Classification of Respiratory Diseases Using the NAO Robot

Rafael Andrade Rodriguez, Jireh Ferroa-Guzman and Willy Ugarte[a]
Universidad Peruana de Ciencias Aplicadas, Lima, Peru

Keywords: Classification, NLP, NAO, Respiratory.

Abstract: This work proposes an interface that connects the NAO robot with a development environment in Azure Machine Learning Classic for the prediction of respiratory diseases. The developed code uses Machine Learning algorithms trained for the prediction of diseases and fatal symptoms in order to provide the user with a scope of his health status and the possible conditions associated with his age, sex, symptoms and severity. During this process, a brief discard of COVID-19 is made with the symptoms obtained, which indicates if they correspond to those of this disease. Additionally, we offer a friendly interaction with the NAO robot to facilitate the exchange of information and, at the end of the algorithm flow, it is always suggested to use a professional doctor to provide users with more details about their current status based on the overall results obtained. The tests carried out on the work show that it is possible to speed up the time of care in medical care centers in Peru through the Nao Robot. Additionally, it has been possible to predict respiratory diseases, which also helps the doctor to have a notion of the patient prognosis.

1 INTRODUCTION

On March 11, 2020, the World Health Organisation (WHO) declared SARS-Cov-2 a pandemic, due to its far-reaching, affecting millions of people in several countries around the world [1]. In Peru, COVID-19 was officially reported on March 6, 2020; Faced with this, the Peruvian state declared on March 15 of the same year, a state of Emergency, considering the speed of progression of the disease and ruled mandatory quarantine at the national level (Miyahira, 2020). On the 25th of the same month, the Peruvian Government established the measures that would lead citizens towards a new social coexistence and the state of emergency was extended due to the serious circumstances that affected the nation as a result of SARS-Cov-2 (Barrutia-Barreto et al., 2021)

Despite the measures taken, the numbers of deaths in Peru continued to grow. For the month of August 2020, Peru reached 613,378 infections, which made it the sixth country with the most reported cases. At that time, we reached 28 thousand deaths due to the pandemic with a mortality rate of 85.8 per thousand inhabitants[2]. During this period, there was evidence of a deficient response by the public health system of Peru, taking into account the number of deaths over the number of infections (Gianella et al., 2021). However, this deficiency did not begin with the COVID-19 pandemic.

In the public sector of the Peruvian health system, the government offers health services to uninsured people in exchange for the payment of a fee through the Integral Health System (SIS), with EsSalud being the entity that offers the services (Gianella et al., 2021). During previous years, complaints and even denunciations have been expressed by patients who have health insurance within this center. Problems such as speed of care, lack of medication and medical malpractice are part of the large list of claims against EsSalud. For example, in 2016, more than 111,000 claims for medical malpractice were filed in EsSalud for poor provision of services, which shows the dissatisfaction of users about the care provided[3].

According to the National Institute of Statistics and Informatics (INEI) of Peru, 25% of patients treated at EsSalud have to wait between 15 to 30 days to be able to schedule a simple medical consultation[3], while for surgical interventions, the time between the date of programming and the date of intervention rises to 2 months. On the other hand, for outpatient med-

[a] https://orcid.org/0000-0002-7510-618X

[1] WHO - https://covid19.who.int/

[2] "Peru has the world's highest COVID death rate. Here's why" - NPR - https://n.pr/3EBRukh

[3] "Complaints for lack of medical care and negligence persist in EsSalud" (in spanish) - https://bit.ly/2XAjVby

Andrade Rodriguez, R., Ferroa-Guzman, J. and Ugarte, W.
Classification of Respiratory Diseases Using the NAO Robot.
DOI: 10.5220/0011782700003411
In *Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2023)*, pages 940-947
ISBN: 978-989-758-626-2; ISSN: 2184-4313

ical care, the waiting time in a Peruvian clinic is 42 minutes on average, a figure that for Essalud rises to 81 minutes. Therefore, many people are discouraged from carrying out medical consultations in the public health sector, also considering other problems such as the level of distrust in medical personnel that still prevails to date. Synthesizing the main problems during consultations, it has been identified: insufficient attention time, high workload, patient anxiety or fear, fear of physical and verbal abuse, unrealistic expectations of patients, fear of demands, patient resistance to change and lack of training in this area. For all these reasons, there are still barriers between doctors and patients, which hinder the efficient exchange of information, which can have an impact on misdiagnoses, which currently cover 11% of cases in the country[4].

