [];
for i = 1:row
    for j = 1:collum

        if (vOut(i,j) == 1)
            aux2(i) = j;
        end
    end
end
vOut = transpose(aux2);
vOut = categorical(vOut);
%Input size = Number features * Number channels, when witout PCA
layers = [
    imageInputLayer([inputSize/4 4 1])

    convolution2dLayer(3,(inputSize)/4,'Padding',[1])
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(1,'Stride',1)

    convolution2dLayer(3,(inputSize)/4,'Padding',[1])
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(1,'Stride',1)

    fullyConnectedLayer(numClasses)
    softmaxLayer
    classificationLayer];

%validationMatlab = {vSet,vOut}
```

```
options = trainingOptions('adam', ...
    'InitialLearnRate',0.003, ...
    'MaxEpochs',20, ...
    'MiniBatchSize',128, ...
    'Shuffle','every-epoch', ...
    'ValidationFrequency',15, ...
    'Verbose',true, ...
    'SquaredGradientDecayFactor',0.9,...
    'GradientThreshold',1,...
    'Plots','none');

trSet = reshape(trSet, [size(trSet,1)/4,4,1,size(trSet,2)]);
vSet = reshape(vSet, [size(vSet,1)/4,4,1,size(vSet,2)]);

net = trainNetwork(trSet,trOut,layers,options);
YPred = classify(net,vSet, 'MiniBatchSize', miniBatchSize);
acc = sum(YPred == vOut)./numel(vOut);
CNN = net;
net.Layers
end
```