International Journal of Advanced Engineering Research and Science (IJAERS)
Peer-Reviewed Journal
ISSN: 2349-6495(P) | 2456-1908(O)
Vol-9, Issue-12; Dec, 2022
Journal Home Page Available: https://ijaers.com/
Article DOI: https://dx.doi.org/10.22161/ijaers.912.73

# Machine Learning-mediated Gait Rating Based on Real Time Data Collected

Vandermi Silva[2] , Diogo Rezende[1], Jogno Vezu[1], Rafael Guedes, Walter Seiffert Silva[1], Andreza B. Mourão[1]

[1]University of the State of Amazonas (UEA), Brazil.
[2]Federal University of Amazonas (UFAM), Brazil.

Received: 22 Nov 2022,

Receive in revised form: 18 Dec 2022,

Accepted: 22 Dec 2022,

Available online: 31 Dec 2022



***Keywords* — Footstep Classification, Deep Learning, Artificial Intelligence, Machine Learning, Health.**

***Abstract*— *Machine Learning is a subgroup of Artificial Intelligence in which we use algorithms and methods to identify patterns and data with high volume. Tools such as Logistic Regression, K-Means, Dummy Classifier, Random Forest, KNN and SVM are very useful in identifying patterns. Deep Learning, a subgroup of Machine Learning, uses algorithms that mimic the neural network of the human brain, this network can be built by stacking layers of neurons, fed by large volumes of data, capable of performing classification tasks, impossible for humans. Tools such as RNA and RNC are examples used in Deep Learning. The use of these tools in the classification of types of footsteps (acquired data to be classified), applied to health, is very useful in speeding up the response of diagnoses with precise answers, helping physicians, physiotherapists, physical educators and scientists to employ and develop more efficient treatments, effective, and improve the health and quality of life of patients.*

## I. INTRODUCTION

Machine Learning (ML) is a sub-area of Artificial Intelligence (AI) whose main objective is the use of algorithms and methods for detecting patterns obtained from data with large volumes. In addition, ML allows predicting future patterns, in addition to classification, and being able to base decision-making on these results (MURPHY, 2012).

The ML field has grown significantly over the last few decades. Currently, ML has been employed in several areas (scientific and commercial) (JORDAN, MITCHELL, 2015). Examples include speech recognition, computer vision, robot control, and natural language processing. Problems solved by ML are usually of high complexity, composed of a large volume of input data. These problems are then divided into smaller problems, which are solved, composing the overall answer to the larger problem (JORDAN, MITCHELL, 2015). Several studies have used ML applications to solve problems such as those seen in (ABEDI, 2012; BENTLEY, 1975; BISHOP, 2006; BREIMAN, 1986; BREIMAN, 2001; BREIMAN, 2017; CANNATA, 2011; HUANG, 2022 ; MASOTTI, 2006; PETRELLI, 2017; PETRELLI, 2020; PETRELLI, 2016; PETRELLI, 2003; PETRELLI, 2003; ZUO, 2011).

A common feature of methods belonging to ML is that they are not developed to process a conceptual model defined a priori, but rather try to discover the complexities of large datasets through the so-called learning process (BISHOP, 2006; SHAI, 2013). The purpose of the process is to convert experience into “expertise” or “knowledge” (SHAI, 2013). In this way, we can make an analogy of this concept with the form of human learning, which learns something new based on lived experiences.

Examples of methods belonging to ML include Logistic Regression, K-Means grouping (clustering algorithm or cluster analysis), Dummy Classifier, Random Forest, KNN (K Nearest Neighbor) and SVM (support-vector machine) (GÉRON, 2019) .

Another important subgroup of Machine Learning, with useful tools for pattern recognition, detection and prediction, is Deep Learning (DL). Deep Learning is currently an extremely active research area, which has achieved great success in a wide range of applications, such as speech recognition, computer vision, among others. Companies like Google and Facebook analyze large volumes of data extracted from various applications using DL concepts, for example, applications for translation, speech pattern recognition and computer vision. (GRACE, et al., 2018; COPELAND, 2016). DL is based on the architecture of the human brain, to build a set of virtual units (perceptron) which will compose an intelligent machine. This is the basis of an artificial neural network (ANN). An ANN is a ML model inspired by the network of biological neurons in the brain (GÉRON, 2019).

Deep Learning is the style of machine learning that is done with a deep neural network, in essence, an accurate perception of artificial intelligence, which looks like a human being and is capable of generating content based on learning from this assimilation. DL algorithms are able to analyze unstructured data without any kind of pre-processing or supervision (GOODFELLOW, BENGIO, COURVILLE, 2016).

Among the numerous existing applications of Machine Learning, we have the health area. Among the targeted works, we can mention the work proposed by Schmidt et al. (2018) and APACHE II (Acute Physiology and Chronic Health Evaluation), A Model for Predicting Mortality in Intensive Care Units Based on Deep Learning, uses the DL technique to predict risk of death to make therapeutic decisions more efficient.

Santos et al. (2017), in An Approach to Classifying Dermatoscopic Images Using Deep Learning with Convolutional Neural Networks, automatically identifies melanoma in images, using DL with convolutional neural networks, obtaining 91.05% accuracy.

In Silva (2017), Detection of Epileptic Seizures in Electroencephalograms Using Deep Learning, aims to classify intracranial electroencephalogram (iEEG) exams, for recognition and cataloging of epileptic seizures in humans.

In the paper by Secretary and Pires (2018), Use of Computer Vision for Automatic Cell Counting in Images Obtained by Microscopes, used DL techniques like CNN in the development of an automatic cell counter to visualize and analyze the images, in order to facilitate diagnostic and treatment.

Further specifying our area of implementation, we can cite the NASCIMENTO (2019) and VIEIRA (2018). The first aims to develop a wearable device, based on inertial sensors and artificial neural network (ANN) to identify the type of stepping during gait, to aid diagnosis and follow-up to be carried out by health professionals. The data obtained were used for feature extraction and arranged as inputs in a multilayer Perceptron-type ANN (Multilayer Perceptron-MLP) to perform the classification of footfall types. The second aims to develop an instrumented insole, based on ceramic piezoelectric sensors and artificial neural networks to identify the type of step; and thus help in analyzes and diagnoses of health professionals. Plantar pressure is used in studies of postural correction, movement analysis, correction of the type of stepping and identification of diseases in the plantar region. With the input data being fed by plantar pressures, they were processed and divided into samples, which were used as the database of the implemented ANN.

In this work, Machine Learning tools will be applied to data acquired through sensors arranged in insoles for characterization, classification and recognition of the types of steps. These data were treated and organized in order to be provided as input for logistic regression, k-means, dummy classifier, random forest, KNN, SVM, RNA and RNC methods. The results were compared and concluded in order to show the method with the most accurate response to this situation.

## II. METHODOLOGY

The analysis of the type of footsteps is of fundamental importance in the health treatment and diagnosis of the most varied types of diseases (NASCIMENTO, 2019). This analysis is provided by the use of inertial sensors, which have low cost, reduced size and low energy consumption (BERAVS et al., 2011). Step analysis systems based on these sensors bring benefits to measure and establish metrics on the individual's health (MARTINEZ-MENDEZ; SEKINE; TAMURA, 2011). Among the possibilities of arrangements for acquiring data related to footsteps, we have the in-shoe systems. These are insole-shaped acquisition systems that are installed inside the shoes, allowing analysis in external environments and in dynamic daily activities. This type of technology allows for greater mobility and its operation is based on measuring the plantar pressure between the foot and the sole of the shoe (PEDAR SYSTEM, 2019; TECKSCAN, 2019).

The Pedar© system Figure 2-A) has up to 1024 capacitive sensors, NiMH battery power, data communication via USB or Bluetooth® and 32 MB internal flash memory for storing information (PEDAR SYSTEM, 2019). The F-scan© system (Figure 2-B) uses 25 resistive sensors per square inch, has an acquisition frequency of up to 600 Hz, battery power and data communication via USB and Wi-Fi™ (TECKSCAN, 2019).


*Fig.1 - in-shoe system..*

Source: (PEDAR SYSTEM, 2019; TECKSCAN, 2019).

