

Health Seek: A Deep Learning-Based Intelligent System to Aid Medical Diagnosis

Manuel López, Antonio Peñalver 

Center of Operations Research, Miguel Hernández University, Elche, Spain

Correspondence to: Antonio Peñalver, a.penalver@umh.es

Keywords: Diagnosis, Artificial Intelligence, Deep Learning, Stroke, Intelligence Systems

Received: November 10, 2021

Accepted: January 26, 2022

Published: January 29, 2022

Copyright © 2022 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

ABSTRACT

Medical diagnosis is one of the most tedious and complex processes that healthcare personnel face in their day-to-day life. To establish an adequate treatment, it is essential to carry out a correct and early evaluation of each patient. Occasionally, given the number of tests that need to be performed, this evaluation process can require a significant amount of time, and can negatively affect the patient's recovery. The objective of this work is the development of a new software that, using Artificial Intelligence (AI), offers the healthcare professional support in the process of diagnosing the patient, as well as preventing the probability of suffering a certain disease, based on test information analytics and demographic information available. The system allows storing multiple models based on Deep Learning (DL), previously trained for the diagnosis of different diseases. These models allow predictions to be made based on available medical information. As a use case, one of these models has been successfully tested in diagnosing stroke events.

1. INTRODUCTION

Medical diagnosis is one of the fundamental tasks of physicians and essential and effective treatment. The diagnosis is based on the analysis of reliable data, for which the final conclusions will only be valid if they are based on exact notions and precise facts. There are a series of premises or basic principles related to the diagnostic process [1]:

- Doctor-patient relationship: essential to obtain the information the doctor needs. If this relationship is adequate, it is possible to provide the patient relief and security.
- Anamnesis: 50% to 75% of diagnoses are made through question-answers.
- Physical examination: complements the anamnesis, since the physical signs are objective and verifiable marks of the disease, which represent solid and indisputable facts.
- Association of symptoms and signs: it is very common for doctors to try to group symptoms and

signs to make the diagnosis.

This process usually requires the accomplishment of different medical tests, such as: blood tests, X-rays, cardiograms, MRIs, etc. Carrying out these tests can be time consuming, which can sometimes be crucial for the correct recovery of the patient.

Nowadays, the large amount of data generated every day and the computing capabilities of the new equipment, as well as the rise and expansion of AI generates a huge amount of application possibilities. Medicine is one of the disciplines in which these advances can be applied with promising results.

Artificial Neural Networks (ANN) are a Machine Learning (ML) and processing paradigm inspired by the functioning of the human nervous system. ANN are made up of a set of neurons interconnected by links. Each neuron takes as inputs the outputs of the neurons of the predecessor layers, each of these inputs is multiplied by a weight, then the partial results are added, and the output is calculated by means of an activation function. This output is in turn the input of the neuron it precedes.

The union of all these interconnected neurons is what makes up the ANN. In this sense, ANN are nothing more than massively interconnected networks in parallel of simple and hierarchically organized elements, which try to interact with objects in the real world in the same way that the biological nervous system does. These methods do not require explicit theory but accurate models and training examples to obtain the most suitable network architecture for a specific problem [2].

The main objective of this work is to develop a software system that can be used by any medical center, as a support tool in the process of medical diagnosis and the management of patient records. With this aim in mind, the developed system allows to store multiple DL-based models, previously trained for the diagnosis of different diseases that allows making predictions from the medical information available.

To design the architecture of the system and to be able to test its correct operation, an ANN has been designed and trained for the prediction of Stroke Disease. According to the World Health Organization (WHO), stroke events are the 2nd leading cause of death globally, responsible for approximately 11% of total deaths. Stroke disease can usually be identified by blood tests, brain imaging (CT, MRI, and X-ray), ECG and EEG and other physiological neurological methods such as Induced Potential Tests [3].

The rest of the paper is organized as follows: in Section 2, we review some other projects related to the application of AI to medical information. Then, Section 3 is devoted to explaining both the Neural Network model training and the architecture of the developed system. Then, in Section 4, we show the results of the evaluation of the tool after a period of test. Last, Section 5 summarizes conclusions and further work.

2. RELATED WORK

A series of previous works that apply ANN and Expert Systems (ES) to the problem of making a diagnosis are detailed below. All of them are focused on the treatment of specific diseases.

In [4], two Multilayer Perceptrons (MP), based on regional CortexID Z-scores, were trained in order to classify different types of dementia and map with MMSE score. The proposed architecture obtains good classification accuracy comparing with physicians' diagnosis using both clinical and laboratory data.

Another ANN-based solution is described in [5]. The system allows to identify three main heart diseases: mitral stenosis, aortic stenosis, and ventricular septal defect. It can also deliver assistance for doctors to make advanced heart diagnosis using real medical data.

In [6], authors developed an ES for quick diagnosis and detection of *Staphylococcus Aureus* bacteria on human skin to help doctors in the treatment of staph infectious diseases. The inference engine of the system provides the answer based on doctors' request using the developed user interface.

The project described in [7] is a binary classification system constructed by means of an ANN to analyze whether a person under study is analyzed to be Parkinson disease patient or not. The classifier uses 16 biological quantities to obtain a classification success rate of around 80%.