Within this context, it is important to consider these shortcomings in the health system, maintain an adequate doctor-patient relationship through patient training and information, encourage health prevention and promote adherence to treatments. For this, Peruvians need a source of information that serves as an assistant and allows them to foresee the presence of certain diseases based on probabilities. To meet these needs, an interface capable of delivering results with suggested illnesses and feedback on symptom severity to patients was developed through brief interaction with the robotic assistant, NAO. The algorithm that is handled in the interface was rigorously selected after performing a comparative analysis with other classification algorithms prioritizing accuracy and avoiding overtraining.

This paper is organized as follows. Therefor, in Section 2, an analysis of the state of the art considered for this work will be made. Section 3, first, introduces the technologies used for the development of the proposed solution.

Finally, Section 4 shows the experimental protocol, the results obtained, and the discussion. To conclude with Section 5

2 RELATED WORKS

The work of Fale (Fale and Gital, 2022) proposes a hybrid of Mamdani type and Fuzzy Sugeno type models by means of a fuzzy controller, follow a sequence of three steps: fuzzification; inference; and defuzzification. Yuan's work (Rozo et al., 2021) focuses on qualitatively detecting normal breathing and Cheyne-Stokes breathing in patients with non-contact heart failure using orthogonal frequency division and multiplexing technology (OFDM). On the other hand, Mubashir's work (Rehman et al., 2021) develops a machine learning (ML) classification model that is intelligent, secure, reliable and contributes to current health systems by exploiting several Machine Learning algorithms to classify eight respiratory anomalies: eupnea, bradypnea, tachypnea, Biot, sighs, Kussmaul, Cheyne-Stokes and central sleep apnea (CSA). All these works are oriented to respiratory diseases, just like ours. However, we use Multiclass Decision Jungle and Two Class Decision Forest as algorithms for prediction, unlike the other methods mentioned.

In (Romero-García et al., 2021) evaluates the performance of symptoms as a diagnostic tool for SARS-CoV-2 infection using Mantel-Haenszel logistic regression. In this area, in (Arslan, 2021), the authors develop a prediction method based on the similarity of the genome of human SARS-CoV-2 and a coronavirus similar to bat SARS-CoV to predict this same disease. Also, in (Brunese et al., 2020), the authors develop a supervised machine learning model that discriminates between COVID-19 and other lung diseases. These three works are based on the detection or prediction of COVID-19 and obtained the accuracies of 83.45%, 99.8% and 96.5% respectively according to the tests performed. Unlike them, our work simply performs a quick discard considering the symptoms mentioned by the user, without additional analysis.

In (Yoon et al., 2019), a deep learning system is developed using a recurrent neural network capable of encoding and deciphering people's postures in images and videos, and then being able to imitate them. Similar to this, in (Filippini et al., 2021), the authors design a CNN-based FER (Facial Expression Recognition) model for facial expression recognition in real-life situations. Both works handle neural networks and employ computer vision, unlike us, who mainly use the Audio service of the NAOqi library.

In (Burns et al., 2022), the authors attempt to prove that the walking speed of the humanoid NAO can be improved without modifying its physical configuration using decision trees and the ANN and NAive Bayes models. On the other hand, in (Hoffmann et al., 2021), the authors develop a process model with the components that are required to pass the recognition test in front of a mirror. Regarding our work, instead of working with decision trees, we manage multiple DAG's for disease prediction.

[4]"Medical error rate is around 11% in hospitals and technology could change this figure" - https://bit.ly/3V0PWHi

3 CLASSIFICATION TASK

3.1 Preliminary Concepts

3.1.1 Human-Robot Interaction

The study of human-robot interactions (also called HRI) represents a multidisciplinary field with contributions from human-computer interaction, artificial intelligence, robotics, natural language understanding, design, and social sciences.