The data obtained in the in-shoe arrangement had the organization shown in figure 2.

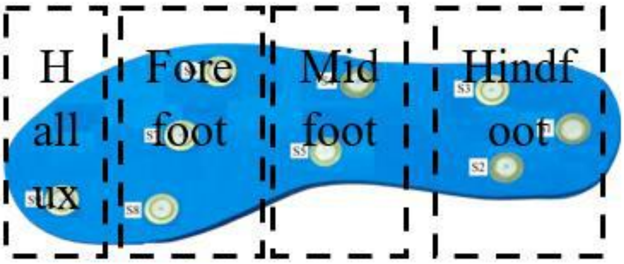


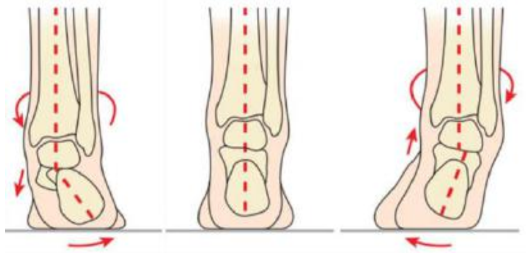

*Fig.2 –Insole, on the right foot, mounted with 9 piezoresistive sensors (s1, s2, s3, s4, s5, s6, s7, s8, s9), for testing.*

Source: Author.

The sensors used were piezoelectric, which have the characteristic of presenting a change in electrical charge proportional to the direction applied to the mechanical stress. The opposite also occurs, that is, there will be a deformation proportional to an application of an electric field (WEBSTER, 1999). These sensors, when subjected to the action of a force, generate a signal in the electrical voltage, adapting to the requirements of the experiment to be used. The tension value varies directly proportional to the force applied to the sensor, which in the experiment is the plantar pressure (VIERIA, 2018).

In the literature, the plantar region can be divided between the regions of the foot into four main parts: Hindfoot, Midfoot, Forefoot and Hallux (WAFAI et al, 2015; RODRIGUES et al, 2014; SHU et al, 2009). Returning to figure 2, we have the arrangement of the sensors in the insole and the designation of each of the regions described. Regarding the number of points in the division of the plantar region, the most used premise is the verification of which analyzes would be developed with the instrumented insole (RAZAK et al, 2012). For example, in the Hindfoot, to measure only the pressure in this region, only one point is needed to cover the region; however, for gait type measurement, two points (minimum) are required (inner and outer regions). This process is repeated for all other plantar regions (WAFAI et al, 2015; RODRIGUES et al, 2014; SHU et al, 2009).

Figure 3 - Types of bone alignments for the right foot, being pronated footing (left), neutral footing (center) and supination footing (right) (rear view).



[redacted]

Source: adapted from Norris (2011)

Having the technical introduction of the data collection tool used with its references, we can classify the types of steps. There are three main types of stepping. The first type is pronation (pronated stepping), which is characterized by the inward misalignment of the bone structures of the ankle, generating greater application of force in the inner region of the foot (Figure 3). The second type is neutral (neutral stepping), the step is performed correctly, better distributing pressure throughout the foot. The third type is the supinated step (supinated step), with the step outside, forcing the outside of the foot (GUIMARÃES et al, 2000; SILVA, 2015).

Having introduced the relevant concepts to the equipment, we developed in more detail the ML methods used, the objective of this work.

The first method to be described is logistic regression. This method is commonly used to estimate the probability that an instance belongs to a particular class. If the estimated probability is greater than 50%, then the model predicts that the instance belongs to this class (called the positive class, labeled "1), and otherwise it predicts that it does not (i.e., it belongs to the negative class, labeled “0”). This makes it a binary classifier.

As with the linear regression model, the regression model calculates a weighted sum of the input features (plus a bias term), but instead of producing the result directly as the Linear Regression model does, it outputs the logistic of this result. , given in the equation below:

$$\hat{p} = h_\theta(x) = \sigma(x^T\theta) \qquad (1)$$

The logistic function, denoted by $\sigma$, is a sigmoid function, which returns a number between 0 and 1. This function is represented by equation (2), shown in figure 4.

$$\sigma(t) = \frac{1}{1+exp(-t)} \qquad (2)$$

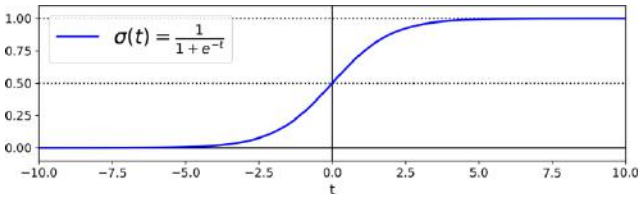



*Fig.4 - Logístic function.*

Source: Author.

Once the Logistic Regression model has estimated the probability $\hat{p} = h_\theta(x)$, which is an instance x belonging to the positive class, it can make its prediction ŷ easily with equation (3):

$$\hat{y} = 0, if\ \hat{p} < 0.5\ or\ \hat{y} = 1, if\ \hat{p} \geq 0.5 \qquad (3)$$

Notice that $\sigma(t) < 0.5$ when $t < 0$, and $\sigma(t) \geq 0.5$ when $t \geq 0$. Thus, the Logistic Regression model predicts 1 if $x^T\theta$ is positive and 0 if it is negative (GÉRON, 2019).

The next method used to recognize the type of footfall was K-Means. This method is based on clustering, that is, depending on the context, data can be labeled in sets with similar characteristics. For ungrouped data, the K-Means method is a simple algorithm capable of grouping data into similar datasets very quickly and efficiently, usually in just a few iterations. It was proposed by Stuart Lloyd at Bell Labs in 1957 as a technique for pulse code modulation, but was not published outside the company until 1982. In 1965, Edward W. Forgy published virtually the same algorithm, so K-Means sometimes is referred to as Lloyd-Forgy (LLOYD, 1982).

K-Means follows some basic steps like:

1 - choose a centroid $c^{(1)}$, randomly from the data set;

2 - choose a new centroid $c^{(i)}$, according to the instance $x^{(i)}$ with probability $D(x^{(i)})^2 / \sum_{j=1}^{m} D(x^{(j)})^2$, where $D(x^{(i)})$ is the distance between the instance $x^{(i)}$ (sample of the dataset) and the closest centroid that has already been chosen. This probability distribution ensures that instances farthest from the already chosen centroids are much more likely to be selected as centroids.

3 - repeat the previous steps until all k centroids have been chosen.

The number of k clusters can be calculated according to the best response obtained by this method. Initially, we can consider k = 5. Not always k = 5 will generate a satisfactory result, which can even generate a poor quality output. Thus, to assist in choosing the most appropriate value of k, we use the Inertia function.

The inertia function does not behave properly when trying to choose k when we have many clusters, as it decreases as we increase k. In fact, the more clusters there are, the closer each instance is to its nearest centroid, and therefore the lower the inertia. Note figure 4:

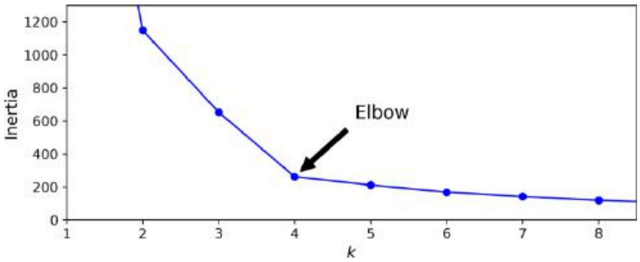



*Fig.4 -inertia function x number of clusters. The curve shown usually contains an inflection point called an "elbow".*

Source: Modified from (GÉRON, 2019).

As shown, the inertia decays rapidly as we increase k up to 4. For values above 4, the function decays more slowly. This curve is roughly arm-shaped, and there is an "elbow" at k = 4. So, if we didn't know, k = 4 would be a good choice: any lower value would be overfitted, while any higher value would not. would have an adequate result, and we could be splitting perfectly good clusters in half for no reason (GÉRON, 2019).