In [8] authors presented a novelty system for fully automatic segmentation of multiple sclerosis lesions from brain tissues using and entropy-based Expectation Maximization (EM) algorithm and Markov

random fields. The proposed method estimates a gaussian mixture model with three kernels as cerebrospinal fluid (CSF), normal tissue and multiple sclerosis lesions providing better results than previous methods in the field.

Momentum Contrast (MoCo) [9] is an AI-based project that uses clinical data such as chest X-rays in order to aid researchers and physicians to predict which patients are at highest risk for deterioration caused by COVID-19.

In recent years, many ANN-based studies have analyzed the problem of diagnosing the possibility of suffering a stroke event. In [10] Chin *et al.* applied data augmentation to CT images of ischemic stroke patients to expand the number of patch images, using them as input to CNN models to detect ischemic stroke with great accuracy. In [11] authors proposed a Res-CNN model that automatically classified acute ischemic stroke in MRIs. The Res-CNN model solved the performance degradation problem using the residual unit.

Dourado *et al.* [12] presented framework for the classification of stroke from CT images applying Convolutional Neural Networks (CNN) in order to identifying a healthy brain, an ischemic stroke or a hemorrhagic stroke. Following the Transfer Learning concept CNN was combined with different consolidated Machine Learning methods such as Bayesian Classifier, Multilayer Perceptron, k-Nearest Neighbor, Random Forest and Support Vector Machines with high accuracy.

Recently, in [13] authors proposed a methodology based on DL models on raw EEG data to predict stroke disease. The model was trained with data collected from real-time electroencephalography (EEG) sensors and used different neural architectures like LSTM and others. Our approach does not use graphical information, but only laboratory biological quantities, obtained from patient analytics and clinical data.

Unlike the works cited above, the system developed in this paper allows the storage of multiple ANN-based models, to aid in the diagnosis of multiple diseases, which is, in our opinion, an important advantage over current systems. The system would even allow the use of some of the models based on neural networks mentioned above that use numerical or categorical variables as input data from laboratory tests or clinical and/or demographic data of the patient. Next sections are devoted to describing the developed system.

3. SYSTEM DESCRIPTION

This section is devoted to describing the architecture of Health Seek, as well as the development of each of the four components that comprise it: a multi-platform mobile application, an ANN for the prediction of cerebrovascular infarcts, a database that stores models and all medical information collected from patients and an API that acts as a connecting link between all components.

For the development of the first functional prototype of the system, an ANN-based model has been designed and trained that provides a estimation of the probability that a specific patient may suffer a stroke event from a set of known biological quantities in a certain moment.

The data is requested from the mobile application, this request is collected by the API to later process the request. For instance, when a physician requests the statistics of a patient, the API will act as follows:

- The App request is collected and processed.
- A query is constructed and performed to obtain all diseases registered in the database.
- For each disease, the corresponding model is loaded from the database and then the patient's data is collected according to the features that a particular model needs.
- The ANN performs the prediction from the recorded data.
- The results are then returned to the App interface and showed to the user.

Once the different components that make up the system have been discussed, **Figure 1** shows a graphic representation of how these components are related to each other.

3.1. Neural Network Architecture

The development of the neural network has been carried out using Python, a high-level, object-oriented

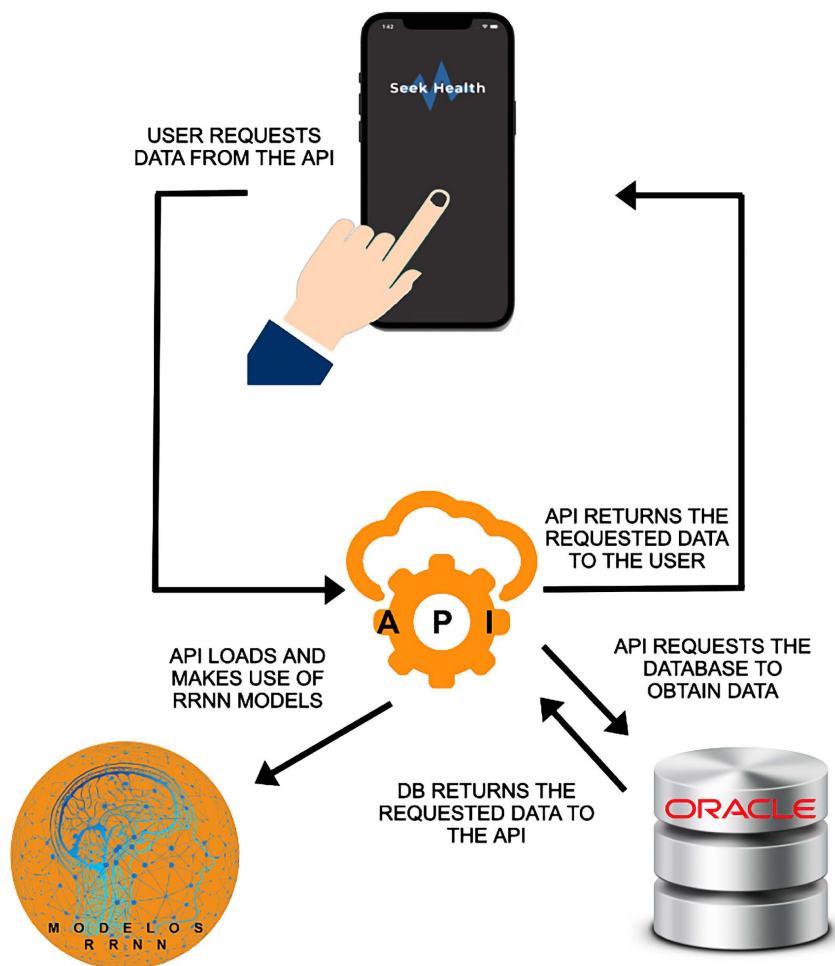


Figure 1. System infrastructure health seek.