1. Robot NAO: Nao is a programmable and autonomous humanoid robot developed by Aldebaran Robotics. Nao is 4.3 kg in weight and has a height of 58 centimeters. It is relatively light and small, which makes it an ideal solution to live with humans. Thanks to its prehensile finger hands with tactile sensors, it is capable of lifting objects of up to 600 grams. The different elements of NAO, such as sensors, motors and software are controlled by a powerful operating system called NAOqi. All versions have an inertial measurement unit with gyrometer, accelerometer and 4 ultrasound sensors, which provide the robot with stability, while the leg versions include 8 force detection resistors and 2 stops. The collaborative robot includes 4 microphones, 2 speakers and 2 high-definition cameras. Also, it presents interesting attributes and features such as a 25-degree movement, 2 HD cameras, 2 speakers, Wi-fi connection and an Intel Atom 1.6 GHz processor[5].
2. The robot has functionalities that the programmer can use as resources to automate processes with the NAO robot. In the interaction with the NAO robot, it is necessary to have a copy connected to a local IP of the home so that it can be connected to a remote computer in which it is going to be programmed. This process of connecting to a network and synchronizing in the working environment of the Python programming language is understood as “AlProxy”.
3. Naoqi: NAOqi is an interpreter between the Nao robot and Python programming that will allow us to interact with the robot. It consists of a framework that will allow to use the functionalities of the robot and implement the Machine Learning algorithm to the robot to process the received input data.
   - “ALProxy” command: This command is used as a means of communication between the programming interface and the NAO robot. Previously, the robot must be connected to the internet so that it can be recognized by the program. Its syntax consists of: an action to perform, the IP address where the robot is connected and the port where the NAO robot is connected.
   - Action Name: NAOqi’s own SDK comes with pre-programmed actions. All these pre-programmed actions can help the programmer perform processes or automations with the robot. From simple commands like saying something by voice to taking pictures and interpreting symbols. According to SoftBank Robotics (2022), these are separated by groups[5].

   (a) NAOQi core: Contains a list of functions that allows you to interact with the NAO robot to perform complex actions.
   (b) NAOqi sensors and led: Contains the action codes of the NAO robot with which it can interact and program.
   (c) NaoQi vision: Contains a library responsible for managing video cameras, stereo cameras and 2D cameras of the NAOqi robot.
   (d) NAOqi Audio: Contains modules for recording and playing audio, as well as for handling the robot’s language.
   (e) NAOqi people perception: Contains commands are used to analyze human behavior around the robot.
   (f) Naoqi Motion: Contains the commands that allow the movement of the NAO robot.

   From these functionalities of the NAOqi library, we take advantage of ALSpeechRecognition, a NAOqi Audio command, which allows to interpret the sounds or words that a human can make. In this way, the NAO robot is able to capture the input data necessary for symptom processing and also, the user information that will be used to send the results.

3.1.2 Basic Notions About Health and Symptomatology

1. Symptomatology: Set of symptoms characteristic of a given disease or grouping of symptoms that occur in a patient.
2. Comorbidity: The presence of two or more associated disorders or diseases in the same person, occurring at the same time or one after the other[6].

[5] “Programming NAO robot with Python” - Softbank Robotics Europe (2015) -https://www.youtube.com/watch?v=iAeis7j5LmE

[6] Co-morbidities - WHO - https://www.who.int/southeastasia/activities/co-morbidities-tb

3. Interconsultation: Occurs when a doctor refers the patient to another specialist doctor to handle communication with different areas of expertise.

3.1.3 Classification Models

1. Multiclass Decision Jungle: Represents an extension or modification of the Decision Forest algorithm. However, this consists of a set of acyclic graphs that are driven by a decision (DAG). Muticlass Decision Jungle has the following advantages:
   (a) By allowing tree branches to merge, a decision DAG typically takes up less memory space and has better generalization performance than a decision tree, albeit at the cost of somewhat longer training time.
   (b) Decision jungles are nonparametric models that can represent nonlinear decision boundaries.
   (c) They perform a selection and classification of built-in features and are resistant in the presence of noisy features.

   The algorithm has many advantages in terms of machine learning and has had considerable success when testing. However, this also has a fundamental limitation that, with a lot of data, the number of nodes in decision trees will grow exponentially with depth, limiting their use to only certain platforms that can support this amount of processing.