As in the logistic regression method, in K-Means we use the Scikit-learn package to apply the methods to the data set.

The dummy classifier method is a type of classifier that does not generate any insights into the data, and classifies them using only simple rules. The behavior of the classifier is completely independent of the training data as trends in the training data are completely ignored and instead it uses one of the strategies to predict the class label. This method is only used as a simple baseline for the other classifiers, i.e. any other classifier is expected to perform better on the given dataset. It is especially useful for datasets where a class imbalance is certain. It is based on the philosophy that any analytical approach to a classification problem must be better than a random guessing approach.

This type of model should not be used in real problems, as explained in the Dummy Classifier documentation on Scikit-learn – "*Dummy Classifier is a classifier that makes predictions using simple rules. This classifier is useful as a simple baseline to compare with other (real) classifiers. Do not use it for real problems.*". (VRECH, 2021).

The SVM (support-vector machine) emerged in 1992, when there was a need for classification and regression tools based on some kind of prediction. It was introduced by Vapnick, Guyon and Boser in COLT-92. To separate any data, we need to define certain classes and depending on the complexity of the data set, we define a classification of linear or non-linear type. The SVM method can be defined as a prediction tool, in which we look for a line or decision boundary called a hyperplane, which separates data sets or classes, thus avoiding data overfit. It uses the assumption of a linear space in a high-order multidimensional space. It is also capable of sorting non-linear data using kernel functions.

Currently, Neural Networks are used in almost all fields of classification and regression and contribute more in Artificial Intelligence. In these, we have the neurons that are responsible for building a network, i.e. grouping similar datasets or similar data classes, and then applying both supervised and unsupervised learning methods that initially showed good results. However, later, as the number of nodes increased, the complexity (COPELAND, 2015). Thus, we conclude that for a small number of nodes, neural networks are more adequate. SVM overcomes these drawbacks and can also be applied to large datasets. Neural Networks are simple and can also use multilayer perceptrons (MLP) where MLP uses recurrent and feedback networks. MLP properties include the approximation of non-linear functions which again may not provide accurate results (DAVID, 1996; KULKAMI, 2013; MITCHELL, 1997).

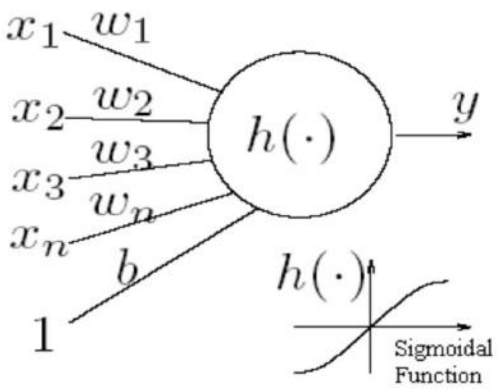



*Fig.5 - Simple Neural Network.*

Source: Skapura, 1996; Mitchell, 1997; Jakkula, 2013.

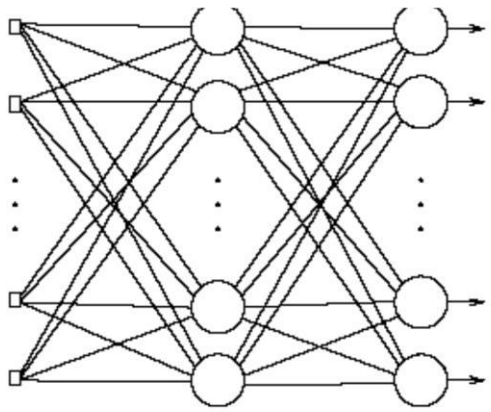


*Fig.6 - Multilayer Perceptron*

Source: Skapura, 1996; Mitchell, 1997; Jakkula, 2013.

The SVM is used for classification and regression. This strategy separates the studied samples by drawing a decision limit. This limit is known as the hyperplane in the case of linear classification. Figure 7 shows the classification of several decision limits, which are capable of classifying different sets of different samples. Thus, the question is to decide which hyperplane should be selected so that we have a better division into sets of samples. For this, a hyperplane that is equal for both sample categories is needed, which means that of all hyperplanes or decision boundaries, only one of them should be selected. To select the hyperplane, follow these steps:

1. Define a function that is the limit between different sets of data (samples);

2. Select a hyperplane and calculate its distance from both sets of data it divides.

i. If the calculated distance is maximum on both sides compared to the previous hyperplane, select this hyperplane as the new decision boundary.

ii. Mark the samples that are close to the hyperplane as support vectors. (helps in selecting the decision threshold).

3. Repeat step 2 until you find the best hyperplane.

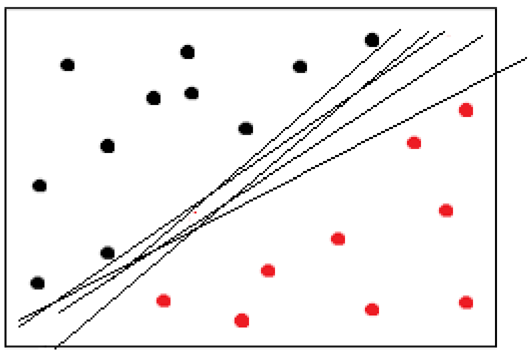


*Fig.7 - Hyperplane*

Source: Skapura, 1996; Mitchell, 1997; Jakkula, 2013.

The random forest is a combination of tree predictors, so that each tree depends on the values of a random vector, sampled independently, and with the same distribution for all trees in the forest. The error generalization for forests converges, up to a limit, as the number of trees in the forest becomes large.

The error generalization of a forest of tree classifiers depends on the individual trees in the forest and on the correlation between these classifiers (BREIMAN, 2001). Using a random selection of features to split each node produces lower error rates and is more robust against noise. Internal estimates monitor error, strength and correlation, and these are used to show the response to increasing the number of features used in the split. Internal estimates are also used to measure the importance of the variable. These ideas also apply to regression. Consider splitting as decisions to be taken from a previous node, and thus constituting the decision tree.

The common element to the execution of the random forest is that, for the k-th tree, a random vector $\phi_k$ is generated, regardless of the random vectors passed $\phi_1,..., \phi_{k-1}$ but with the same distribution; and a tree is built using the dataset for training and $\phi_k$, resulting in a classifier h(**x**, $\phi_k$), where x is an input vector. For example, by taking the random vector $\phi$, N resulting nodes are randomly generated, where N is the number of elements in the training set. The random choice of one of these nodes consists of a randomly chosen integer between 1 and k. The nature and dimensionality depends on its use in building trees. After a large number of trees are generated, they vote for the class with the most occurrences. Therefore, this procedure is called a random forest. According to BREIMAN (2001).

A random forest is a classifier consisting of a collection of classifiers {h(**x**, $\phi_k$), k = 1, . . .}, where the {$\phi_k$} are randomly distributed identically and independently distributed vectors and each tree points to the most frequent class of input x. For more formalism and definitions, (BREIMAN, 2001) must be consulted.

The KNN (k-nearest neighbors) classification method classifies new data into categories, assuming that they have similarity with existing data, and classifying them into existing data categories, according to their similarity. The KNN algorithm can be used for regression and classification (ROZOS, 2022). This algorithm uses an instance based on a non-parametric model (RUSSELL, 2010). KNN is non-parametric in that it depends on the evaluated data to be operated on (in contrast, parametric models need data only during model training), and it is instance-based in that it takes into account the similarity with the instances in the training set on which the inference will be made.