and interpreted programming language. One of the most important benefits of Python is that both the standard library and the interpreter are freely available and there are many libraries that facilitate the creation of neural network architectures, such as Pytorch or TensorFlow. For this specific work, TensorFlow has been used under the Keras high-level library, which provides speed and ease in the development of applications of this type.

For the design and evaluation of the system, an ANN to predict whether a patient is likely to get stroke has been first designed and then trained using a dataset [14] consisting of 4996 patients with the features showed in **Figure 2**. The dataset collects information about demographic and laboratory analytics information for patients in order to predict the risk of suffering a cerebral stroke using different parameters like gender, age, previous diseases, or smoking status among others. The system is easily scalable with new ANN-based models for other diseases that are previously trained and stored in the application database.

The main objective of using an ANN is the elaboration of a model that allows making predictions, in terms of probability, about whether certain patients will develop a cerebral stroke. To do this, a Multi-Layer Perceptron (MLP) has been first trained and then used as a binary classifier. To improve the precision of the results obtained, different architectures and combination of input variables have been tested.

Before the training, it was necessary to carry out a cleaning and data preparation process, since the dataset contained several records with null values in some features, such as the "bml" feature. The average value of the rest of non-null records was used as value for "bml" feature.

Attribute information

- 1) Id: unique identifier
 - 2) gender: "Male", "Female" or "Other"
 - 3) age: age of the patient
 - 4) hypertension: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
 - 5) heart_disease: 0 if the patient doesn't have any heart disease, 1 if the patient has a heart disease
 - 6) ever_married: "No" or "Yes"
 - 7) work_type: "children", "Govt_job", "Never_worked", "Private" or "Self-employed"
 - 8) Residence_type: "Rural" or "Urban"
 - 9) avg_glucose_level: average glucose level in blood
 - 10) bml: body mass index
 - 11) smoking_status: "formerly smoked", "never smoked", "smokes" or "Unknown"
 - 12) stroke: 1 if the patient had a stroke or 0 if not
- *Note: "Unknown" in smoking_status means that the information is unavailable for this patient

Figure 2. List of features used in the model.

In addition, there was a significant problem with unbalanced data that needed to be resolved. **Figure 3** shows the problem of unbalanced data, with a proportion of non-disease cases (Stroke = 0) much higher than cases with disease (Stroke = 1). This fact caused the model to not achieve adequate performance, generating overfitting. To solve this problem, a new sample of 350 records with zero stroke and 249 records with a value of one was obtained by applying a subsampling technique [15].

After the data cleaning process was completed, the dataset was decomposed into three different subsets: test, validation, and train. This was followed by a study of the useful features those that improved the predictive performance. For this, a Correlation Matrix [16] was used, which represents the correlations between pairs of variables for a specific patient in the dataset. This method requires all the features to be numeric, but as **Figure 4** shows, the dataset had some categorical features that needed to be fixed.

The process of converting categorical variables to numeric can be done in different ways:

- Convert the original variable to a binary one, with only two possible values, as in the case of the "sex" variable.
- Dummify the variable to obtain a list of numbers, each corresponding to a particular category.
- Assign each possible category a different numerical value.

In the case of features with two possible values, such as "sex" and "ever_married", they were converted to a binary variable. The rest of the categorical features were blurred. Generally, the process of dummifying a variable tends to provide better results in the behavior of the neural network, although this is not always the case. Once all the features were fixed, the correlation matrix was calculated. **Figure 5** shows the correlation matrix for the dataset used.

In view of the correlation matrix, it can be observed that the features that seemed to be more related to the target variable were "sex", "hypertension", "heart_disease", "avg_glucose_level" and "ever_married".

Next, the ANN architecture that best fit the data was designed. Previously, several experiments were carried out where different architectures were tested: layers were added and/or deleted, the number of neurons in each layer was modified, dropout was added, the activation function was changed, the learning rate was altered, different features were added and deleted, etc. After this process, the appropriate network architecture for the specific problem was obtained. Results obtained after training with the features chosen from the correlation matrix are showed in **Figure 6**.