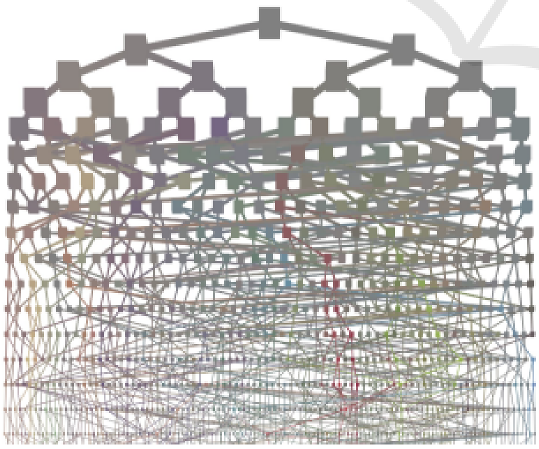


Figure 1: DAG visualization (Shotton et al., 2013).

2. Two Class Decision Forest: A decision forest describes a model made of multiple decision trees. The prediction of a decision forest is the aggregation of the predictions of each decision tree.

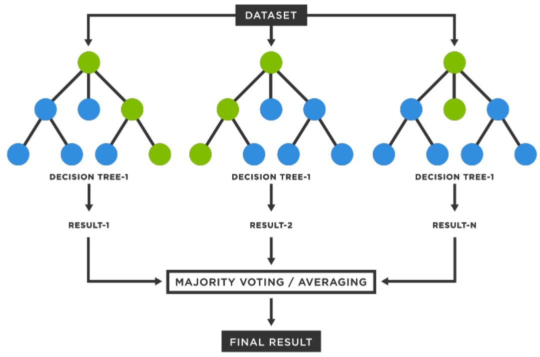


Figure 2: Decision Forest flow. (Hänsch, 2021).

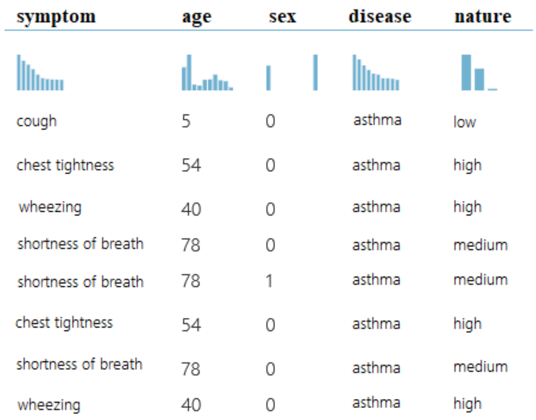
3.2 Method

Our main contribution is the prediction of diseases based on symptoms using a Multiclass Decision Jungle. This algorithm is powered by a dataset published on the Kaggle platform by American Health Info[7]. For its adaptation in the Azure environment, we have made some improvements such as data normalization; That is, those symptoms that were written differently, but corresponded to the same, were unified into one. Also, because the dataset was unbalanced; in other words, with unrepresented classes, we aggregate records with missing severities (high, medium, or low) for each symptom of each disease; so that the code can predict diseases with any level of severity of their symptoms. To exemplify this idea, if there are no records of people with disease: asthma and symptom: chest tightness, specifically with severity: low, it is assumed that no user with these last two characteristics can have asthma.

From the dataset handled, we use "Diseases", as a column to predict, and the columns of: "symptom", "sex", "age" and "severity" as predictive variables. We also use this dataset with an additional column of Mortality, which indicates whether a symptom considering age and severity, can become fatal. To do this, the code sends the data to another development environment and tells the user if it detects several deadly symptoms.

On the Azure Machine Learning Classic platform, we first load the normalized dataset and section the columns relevant to our algorithm. Then, we use 88% of the dataset for processing and 12% for testing. Records for both proportions are random. Then, we generate the training model, which connects with the multiclass decision jungle algorithm. This has at the beginning of 16 DAG's *(directed acyclic graphs)*, with a maximum depth of 156 DAG's and

[7]https://www.kaggle.com/datasets/abbotpatcher/respiratory-symptoms-and-treatment



symptom	age	sex	disease	nature
cough	5	0	asthma	low
chest tightness	54	0	asthma	high
wheezing	40	0	asthma	high
shortness of breath	78	0	asthma	medium
shortness of breath	78	1	asthma	medium
chest tightness	54	0	asthma	high
shortness of breath	78	0	asthma	medium
wheezing	40	0	asthma	high

Figure 3: Image of the updated Dataset.

width of 140 DAG's. Subsequently, we create the Score Model, which contains the percentages calculated with each test record selected for the testing process and is part of the basis for knowing the accuracy of the algorithm in general. Finally, the Evaluate model shows us the metrics calculated on the same platform and we connect the input and output services to make our architecture functional.