The KNN methodology returns a set of observations made during an iteration prior to the current one, during model simulation and training. The model uncertainty can be estimated by the formula:

$$s = f(\mathbf{KNN}(k,\boldsymbol{x})) \quad (4)$$

where s is the value related to the uncertainty calculated during the iterations, when the observed state is x, x is the vector (or scalar) that defines the state of the model, KNN(k,x) returns the set of k observations, which according to KNN, has more similarity to x, and $f$: $R^k \to R$ is the function that returns the set of values related according to some statistical property of the set given by KNN(k,x), in a typical KNN mean regression application. The previous values refer to the time instance t. The t variable is omitted from equation (4) to simplify understanding. As an example for understanding, we will make some considerations and assumptions. We consider that 3 functions were used as f in equation (4), 90% of the data, 10% of the data and the mean value. Regarding the parameter k, this is the hyperparameter. For low values of k, it will result in overfitting while for high values of k it will result in underfitting (RUSSELL, 2010) and bias. The values of this parameter can vary from 10 (for small datasets) to 1000 (for large datasets, with hundreds of thousands of records). According to the value x, the following considerations can be made:

- 1D. This is the simplest approximation that includes only the dataset accessed from the time-lapse model. *t*, *x* = $Q_t$. KNN returns observations of *k* that correspond to period/distance calibration values that are closest to $Q_t$.
- 2D. The array of elements has two components returned during simulation or model training, $\boldsymbol{x} = (Q_t, Q_{t-1})$, KNN returns the *k* observations that correspond to the *k* calibration vectors, for each step, that are closest to the vector's 2D Euclidean space $(Q_t, Q_{t-1})$.
- 2D. The vector of elements are the responses of $Q_t$ and the changes in responses obtained between *t* - 1 e *t*, $\boldsymbol{x} = (Q_t, Q_{t-1} - Q_t)$.
- 2D. The vector of elements such as responses from $Q_t$, and a binary value, 0 if the response increases, and 1 if it does not. This binary value can be obtained with the function $\varphi(.) = max(0, (.)/|.|)$, $\boldsymbol{x} = (Q_t, \varphi(Q_{t-1} - Q_t))$.

In the last two options, the elements of vector x need to be scaled to be used as Euclidean distances. For this reason, z-score normalization can be employed (MINSKY, 1969; TOWARDS DATA SCIENCE, 2022), the normalization of the parameters are obtained from the training set only, and thus the normalization is applied to both data sets. .

In this way, the KNN method uses the Euclidean distance of data in relation to sets of data categorized into classes, due to the similarity that these data have among themselves, and thus grouped into sets that categorize them due to their similarities. With the calculation of this distance, of the new record in relation to the existing sets, we can categorize it into one of these sets, classifying it based on the smallest of the distances found while using the KNN.

The methodology of Artificial Neural Networks (ANN) has as main characteristics of neurocomputing, its development and applications. The main attention is given to feedforward Neural Networks, especially to the error caused in backpropagation algorithms and in backpropagation neural networks (BPNN`s).

The nervous systems of living organisms are generally composed of three parts: the central nervous system, the peripheral nervous system, and the autonomic nervous system. The central nervous system contains the brain and spinal cord. It is a huge network composed of neural units, connections and joints. The main function of this system is to control the activity of the entire organism based on information processing. The information signals are transmitted along the peripheral nervous system that has contact with external sensors and effectors. The autonomic nervous system oversees the activity of internal organs. The most sophisticated part of the nervous system is the brain. It can be considered as a highly complex, non-linear and parallel information processing system. The basic elements of the brain are neural cells called neurons (WASZCZYSZYN, 1999).

Artificial neural networks (ANNs) are basic models of a biological nervous system. ANN models try to simulate the behavior of the human brain. Especially, Artificial Neural Networks (ANN's) are used using the following expression (5):

compute = storage + transmission + processing. (5)

The use of ANN's in computing is called neurocomputing.

For an artificial neuron (AN) model, N is the various inputs to an output. The body of the neuron is composed of: sum of the junctions of the neurons $\sum$ and the activation function F.

In the model shown in figure 8, the variables and parameters used are:

$\boldsymbol{x} = \{x_l, \ldots, x_N\}$ - vector with entries (6)

$\boldsymbol{w} = \{w_l, \ldots, w_N\}$ - weights vector (7)

$\boldsymbol{b} = -\theta = w_0$ - constant components (bias) (8)

$\theta$ - limit (9)

$v = u + b = net - \theta = \sum_{j=l}^{N} w_j x_j - \theta = \sum_{j=0}^{N} w_j x_j$ - potential AN (10)

$F(v)$ - activation function (11)

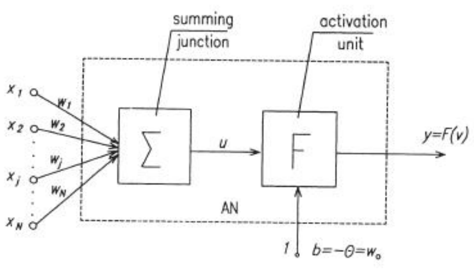



*Fig.8 - Model of an artificial neuron.*

Source: (WASZCZYSZYN, 1999).

There are several activation functions that can be used as functions of the linear type, binary step, bipolar step, sigmoid (logistic or binary sigmoid), bipolar sigmoid. (WASZCZYSZYN, 1999).

Among the various types of neural networks, there are a variety of complex connections between neurons, which give neural networks a high processing potential (WASZCZYSZYN, 1999). There are 3 main types of neural network architectures: feedforward, recurrent and cellular. In the feedforward neural network, signals are transmitted in one direction, from inputs to outputs. The standard architecture of this network corresponds to layers of neurons. Normally neurons are not connected to each other in a layer or layer, but they are connected with the neurons of the previous layer and with the neurons of the next layer (WASZCZYSZYN, 1999).

Convolutional neural networks (CNN) are biologically inspired architectures capable of being trained and learning invariant representations to scale, translation, rotation and related transformations (LECUN; KAVUKCUOGLU; FARABET, 2010). One of the key issues in pattern recognition in images is to know what is the best way to represent the characteristics of the data to be recognized in a robust and invariant way to lighting, orientation, pose, occlusion, among others. Although several descriptors have been developed to extract these features artificially, it is desirable that a recognition system be able to extract this representation automatically through the raw data, in the case of image recognition, the images themselves (JURASZEK, 2014 ).

The CNN (Convolutional Neural Network) emerged to represent this type of architecture. CNN make up one of the types of algorithms in the area known as deep learning and are designed for use with two-dimensional data, making them a good candidate for solving problems involving image recognition (AREL; ROSE; KARNOWSKI, 2010).

CNN are multistage architectures capable of being trained. Receptive fields are highly correlated to the location of the stimulus in the captured image. CNN use this concept by forcing a pattern of connectivity between layers of artificial neurons. Figure 9 shows this organization where a layer i is connected to a small sub-region of layer i≠1. In this example the layer m≠1 corresponds to the input image. The upper layer m has a receptive field of size 3, where each neuron receives the stimulus from 3 neurons in the previous layer. The m+1 layer is similar to the previous layer, having a receptive field of size 3 with respect to the previous layer, however, with a receptive field of size 5 with respect to the input image.

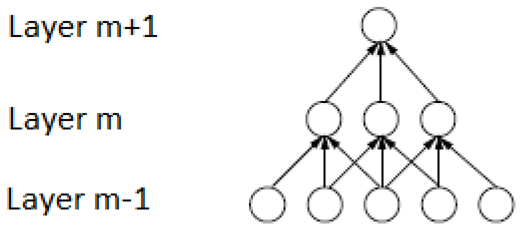



*Fig.9 - Organization of receptive fields in a CNN*

Source: Author.

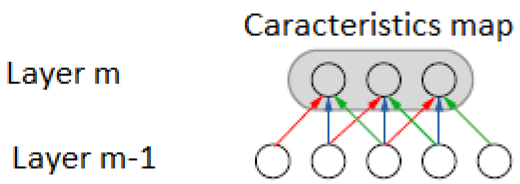



*Fig.10 - Sharing parameters for creating a feature map.*

Source: Author.

Considering this analogy, each receptive field is considered a non-linear filter where their weights must be learned so that the neuron is activated only when a certain stimulus is present in the area where the filter was applied. Each filter is applied to the entire input image (or previous layer) in a convolutional way, the result of applying this filter is called a feature map. Each feature map shares the same parameters. Figure 10 shows the sharing of parameters. This strategy ensures that a given feature will be detected by the feature map regardless of its position in the input image (or map).