After several experiments were conducted, it was found that there were features that only contributed

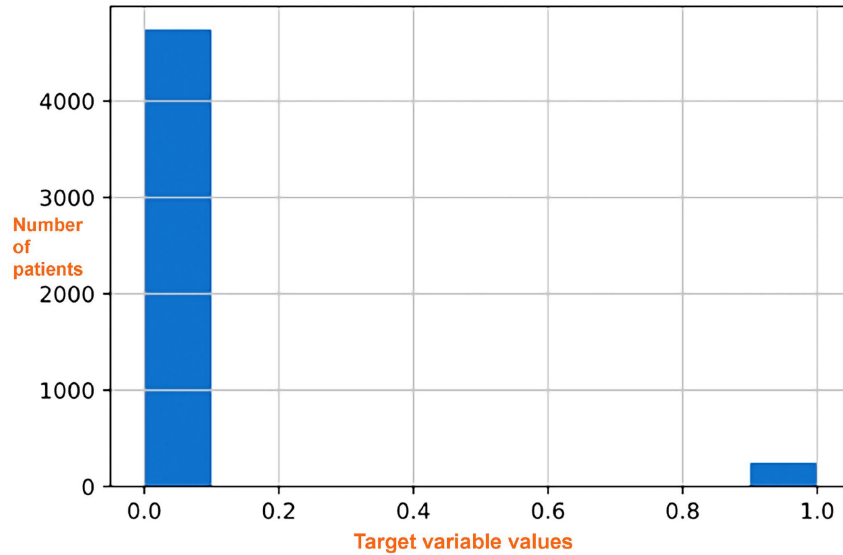


Figure 3. Initial distribution of the target variable (stroke).

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

Figure 4. Structure of the dataset with the features considered.

	sex	age	hypertension	heart_disease	ever_married	avg_glucose_level	stroke	work_type_Cuenta propia	work_type_f
sex	1.000000	0.057112	-0.009366	0.141582	0.073601	0.119851	0.052152	-0.062259	
age	0.057112	1.000000	0.285346	0.280065	0.574830	0.283928	0.542299	0.291432	
hypertension	-0.009366	0.285346	1.000000	0.061913	0.093616	0.172973	0.247019	0.170978	
heart_disease	0.141582	0.280065	0.061913	1.000000	0.093368	0.255815	0.259166	0.009695	
ever_married	0.073601	0.574830	0.093616	0.093368	1.000000	0.195714	0.231581	0.159786	
avg_glucose_level	0.119851	0.283928	0.172973	0.255815	0.195714	1.000000	0.282338	-0.001767	
stroke	0.052152	0.542299	0.247019	0.259166	0.231581	0.282338	1.000000	0.142732	
work_type_Cuenta propia	-0.062259	0.291432	0.170978	0.009695	0.159786	-0.001767	0.142732	1.000000	
work_type_Funcionario	-0.039776	0.095166	0.043436	-0.016665	0.093353	0.020359	-0.032120	-0.194832	1.000000
work_type_Nunca	0.004854	-0.557086	-0.120885	-0.094924	-0.501161	-0.099051	-0.198904	-0.136184	
work_type_Privado	0.075779	-0.015779	-0.107144	0.052632	0.064105	0.038593	0.008356	-0.612106	
residence_type_Rural	0.081513	0.009868	0.030808	-0.015924	0.057724	0.018900	-0.057837	0.046992	
residence_type_Urbano	-0.081513	-0.009868	-0.030808	0.015924	-0.057724	-0.018900	0.057837	-0.046992	
smoking_status_Desconocido	0.034918	-0.301477	-0.190415	-0.045328	-0.287181	-0.096719	-0.119200	-0.118877	
smoking_status_Nunca	-0.124038	0.097453	0.117743	-0.092034	0.022470	-0.004314	-0.020314	0.072328	
smoking_status_fuma	0.084146	-0.005772	0.034349	0.106201	0.103767	0.035243	-0.011193	-0.022524	
smoking_status_fumaba	0.032985	0.212842	0.030452	0.059987	0.184664	0.076304	0.163853	0.062010	

Figure 5. Correlation matrix after pre-processing the dataset.

noise to the model. To solve this problem, the Principal Component Analysis (PCA) algorithm [17] was applied and it was found out that the best results were obtained using only five features. Finally, the most relevant features providing better performance for the model are shown in Table 1. Then, in Table 2, the final network architecture is described. Figure 7 shows the model accuracy and model loss after the training