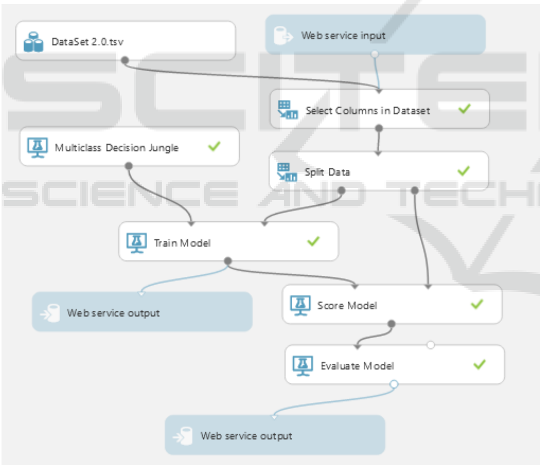


Figure 4: Architecture in Azure Machine Learning.

According to the flow, in the first instance, the user indicates his personal data to the robot: Name, Age, Sex, Symptoms and Severities; for the latter variables, an iterative flow is traversed that detects when there are 3 or more symptoms of COVID-19. In case the detection is positive, the user is consulted if a discard test was done; If so, continue with the code and otherwise, the flow is cut off indicating that a test is carried out as soon as possible. After completion of symptom and mortality uptake; The NAO robot sends the captured symptoms to the first development environment in Azure, which will process one by one and save the individual results, to then be averaged and thus, calculate those diseases with more probability. After this process, it connects with the second developmental environment and comments are issued regarding the mortality of the symptoms. Finally, the user is queried for their WhatsApp number and, after having captured it, the overall results are sent. During the entire process of connection to the development environments, it is necessary to execute a reconnection script with the NAO robot to connect to a network that has access to the internet; which is required to receive results from Azure. Subsequently, we link again with the robot so that it continues its flow.

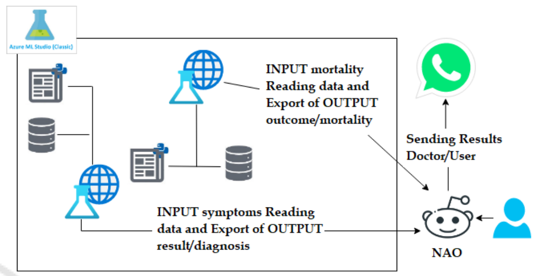


Figure 5: Connection Flow with the environments.

This developed flow has allowed us to speed up and have a shorter attention time, thanks to the fluid conversation facilitated by the humanoid robot. Additionally, during the development and tests carried out, we have been able to observe certain characteristics of the robot that allow the correct reception of the input data. These are as follows:

1. When talking to the NAO: We have to position ourselves at a considerable height (higher than that of the robot), because the microphones are located on the upper front of the robot (Fig. 1)
2. Response time: When the robot makes a query, you have to wait approximately 3 seconds to respond to it.
3. Detection of the person: As seen in Fig. 1, the sensors are located in the chest of the robot, which means that it will detect the person in front of it, therefore, it is recommended that, when talking to the robot, the patient is in front of it, Not on the sides.

Likewise, the symptoms have been classified in order to simplify the process of receiving input data so that the user can mention those symptoms recorded that fit their own during the consultation. The classifications are as follows:

1. Cough Related
2. Related to breathing
3. Pains

4. Weight loss
5. Fatigue related
6. General discomfort

On the limitations and scope of our proposal, we have the following:

1. Our model recognizes a total of 65 symptoms and 18 different respiratory diseases.
2. Our model is limited to respiratory diseases.
3. We must consider a waiting time to be able to give an answer to each question asked by the robot. This time is 5 seconds.
4. The flow does not include adding new symptoms or diseases (conditioned by the dataset).