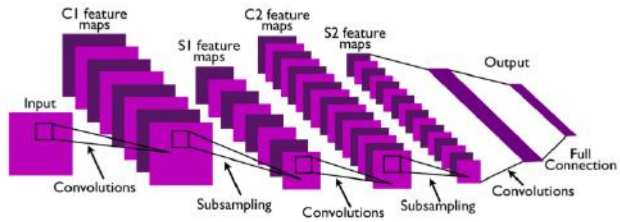
The data inputs for each stage are a set of feature maps. When applied using color images, the first stage input consists of the three color channels of the image. Each two-dimensional vector works as a feature map. At the output of each stage, each map matches the convolution of the input map through a filter. Applying the filter to the map highlights some features. Each filter is responsible for highlighting a different feature. In the first stage, filters highlight lines and gradients in different orientations.

A feature map is obtained by convoluting an input image through a linear filter followed by adding a bias term and applying a non-linear function. being the layer *k*, the filters determined by a set of weights $W^k$and a bias term $b_k$ and the convolution operator *, the Equation 12 (LECUN; KAVUKCUOGLU; FARABET, 2010) shows getting the feature map $h^k$ for a non-linear function *f*.

$$h_{ij}^k = f((W^k * k)_{ij} + b_k) \qquad (12)$$

Each stage is composed of three stages, filtering (filter bank layer), non-linearity layer and reduction stage (feature pooling layer), which represents the receptive field. A CNN can be composed of one or more stages which each contain the three steps. Figure 10 shows a CNN with a single input feature map (eg a grayscale image) with two convolutional stages C1+S1 e C2+S2.

Figure 11 - Convolutional neural network with two stages




Source: (LECUN; KAVUKCUOGLU; FARABET, 2010).

In JURASZEK (2014) shows a detailed process for implementing the process steps of a CNN.

## I.FIGURES AND TABLES

All data used in this work were acquired from insoles with 9 sensors attached to them, as previously described, for the right and left feet. Analyzes were performed for data from both feet. For this, the data were normalized, so as to be within a range of minimum and maximum values (from 0 to 1), removing from the data the sensors that contained only null values.

With the KNN clustering method, the inertia curve was used to calculate the appropriate k value to be used in the K-means algorithm. In this work, k = 3, as shown in Figure 12, and the criteria for choosing k previously discussed.

Thus, the sensors present were aggregated into sets (clusters), as shown in Figure 13, according to the types of steps, as follows: cluster 0 - pronated step, cluster 1 - supinated step, cluster 3 - neutral step. The total number of samples considered was 5816, of which, after labeling, 2358 (36.16%) are supine, 2099 (23.35%) are prone and 1359 (23.35%) are neutral.

With the data labeled and identified, we now have the necessary information for training and testing the machine learning methodologies explained in the previous section. Thus, we will start with the Random Forest methodology.

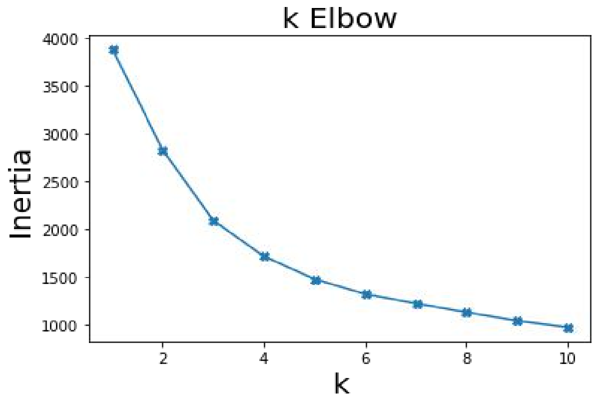



*Fig.12 - Choice of k value according to the inertia graph.*

Source: Author.

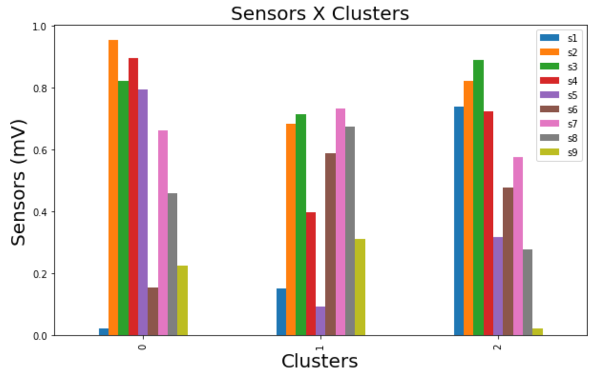



*Fig.13 - Clustering of data into clusters due to their characteristics.*

Source: Author.

With the Random Forest, 65% of the data were used for training the algorithm, and 35% of the data for testing. As a result of the trained model, an accuracy of 98.13% was obtained.

Table 1 shows the results obtained, as well as figure 14, which shows the confusion matrix.

Table 1 -Random Forest metrics after model training and testing.

| | Precision | Recall | f1-score | support |
|---|---|---|---|---|
| neutral | 0.98 | 0.99 | 0.98 | 761 |
| pronated | 0.99 | 0.98 | 0.98 | 460 |
| supinated | 0.98 | 0.98 | 0.98 | 815 |
| acurracy | | | 0.98 | 2036 |
| macro avg | 0.98 | 0.98 | 0.98 | 2036 |
| weighted avg | 0.98 | 0.98 | 0.98 | 2036 |

Source: Author.

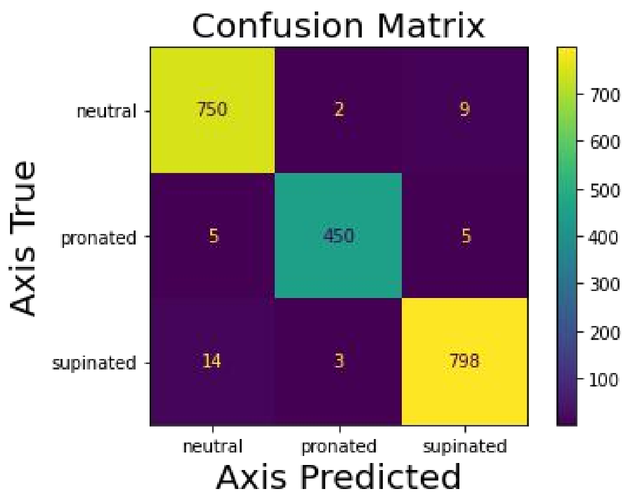



*Fig.14 - Confusion matrix according to Random forest.*

Source: Author.

Note that of the 35% (2036 samples) of the test data, 750 were correctly classified as neutral, 450 were classified as correctly prone, and 798 were correctly classified as supine.

Using the ANN methodology, 65% of the data were used for training the algorithm, and 35% of the data for testing. As a result of the trained model, an accuracy of 99.21% was obtained.

Table 2 shows the results obtained, as well as figure 15, which shows the confusion matrix.

*Table 2 - ANN metrics after model training and testing.*

| | Precision | Recall | f1-score | support |
|---|---|---|---|---|
| neutral | 0.99 | 1.00 | 0.99 | 761 |
| pronated | 1.00 | 0.99 | 0.99 | 460 |
| supinated | 1.00 | 0.99 | 0.99 | 815 |
| acurracy | | | 0.99 | 2036 |
| macro avg | 0.99 | 0.99 | 0.99 | 2036 |
| weighted avg | 0.99 | 0.99 | 0.99 | 2036 |

Source: Author.

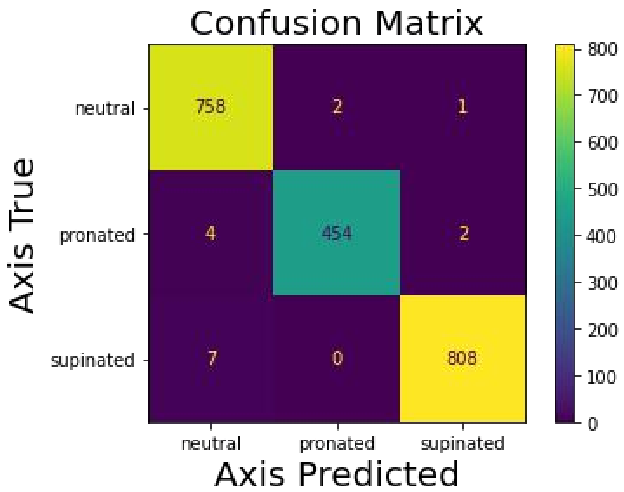



*Fig.15 - Confusion matrix according to ANN.*

Source: Author.