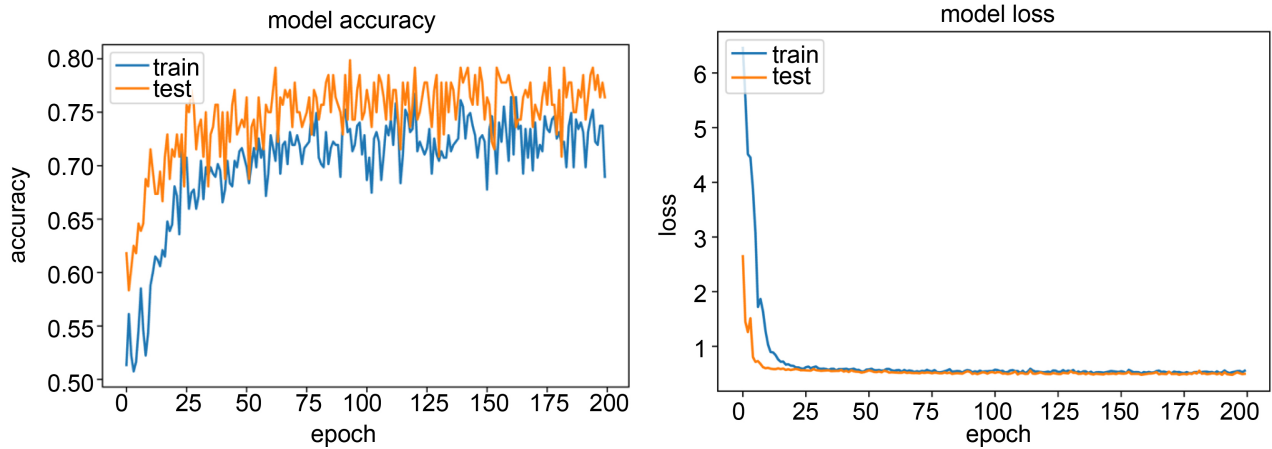


Figure 6. Results obtained after training with the features chosen from the correlation matrix.

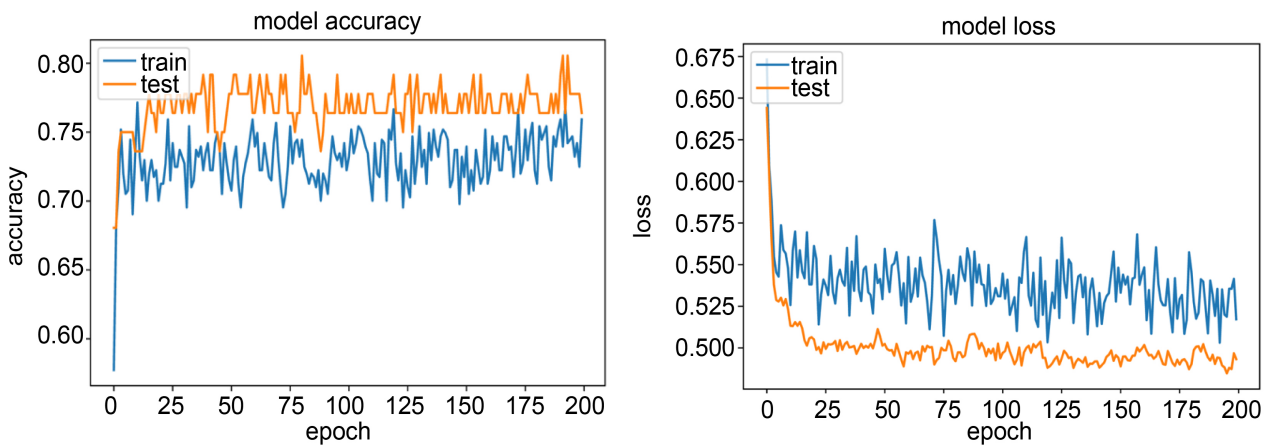


Figure 7. Model accuracy and loss for the final architecture.

Table 1. Most relevant characteristics.

Features providing best results
sex
age
hypertension
heart_disease
avg_glucose_level

process using the specified architecture for the model.

For the evaluation of the model, a sample of 120 patients extracted from the dataset was used. It should be noted that, as shown in [Figure 3](#), the values of the target variable of the dataset are binary values (0 and 1). In our case, the model presented yields an output in terms of probability, where the cut-off has been set at 50%. Then, the results obtained were evaluated and compared with those expected. [Tables 3-5](#) show the evolution of the results with the different models tested and described above.

For model 3, the same features and architecture were used as in model 2 but applying the PCA algorithm.

Table 2. Final neural network architecture.

Optimizer	Adam
Hidden layers	1 hidden layer with 9 neurons
Activate hidden layer function	Relu
Dropout grade	0.20 (in hidden layer and input layer)
Learning rate	0.001
Batch size	10
MaxNorm	3
Epochs	200
Validation Split	0.15
Output layer activation function	sigmoid

Table 3. Features (correlation matrix) and architecture of model 1.

Features providing best results
age
hypertension
heart_disease
ver_married
avg_glucose_level

Optimizer	Adam
Hidden layers	1 hidden layer with 9 neurons
Activate hidden layer function	Relu
Dropout grade	0.20 (in hidden layer)
Learning rate	0.001
Batch size	10
MaxNorm	3
Epochs	200
Validation Split	0.15
Output layer activation function	sigmoid

As can be seen in [Table 5](#), model 1 provides a 77.5% success rate. For model 2, it was observed that there were characteristics that introduced noise to the model, so we carried out different tests in an empirical way, adding and eliminating characteristics to the model. With the characteristics shown in [Table 4](#), it was possible to increase up to 80% of success. With respect to model 3, using the same structure and characteristics as in model 2, we applied the PCA algorithm, testing with different numbers of components, obtaining the best result with 4 components, reaching 82% accuracy. In general terms, it can be observed

Table 4. Features (testing various features) and architecture of model 2.

Features providing best results	
	sex
	age
	hypertension
	heart_disease
	avg_glucose_level
Optimizer	Adam
Hidden layers	1 hidden layer with 9 neurons
Activate hidden layer function	Relu
Dropout grade	0.20 (in hidden layer and input layer)
Learning rate	0.001
Batch size	10
MaxNorm	3
Epochs	200
Validation Split	0.15
Output layer activation function	sigmoid

Table 5. Evaluation of the models used.