The use of the NAO robot for our proposal, unlike any other technological solution, is justified by relying on one of the causes of our main problem: the level of distrust that exists towards medical personnel, which hinders an efficient exchange of information with patients. Faced with this, a study entitled "Making eye contact with a robot: Psychophysiological responses to eye contact with a human and with a humanoid robot" concludes, after an experimental process, that in human-robot and human-human condition, eye contact in front of the deviated gaze caused a greater conductance of the skin, responses associated with positive affect and deceleration of the heart, that index attention allocation. In conclusion, eye contact provokes affective and attentional reactions when shared with a humanoid robot as well as with another human.

4 EXPERIMENTATION

In this section, we present the experimental analysis to demonstrate the feasibility of the proposal. We will outline our experimental process and explain all the considerations that were fundamental to obtain the final results.

4.1 Experimental Protocol

To carry out the experiments, various tools have been used, both hardware and software. First, the NAO v.6 humanoid robot enabled interaction with users as well as data capture in the experimental process. The programming language of Python 2.7 and 3.10 has been handled, along with the Naoqi framework. Regarding our algorithms in Azure Machine Learning classic, we have made use of the web service. On the other hand, the computer that was responsible for executing the developed code has an AMD Ryzen 5 2500U processor with 2.00 GHz and 24.0 GB of RAM.

Within our experimentation we have also tested with different algorithms to be able to know which of them would be better to be able to predict the diseases that we are going to enter. Among these algorithms are the Multiclass Decision Jungle, Multiclass Decision Forest, Multiclass Logitic Regression and Multiclass Neural Network. Additionally, these algorithms have internal parameters that we have been modifying in order to find the most appropriate model for the prediction we want. In Table 1, we can find the results of the algorithms with the default values given by the Azure Machine Learning Classic; and in Table 1b you can find the results of the same algorithms with the modified values. In Table 1c you can find the experimentation that has been carried out for the part of the algorithm that predicts the mortality of the disease.

Our code is currently available at https://github.com/gareia/Dr_Nao.git.

4.2 Results

In this section, we will detail the results of the tests performed and show some videos in which the workflow and each functionality mentioned below are visualized. The complete flow of our proposal can be visualized in our video[8].

Reception Input Data. The reception of name is conditioned by a dataset, which initially overloaded the vocabulary allowed by Naoqi. Other input values such as Age and Sex are received without major problems; However, in the symptoms section, at times the mentioned symptom is not recognized. Here, several factors such as ambient noise or similarity in the name of different symptoms can influence.

Connection to Development Environment. The API Key and URL generated by Azure Machine Learning classic allow us to connect with the development environment. This is functional at all times and the disconnection/reconnection script with NAO, as well as the network with defined internet access, works properly without errors.

Discard COVID-19: The flow is properly met. When symptoms related to COVID-19 are detected, the user is asked if the COVID-19 test was performed to end the flow in case of responding negatively.

Reception and Sending of Results. It was possible to send results through a WhatsApp account linked to the selected browser. It is recommended that it is

[8]https://youtu.be/sw7LpUie2TA

Table 1: Machine Learning Algorithms Metrics Charts.

(a) Default Values.

	Multiclass Decision Jungle	Multiclass Decision Forest	Multiclass Logitic Regression	Multiclass Neural Network
Overall accuracy	.820627	.805957	.914870	.907313
Average accuracy	.980070	.978440	.990541	.989701
Micro-averaged precision	.820627	.805957	.914870	.907313
Macro-averaged precision	.861564	.838173	.934235	.918068
Micro-averaged recall	.820627	.805957	.914870	.918068
Macro-averaged recall	.760552	.768652	.896562	.903236

(b) Modified values.

	Multiclass Decision Jungle	Multiclass Decision Forest	Multiclass Logitic Regression	Multiclass Neural Network
Overall accuracy	.817070	.805957	.918649	.907535
Average accuracy	.979674	.978440	.990961	.989726
Micro-averaged precision	.817070	.805957	.918649	.907535
Macro-averaged precision	.862696	.838173	.937841	.918490
Micro-averaged recall	,817070	.805957	.918649	.907535
Macro-averaged recall	.755444	.768652	.899862	.901638

(c) Mortality prediction models.