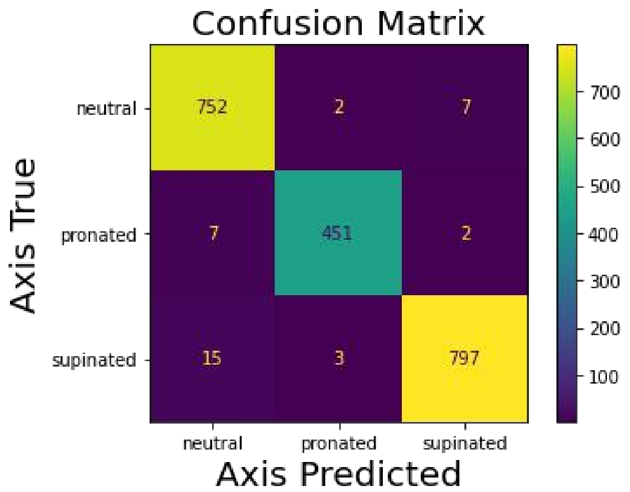



*Fig.16 - Confusion matrix according to KNN.*

Source: Author.

Note that of the 35% (2036 samples) of the test data, 758 were correctly classified as neutral, 454 were correctly classified as prone, and 808 were correctly classified as supine.

Using the KNN methodology, 65% of the data were used for training the algorithm, and 35% of the data for testing. As a result of the trained model, an accuracy of 98.23% was obtained.

Table 3 shows the results obtained, as well as figure 16, which shows the confusion matrix.

*Table 3 - KNN metrics after model training and testing.*

| | Precision | Recall | f1-score | support |
|---|---|---|---|---|
| **neutral** | 0.97 | 0.99 | 0.98 | 761 |
| **pronated** | 0.99 | 0.98 | 0.98 | 460 |
| **supinated** | 0.99 | 0.98 | 0.98 | 815 |
| **acurracy** | | | 0.98 | 2036 |
| **macro avg** | 0.98 | 0.98 | 0.98 | 2036 |
| **weighted avg** | 0.98 | 0.98 | 0.98 | 2036 |

Source: Author.

Note that of the 35% (2036 samples) of the test data, 752 were correctly classified as neutral, 451 were correctly classified as prone, and 797 were correctly classified as supine.

Using the RNC methodology, 65% of the data were used for training the algorithm, and 35% of the data for testing. As a result of the trained model, an accuracy of 96.70% was obtained.

Table 4 shows the results obtained, as well as figure 17, which shows the confusion matrix.

*Table 4 - RNC metrics after model training and testing.*

| | Precision | Recall | f1-score | support |
|---|---|---|---|---|
| **neutral** | 0.95 | 0.98 | 0.96 | 761 |
| **pronated** | 1.00 | 0.95 | 0.97 | 460 |
| **supinated** | 0.97 | 0.97 | 0.97 | 815 |
| **acurracy** | | | 0.97 | 2036 |
| **macro avg** | 0.97 | 0.96 | 0.97 | 2036 |
| **weighted avg** | 0.97 | 0.97 | 0.97 | 2036 |

Source: Author.

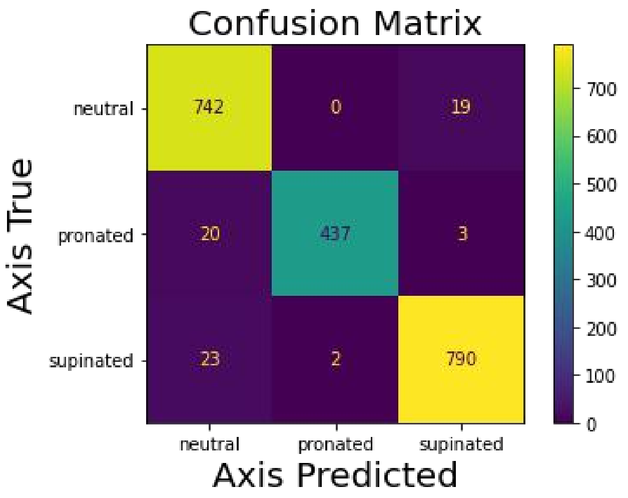



*Fig.17 - Confusion matrix according to RNC.*

Source: Author.

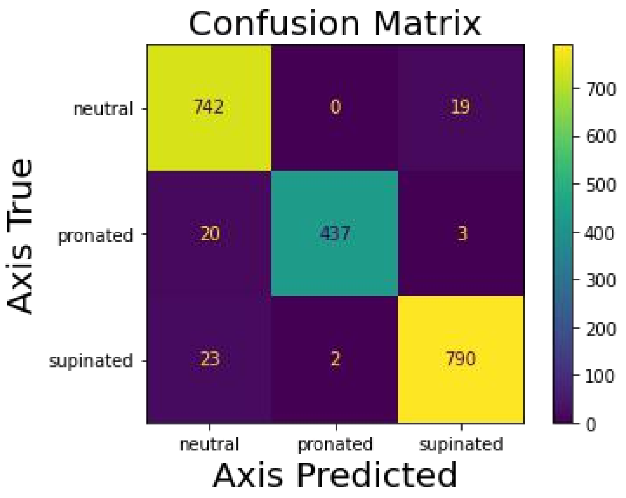



*Fig.17 - Confusion matrix according to SVM.*

Source: Author.

Note that of the 35% (2036 samples) of the test data, 742 were correctly classified as neutral, 437 were classified as correctly prone, and 790 were correctly classified as supine.

Using the SVM methodology, 65% of the data were used for training the algorithm, and 35% of the data for testing. As a result of the trained model, an accuracy of 96.70% was obtained.

Table 5 shows the results obtained, as well as figure 17, which shows the confusion matrix.

Table 5 - SVM metrics after model training and testing.

| | **Precision** | **Recall** | **f1-score** | **support** |
|---|---|---|---|---|
| **neutral** | **0.95** | **0.98** | **0.96** | **761** |
| **pronated** | **1.00** | **0.95** | **0.97** | **460** |
| **supinated** | **0.97** | **0.97** | **0.97** | **815** |
| **acurracy** | | | **0.97** | **2036** |
| **macro avg** | **0.97** | **0.96** | **0.97** | **2036** |
| **weighted avg** | **0.97** | **0.97** | **0.97** | **2036** |

Source: Author.

Note that of the 35% (2036 samples) of the test data, 742 were correctly classified as neutral, 437 were classified as correctly prone, and 790 were correctly classified as supine.

Using the Dummy Classifier methodology, 65% of the data were used for training the algorithm, and 35% of the data for testing. As a result of the trained model, an accuracy of 40.02% was obtained.

Table 6 shows the results obtained, as well as figure 18, which shows the confusion matrix.

Table 6 - SVM metrics after model training and testing.

| | **Precision** | **Recall** | **f1-score** | **support** |
|---|---|---|---|---|
| **neutral** | **0** | **0** | **0** | **761** |
| **pronated** | **0** | **0** | **0** | **460** |
| **supinated** | **0.40** | **1.00** | **0.57** | **815** |
| **acurracy** | | | **0.40** | **2036** |
| **macro avg** | **0.13** | **0.33** | **0.19** | **2036** |
| **weighted avg** | **0.16** | **0.40** | **0.23** | **2036** |

Source: Author.

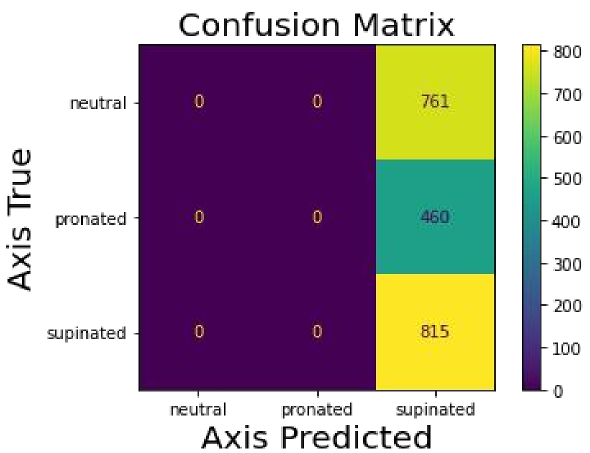



*Fig.18 - Confusion matrix according to the Dummy Classifier.*

Source: Author.