MODEL 1		
	Predicted yes	Predicted no
True yes	40	6
True no	21	53
Accuracy		77.5%
MODEL 2		
	Predicted yes	Predicted no
True yes	37	10
True no	14	59
Accuracy		80.0%
MODEL 3 (PCA)		
	Predicted yes	Predicted no
True yes	40	7
True no	15	58
Accuracy		82.0%

that PCA improves true negatives and reduces false positives and negatives, achieving better accuracy in prediction capacity.

3.2. Mobile App

For the development of the mobile application, the open-source Flutter SDK was used, under the Dart programming language. This language was developed by Google and is used to create mobile, desktop, back-end, and web applications. Dart is an object-oriented language, defined by classes, using a C-style syntax, although in some respects it is reminiscent of JavaScript, python, and Java. On the other hand, Flutter is an SDK also developed by Google in order to create multi-platform mobile applications. The main advantages of Flutter are: Native compilation for both Android and iOS with high performance, very flexible creation of graphical user interfaces and fast development.

As user interface, it was decided to use a mobile application, developed with multi-platform tools, as minimal changes in the coding, allow us to obtain versions for the main current mobile platforms: iOS Android or even, a web-based version.

The user interface was designed to be attractive, intuitive, and easy to use by healthcare and non-technical personnel. This first version of the application has focused on the role of the doctor, allowing them to view patient data, consult the results of laboratory analytics and find out the possibility of a patient developing a certain disease for which the system has a pre-trained model (cerebral stroke in this case).

Figure 8 shows some of the application screens, which illustrate part of the implemented functionality. Starting from left to right you can see the user login screen, the profile information assigned to the current user, the patient search screen, the current patient information form, including the results of the last laboratory tests and the estimation of the probability of suffering a disease for a particular patient. Furthermore, the application allows to define different colors depending on the range of probability of suffering a disease obtained.

3.3. System Database

The design of the application's database is a crucial aspect of the system, as it must allow a straightforward storage of different models to aid in the diagnosis of different diseases. The database must also store medical centers, users, patients, and appropriate clinical information for each of the diseases that will be later used by the system as input for the ANN. Patient information includes demographic data such as sex, age, location, height, weight, etc. and data from various types of laboratory tests such as imaging,

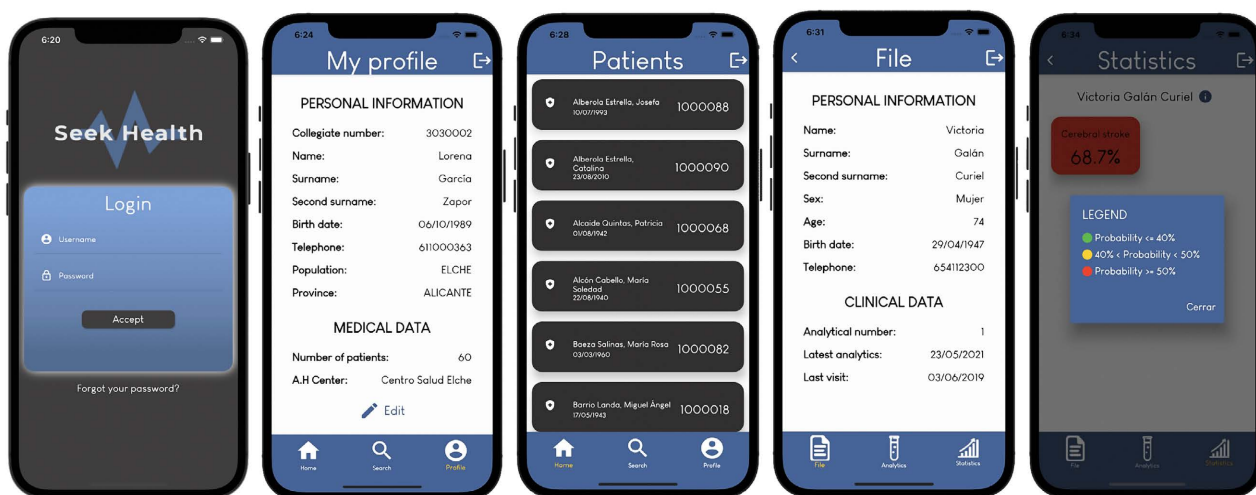


Figure 8. Some screens of the mobile app.

blood, and urine tests, etc. **Figure 9** shows the Entity-Relationship model of the database with main tables and relations of the system.

The Database Management System (DBMS) used in this project has been oracle, mainly because it is one of the most widely used DBMS and allows compatibility with numerous clinic and hospital systems, as well as the level of security it offers. In this case, the data to be saved are (among others) clinical data of patients, so special attention must be given to safety.

3.4. API REST

The language used for API programming has been Python again, on the Flask framework. This framework allows developing web applications, web services in a very simple and agile way. The fact that the API is also developed in Python makes integration with neural network models more natural. Communication between the API and the mobile application is done under the https protocol. The mobile application sends https requests, the API processes those requests and returns the result in JSON format.

The developed system includes a REST API that acts as a middleware between the mobile application and the database. This scheme allows the application to be easily adapted to different types of devices and allows working with different ANN-based models stored in the database, regardless of the input variables used in each particular model. Each and every one of the implemented functions follow the flow diagram showed in **Figure 10**.