	Two Class Bayes Point Machine	Two Class Averaged Perceptron	Two Class Boosted Decision Tree	Two Class Decision Forest
Accuracy	.962	.273	.273	.988
Precision	.951	.273	.273	.984
Recall precision	.907	1.000	1.000	.972
F1 Score	.929	.430	.430	.978
Threshold	.500	.500	.500	.500
AUC	.992	.164	.000	.999

already open and that it has synchronized correctly to avoid unnecessary delays.

According to Tables 1 and 1b, and an analysis carried out, for the detection of diseases, we have chosen the Multiclass Decision Jungle as an algorithm for the detection and diagnosis of diseases. On the other hand, according to Table 1c, for the detection of mortality we have selected the Two Class Decision Forest.

4.3 Discussions

The reason we chose the Multiclass Decision Jungle over the other classification algorithms is justified in the precision matrix. The matrix, as shown in *Figure 7*, with the lowest amount of empty blocks is beneficial for the proposal, since it rules out fewer diseases during detection and favors obtaining more real results. In addition to that, it is the one that has the best parameters and results at the time of experimentation, not having an overtraining or having too low values.

The reason we chose the Two Class Decision Forest is that, over the other classification algorithms we have tested, this is the one that has returned us better results. This, because it has a better Accuracy and Precision that will help to obtain better results from our algorithm. Additionally, it is better than the Two Class Averaged Perceptron and Two Class Boosted Decision Tree algorithms, as these return very low values to be selected. Regarding the Two Class Decision Forest, it obtained metrics quite similar to the Two Class Bayes Point Machine, so any of these would have been useful.

Initially, we want to recognize the user's name by assigning a Dataset of names; However, the number

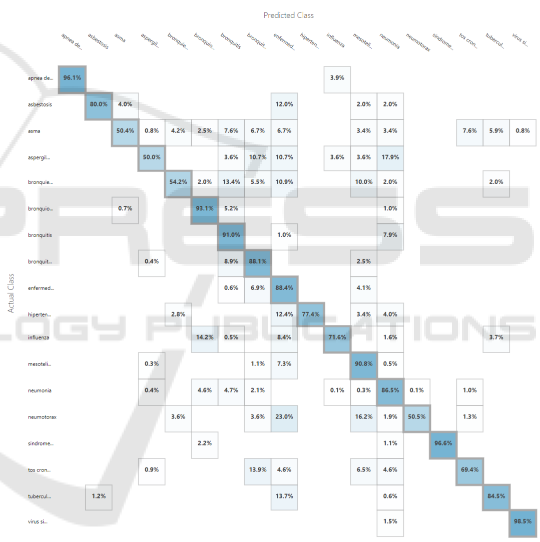


Figure 6: Multiclass Decision Jungle Precision Matrix.

of records exceeded 40,000 names, which overloaded the vocabulary of the robot and began to present errors and slowness during testing. From there, we limit the number of names to recognize and avoid overloading the vocabulary with a lot of data. Among those that occupy the most space are combinations of numbers and names. Subsequently, we make improvements so that, before receiving each input data, the vocabulary is configured to be more limited and less overloaded.

5 CONCLUSIONS

First, the Nao has technical tools that have very been useful, however, it has limitations in terms of Speech

Recognition. Additionally, it has sensitivity to noise, that is, if in the place where we are there is too much noise around, the Nao will not be able to detect the voice of the person or may have difficulties. Testing the code was difficult at this stage because it's not possible to test with Choreography if input data in audio format is needed. Also, the person cannot be at a great distance from the microphones of the Robot (which are in his head), otherwise, the listening of this will be low and may have problems to understand the message. According to this, we noticed that the Nao robot has a greater facility to capture numbers when listening to them, than large words. You are more likely to ask to repeat the word than to ask to repeat the dictated number. Another limitation is related to Internet access, because if it were possible to access the Internet connected to the Nao robot, our full flow time will be severely reduced. Finally, special care must be taken when training with the classification algorithms, because some datasets can generate overtraining, which would generate irregular results (Burga-Gutierrez et al., 2020).