Note that of the 35% (2036 samples) of the test data, 0 were correctly classified as neutral, 0 were classified as correctly prone, and 815 were correctly classified as supine.

Using the Logistic Regression methodology, 65% of the data were used for training the algorithm, and 35% of the data for testing. As a result of the trained model, an accuracy of 98.33% was obtained.

Table 7 shows the results obtained, as well as figure 19, which shows the confusion matrix.

*Table 7 - Logistic Regression Metrics after model training and testing.*

| | Precision | Recall | f1-score | support |
|---|---|---|---|---|
| neutral | 0.98 | 0.99 | 0.98 | 761 |
| pronated | 1.00 | 0.97 | 0.98 | 460 |
| supinated | 0.98 | 0.99 | 0.98 | 815 |
| acurracy | | | 0.98 | 2036 |
| macro avg | 0.98 | 0.98 | 0.98 | 2036 |
| weighted avg | 0.98 | 0.98 | 0.98 | 2036 |

Source: Author.

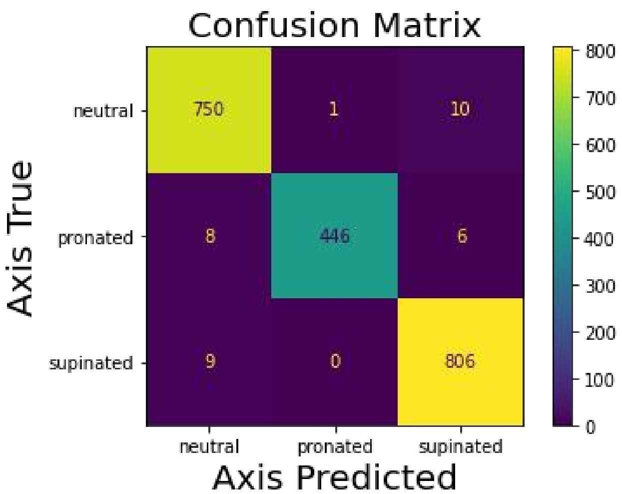



*Fig.18 - Confusion matrix according to Logistic Regression.*

Source: Author.

Note that of the 35% (2036 samples) of the test data, 750 were correctly classified as neutral, 446 were correctly classified as prone, and 806 were correctly classified as supine.

## III. CONCLUSION

After the tests using the Machine Learning and Deep Learning methodologies, we can conclude that, for the data set used, we obtained better answers, in descending order of accuracy, from the ANN methods (99.21%), Logistic Regression (98.33% ), KNN (98.23%), Random Forest (98.13%), RNC and SVM (96.70%), and Dummy Classifier (40.02%). This low value presented by the Dummy Classifier is justified in the description of the same in this work, in addition to this method not promoting satisfactory accuracy by comparing the samples with simple rules, without taking into account the relationship that the data sets can develop, this method of sorting should not be used on real data. The other methods shown had satisfactory accuracy, with values above 96% for all of them in identifying the type of step present in the test and training data.

## IV. ACKNOWLEDGEMENTS

This article is the result of the PD&I Smart IoT Gait Analysis System project, carried out by the University of the State of Amazonas (UEA), in partnership with Samsung Eletrônica da Amazônia Ltda., using resources from Federal Law nº 8.387/1991, being its dissemination and publicity in accordance with the provisions of article 39 of Decree No. 10.521/2020.

## REFERENCES

[1] ABEDI, M. NOROUZI, G-H. BAROUDI, A. (2012) Support vector machine for multi2021classification of mineral prospectivity areas. Comput Geosci 46:272–283. HTTPS://DOI.ORG/10.1016/J.CAGEO.2011.12.014

[2] 2021AREL, I.; ROSE, D.; KARNOWSKI, T. Deep machine learning 2021a new frontier in artificial intelligence research [research frontier]. Computational Intelligence Magazine, IEEE, v. 5, n. 4, p. 13–18, 2010

[3] 2021BENTLEY, JL. (1975) Multidimensional binary search trees used for associative searching. Commun ACM 18(9):509–517. https://doi.org/10.1145/361002.361007

[4] 2021BERAVS, T. et al. Development and validation of a wearable inertial measurement system for use with lower limb exoskeletons. 2011 11th IEEE-RAS International Conference on Humanoid Robots. Anais... In: 2011 11TH IEEE-RAS INTERNATIONAL CONFERENCE ON HUMANOID ROBOTS (HUMANOIDS 2011). Bled, Slovenia: IEEE, out. 2011Disponível em: http://ieeexplore.ieee.org/document/6100914. Acesso em: 4 out. 2022