Table 6 shows the different methods included in the developed API, the input parameters and a description of their functionality:

4. EVALUATION

This section is devoted to describing the experiments performed to verify the correct operation of the system. We conducted several experiments with five patients of different ages, in order to evaluate the accuracy of the results obtained and the evolution of the prediction of each patient in developing a stroke event based on the demographic values and laboratory tests performed.

To do this, making use of analytical records made by patients over the years, the most relevant five features obtained through PCA, used by the model as described in Section 3.1 Neural Network Architecture have been extracted. The data used for each patient, as well as the predictions made by the model are

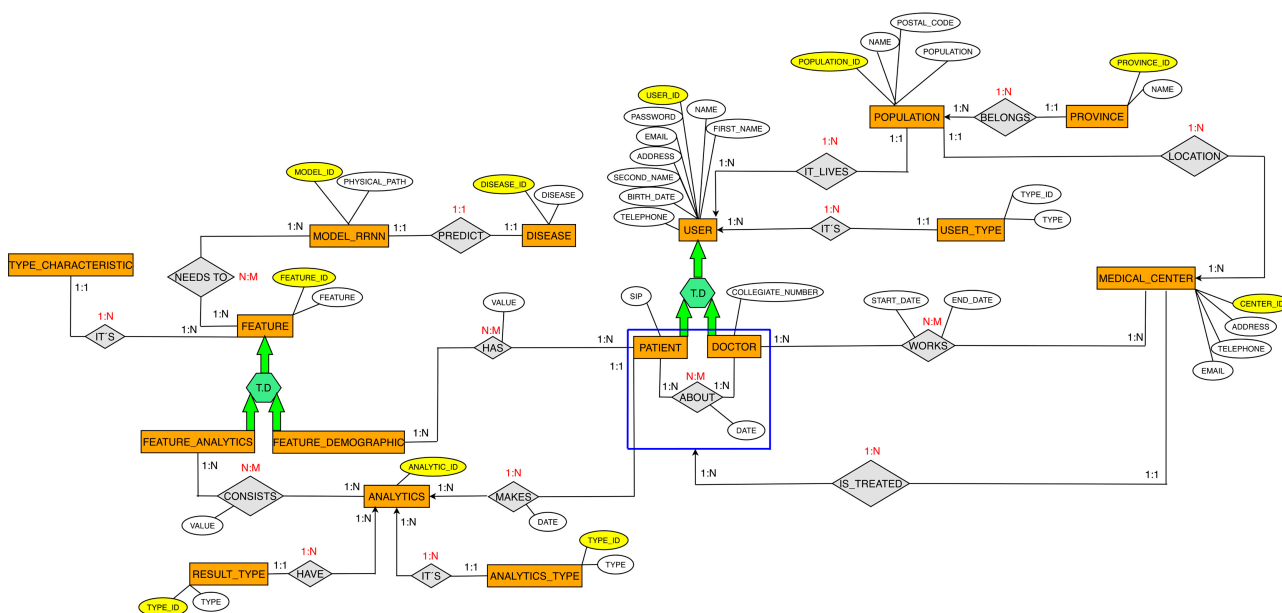


Figure 9. Entity-relationship design of the database.