For future improvements, more diseases can be added to the dataset so that it can cover a larger field and can be run again with the same algorithm. Although our premise is that the input data is said aloud, our flow time can decrease if the patient, instead of dictating the symptoms one by one, can have a table with the total symptom numbers and tell the numbers to the robot. Similar to this, the reception of the telephone number generates that the time flow increases considerably. We suggest that this data can be typed and the results can be sent to the doctor if applicable. Additionally, by skipping this step, we can avoid scaring the patient, because we do not know how sensitive he may be and may even misinterpret the robot's comments on the results. Furthermore, combining our approach with other kinds of smart health allocation systems (Ugarte, 2022).

REFERENCES

Arslan, H. (2021). COVID-19 prediction based on genome similarity of human sars-cov-2 and bat sars-cov-like coronavirus. *Comput. Ind. Eng.*, 161.

Barrutia-Barreto, I., Sánchez-Sánchez, R. M., and Silva-Marchan, H. A. (2021). Consecuencias económicas y sociales de la inamovilidad humana bajo covid – 19 caso de estudio perú. *Lecturas de Economía*, 1(94).

Brunese, L., Martinelli, F., Mercaldo, F., and Santone, A. (2020). Machine learning for coronavirus covid-19 detection from chest x-rays. *Procedia Computer Science*, 176.

Burga-Gutierrez, E., Vasquez-Chauca, B., and Ugarte, W. (2020). Comparative analysis of question answering models for HRI tasks with NAO in spanish. In *SIMBig*.

Burns, R. B., Lee, H., Seifi, H., Faulkner, R., and Kuchenbecker, K. J. (2022). Endowing a NAO robot with practical social-touch perception. *Frontiers Robotics AI*, 9.

Fale, M. I. and Gital, A. Y. (2022). Dr. flynxz - A first aid mamdani-sugeno-type fuzzy expert system for differential symptoms-based diagnosis. *J. King Saud Univ. Comput. Inf. Sci.*, 34(4).

Filippini, C., Perpetuini, D., Cardone, D., and Merla, A. (2021). Improving human-robot interaction by enhancing NAO robot awareness of human facial expression. *Sensors*, 21(19).

Gianella, C., Gideon, J., and Romero, M. J. (2021). What does covid-19 tell us about the peruvian health system? *Canadian Journal of Development Studies / Revue canadienne d'études du développement*, 42(1-2).

Hänsch, R. (2021). *Handbook of Random Forests - Theory and Applications for Remote Sensing*. Series in Computer Vision. WorldScientific.

Hoffmann, M., Wang, S., Outrata, V., Alzueta, E., and Lanillos, P. (2021). Robot in the mirror: Toward an embodied computational model of mirror self-recognition. *Künstliche Intell.*, 35(1).

Miyahira, J. (2020). Lo que nos puede traer la pandemia. *Revista Medica Herediana*, 31(2).

Rehman, M., Shah, R. A., Khan, M. B., Shah, S. A., Abuali, N. A., Yang, X., Alomainy, A., Imran, M. A., and Abbasi, Q. H. (2021). Improving machine learning classification accuracy for breathing abnormalities by enhancing dataset. *Sensors*, 21(20).

Romero-García, R., Martínez-Tomás, R., Pozo, P., de la Paz, F., and Sarriá, E. (2021). Q-CHAT-NAO: A robotic approach to autism screening in toddlers. *J. Biomed. Informatics*, 118.

Rozo, A., Buil, J., Moeyersons, J., Morales, J. F., van der Westen, R. G., Lijnen, L., Smeets, C., Jantzen, S., Monpellier, V., Ruttens, D., Hoof, C. V., Huffel, S. V., Groenendaal, W., and Varon, C. (2021). Controlled breathing effect on respiration quality assessment using machine learning approaches. In *IEEE CinC*.

Shotton, J., Sharp, T., Kohli, P., Nowozin, S., Winn, J. M., and Criminisi, A. (2013). Decision jungles: Compact and rich models for classification. In *NIPS*.

Ugarte, W. (2022). Vaccination planning in peru using constraint programming. In *ICAART*.

Yoon, Y., Ko, W., Jang, M., Lee, J., Kim, J., and Lee, G. (2019). Robots learn social skills: End-to-end learning of co-speech gesture generation for humanoid robots. In *IEEE ICRA*.