[5] 2021Bishop C (2006) Pattern recognition and machine learning. Springer

[6] 2021BISHOP, C. (2006) Pattern recognition and machine learning. Springer

[7] 2021BREIMAN, L. (1996) Bagging predictors. Mach Learn 24(2):123–140. https://doi.org/10.1023/A:1018054314350
[8] 2021BREIMAN, L. (2001) Random forests. Mach Learn 45(1):5–32. https://doi.org/10.1023/A:
[9] 2021BREIMAN, L. FRIEDMAN, JH. OLSHEN, RA. STONE, CJ. (2017) Classification and regression trees. CRC Press. https://doi.org/10.1201/9781315139470
[10] 2021BREIMAN, Leo. Random Forest. Machine Learning, Statistics Department, University of California, Berkeley, CA 94720, 45, 5–32, 2001
[11] 2021Cannata A, Montalto P, Aliotta M,Cassisi C, Pulvirenti A, Privitera, E, Patanè, D(2011) Clustering and classification of infrasonic events at Mount Etna using pattern recognition techniques. Geophys J Int 185(1):253–264. https://doi.org/10.1111/j.1365-246X.2011.04951.x
[12] 2021COPELAND, B. R. Is Free Trade Good for the Environment? The American Economic Review, 2015.
[13] 2021David M Skapura, Building Neural Networks, ACM press, 1996
[14] 2021GÉRON, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems. 2nd edition. O'Reilly Media, 2019.
[15] 2021GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A.. Deep Learning. Cambridge: MIT Press, 2016. Disponível em: http://www.deeplearningbook.org.
[16] 2021GRACE, K., SALVATIER, J., DAFOE, A., ZHANG, B., EVANS, O. When Will AI Exceed Human Performance? Evidence from AI Experts., 2017.
[17] 2021GUIMARÃES, G. V.; et al. Pés: devemos avaliá-los ao praticar atividade físico esportiva? Revista Brasileira de Medicina do Esporte, v. 6, n. 2, p.57-59, 2000.
[18] 2021Huang C, Davis L, Townshend J (2002) An assessment of support vector machines
[19] 2021JORDAN, M. MITCHELL, T. (2015) Machine learning: trends, perspectives, and prospects. Science 349(6245):255–260. https://doi.org/10.1126/science.aaa8415
[20] 2021JURASZEK, Guilherme. RECONHECIMENTO DE PRODUTOS POR IMAGEM UTILIZANDO PALAVRAS VISUAIS E REDES NEURAIS CONVOLUCIONAIS. Dissertação (mestrado) – Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, Programa de Pós-Graduação em Computação Aplicada, Joinville, 2014.
[21] 2021LECUN, Y.; KAVUKCUOGLU, K.; FARABET, C. Convolutional networks and applications in vision. In: Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on. [S.l.: s.n.], 2010. p. 253–256.
[22] 2021LLOYD, S.P. Least squares quantization in PCM, IEEE Transactions on Information Theory. 1982, 28, 129–136.
[23] 2021MARTINEZ-MENDEZ, R.; SEKINE, M.; TAMURA, T. Detection of anticipatory postural adjustments prior to gait initiation using inertial wearable sensors. Journal of NeuroEngineering and Rehabilitation, v. 8, n. 1, p. 17, 2011.
[24] 2021Masotti M, Falsaperla S, LangerH, Spampinato S,Campanini R. (2006). Application of support vector machine to the classification of volcanic tremor at Etna, Italy.Geophys Res Lett 33(20). https://doi.org/10.1029/2006GL027441
[25] 2021Minsky, M.; Papert, S. Perceptrons: An Introduction to Computational Geometry; MIT Press: Cambridge, MA, USA, 1969.
[26] 2021ML | Dummy classifiers using sklearn. Geeksforgeeks, 2019. Disponível em: <https://www.geeksforgeeks.org/ml-dummy-classifiers-using-sklearn/>. Acesso em: 28 ago. 2022.
[27] 2021MURPHY, K. (2012) Machine learning. MIT Press
[28] 2021NASCIMENTO, Lucas. Dissertação (Mestrado em Engenharia Elétrica)2021Programa de Pós-Graduação em Engenharia Elétrica. Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2019.
[29] 2021NORRIS, C. M. Managing sports injuries: a guide for students and clinicians. 4. ed. Manchester: Elsevier Health Sciences, 2011. 432 p.
[30] 2021PEDAR SYSTEM. The pedar® system: The quality in-shoe dynamic pressure measuring system, 2019. Disponível em: <https://www.novelusa.com/index.php?fuseaction=systems.pedar>. Acesso em: 17 ago. 2022.
[31] 2021Petrelli M, Bizzarri R, Morgavi D, Baldanza A, Perugini D (2017) Combining machine learning techniques, microanalyses and large geochemical datasets for tephrochronological studies in complex volcanic areas: Newage constraints for the Pleistocene Magmatism of central Italy. Quat Geochronol 40:33–44. https://doi.org/10.1016/j.quageo.2016.12.003
[32] 2021Petrelli M, Caricchi L, Perugini D (2020) Machine learning thermo2021barometry: application to clinopyroxene-bearing magmas. J Geophys Res: Solid Earth 125(9). https://doi.org/10.1029/2020JB020130
[33] 2021Petrelli M, Perugini D (2016) Solving petrological problems through machine learning: the study case of tectonic discrimination using geochemical and isotopic data. Contribute Mineral Petrol 171(10). https://doi.org/10.1007/s00410-016-1292-2
[34] 2021Petrelli M, Perugini D, Moroni B, Poli G (2003) Determination of travertine provenance from ancient buildings using self-organizing maps and fuzzy logic. Appl Artif Intell 17(8–9):885–900. https://doi.org/10.1080/713827251
[35] 2021Preventing Data Leakage in Your Machine Learning Model. Available online: https://towardsdatascience.com/preventing-dataleakage-in-your-machine-learning-model-9ae54b3cd1fb (accessed on 1 August 2022).
[36] 2021RAZAK, A. H. A.; et al. Foot plantar pressure measurement system: a review. Sensors, v. 12, n. 12, p.9884-9912, 23 jul. 2012.
[37] 2021RODRIGUES, J. R.; et al. Influence of application of the inelastic taping pressure of runners pronators. Manual Therapy, Posturology & Rehabilitation Journal, v. 12, p.224-260, 2014.
[38] 2021ROZOS, Evangelos. KOUTSOYIANNIS, Demetris. MONTANARI, Alberto. KNN vs. Bluecat—Machine Learning vs. Classical Statistics. Hydrology 2022, 9,

[39] 2021Russell, S.; Norvig, P. Artificial Intelligence; Prentice-Hall: Upper Saddle River, NJ, USA, 2010.
[40] 2021SANTOS, C. P.; et al. Caracterização do sensor piezoelétrico para sua utilização em dispositivos embarcados. In: SEMINÁRIO DE ELETRÔNICA E AUTOMAÇÃO, 7., 2016, Ponta Grossa. Anais. Ponta Grossa: Utfpr, 2016. p. 1 20216.
[41] 2021SCHMIDT, D.; SILVA, D. B.; COSTA, C. A.; RIGHI, R. R.. Um Modelo de Predição de Mortalidade em Unidades de Terapia Intensiva Baseado em Deep Learning. Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS_CSBC), [S.l.], v. 18, n. 1/2018, july 2018.
[42] 2021SECRETÁRIO, J. H. A.; PIRES, R.. Uso de visão computacional para contagem automática de células em imagens obtidas por microscópios, IFSP – Câmpus São Paulo – 2018.
[43] 2021Shai S-S, Shai B.-D (2013) Understanding machine learning: from theory to algorithms, vol. 9781107057. Cambridge University Press. https://doi.org/10.1017/CBO9781107298019
[44] 2021SHU L, H. T.; WANG Y, L. Q.; FENG, D.; TAO, X. In-shoe plantar pressure measurement and analysis system based on fabric pressure sensing array. IEEE Trans. Inf. Technol. Biomed. 2009, 14. p.767-775.
[45] 2021SHU L, H. T.; WANG Y, L. Q.; FENG, D.; TAO, X. In-shoe plantar pressure measurement and analysis system based on fabric pressure sensing array. IEEE Trans. Inf. Technol. Biomed. 2009, 14. p.767-775.
[46] 2021SILVA, J. L. K. M. Análise da correlação de métodos de avaliação da pisada relacionada à ativação neuromuscular. 2015. 97 f. Dissertação 2021Pós-Graduação em Engenharia Biomédica, Universidade Tecnológica Federal do Paraná. Curitiba, 2015.
[47] 2021SILVA, J. M. S. C. F. Deteção de convulsões epiléticas em eletroencefalogramas usando Deep Learning. ISEP – Instituto Superior de Engenharia do Porto. 2017.
[48] 2021SOMVANSHI, Madan. CHAVAN, Pranjali. TAMBADE, Shital. SHINDE, S. V., "A review of machine learning techniques using decision tree and support vector machine," 2016 International Conference on Computing Communication Control and automation (ICCUBEA), 2016, pp. 1-7, doi: 10.1109/ICCUBEA.2016.7860040.
[49] 2021TECKSCAN. F-scan System, 2019. Disponível em: <https://www.tekscan.com/products-solutions/systems/f-scan-system>. Acesso em: 17 ago. 2022.
[50] 2021Tom Mitchell, Machine Learning, McGraw-Hill Computer science series, 1997
[51] 2021U.V Kulkarni, S.V Shinde, "Neuro –fuzzy classifier based on the Gaussian membership
[52] function",4th ICCCNT 2013,July 4-6,2013,Tiruchengode,India.
[53] 2021VIEIRA, Mário. Dissertação (Mestrado em Engenharia Elétrica) 2021Programa de Pós-Graduação em Engenharia Elétrica. Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2018.
[54] 2021Vikramaditya Jakkula , "Tutorial on Support Vector Machine" ,2013
[55] 2021VRECH, Giovani. SAITO, Kiyoshi, C MARA, Carlos. PROCESSAMENTO DE LINGUAGEM NATURAL NA RESOLUÇÃO DE PROBLEMAS DE CLASSIFICAÇÃO. Revista Ubiquidade, ISSN 2236-9031 – v.4, n.1 – jan. a jul. de 2021, p. 54
[56] 2021WAFAI, L.; et al. Identification of Foot Pathologies Based on Plantar Pressure Asymmetry. Sensors, v. 15, n. 8, p.20392-20408, ago. 2015.
[57] 2021WASZCZYSZYN, Z. (ed.), Neural Networks in the Analysis and Design of Structures Ed. Springer-Verlag Wien, 1999.
[58] 2021WEBSTER, J. G. The Measurement, Instrumentation and Sensors: Handbook. Boca Raton: Crc Press, 1999. 2588 p.
[59] 2021Zuo R, Carranza E (2011) Support vector machine: a tool for mapping mineral prospectivity. Comput Geosci 37(12):1967–1975. https://doi.org/10.1016/j.cageo.2010.09.014014311601100 40323101. https://doi.org/10.3390/hydrology90601011010933404324
[60] for land cover classification. Int J Remote Sens 23(4):725–749. https://doi.org/10.1080/