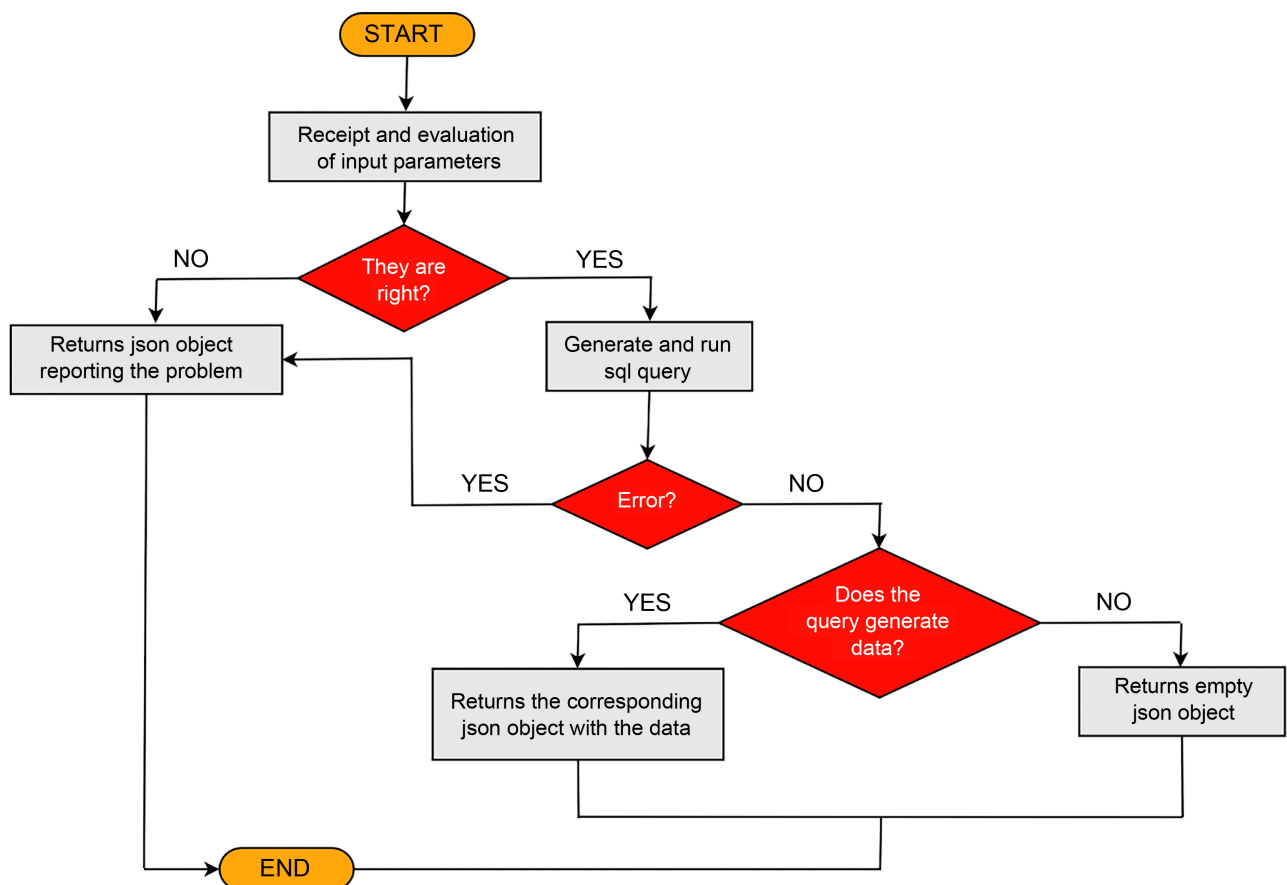


Figure 10. API methods flowchart.

shown in **Table 7**. For each of the five patients, the values of three laboratory tests performed over time as well as Sex and Age have been considered. The names of the patients have been anonymized and are therefore fictitious.

Figure 11 shows the evolutionary graphs for each of the patients in **Table 7**. The results obtained by the model can be observed, as well as the trend over the years of each patient in the probability of development of the disease.

Although the objective of the present work was not to achieve high accuracy in the estimation of the model, the results show that the estimation performed by our Neural Network is consistent with the evolution of the data provided as input to the model. In any case, as discussed in the previous sections, the designed system is scalable to different diseases or even models, including those of the state of the art discussed in Section 2: Related Work, as long as they are based on using Artificial Neural Network models.

5. CONCLUSIONS AND FURTHER WORK

In this work, a new system based on AI that allows to store multiple pre-trained ANN models to aid in the diagnosis of different diseases has been proposed. This first functional prototype described consists of a multi-platform application with a user interface adapted to be used in mobile devices, a database for the storage of the models, a first ANN model with a suitable architecture for the prediction of cerebral strokes and a rest API that acts as a middleware between the user interface and the database and allows the system to be easily scaled to new diseases and models.

Although this first version of the tool only includes one model, the experiments carried out show that the system fulfills the purpose for which it was designed and encourages us to include new models and

Table 6. API methods.

METHOD	PARAMETERS	DESCRIPTION
/file	Patient id	Returns the data (personal and medical) of the patient specified as a parameter.
METHOD	PARAMETERS	DESCRIPTION
/doctor	Doctor id	Returns the data of the physician specified as parameter.
METHOD	PARAMETERS	DESCRIPTION
/statistics	Patient id	Using the patient's identifier included as a parameter, it retrieves the previously trained models from the database and returns a prediction of the probability of suffering a certain disease for each of them.
METHOD	PARAMETERS	DESCRIPTION
/analytical_values	Analytic id	Returns the values of a specific laboratory analysis based on its identifier
METHOD	PARAMETERS	DESCRIPTION
/analytics	Patient id	Given the id of a patient, this method returns the laboratory tests stored in the database ordered by most recent date.
METHOD	PARAMETERS	DESCRIPTION
/patients	Doctor Id. Filter [Optional]	If the filter parameter is not specified, the method returns all patients associated with the doctor whose identifier matches the one specified as a parameter. If the filter parameter is specified, a search is made for patients in the database
METHOD	PARAMETERS	DESCRIPTION
/login	Nickname, Password	This method is used to authenticate a user, using their username and password

diseases in later versions. The experiment results show that the estimation performed by our ANN Network is consistent with the evolution of the data provided as input to the model.

As future work, it is intended in the first place to develop and add new models for other diseases. Specifically, we are working on a model for the early detection of urinary tract infection to determine whether it is necessary to prescribe antibiotics before the results of the culture performed on the patient are available, which could significantly improve the recovering of the patient. Secondly, a graphical user interface is being developed for the creation and training of new ANN models, with the aim of simplifying as much as possible the process of incorporating a new model into the system.

Table 7. Evolution of the biological and demographic quantities of the patients and prediction of disease obtained.

Héctor		INPUT DATA				MODEL OUTPUT	
date analytic	Sex	age	hypertension	heart_disease	Avg_glucose_level (mg/dl)	Prediction (%)	
01/2017	Male	53	No	No	113.21	32.75	
04/2018	Male	54	Yes	No	120.78	63.10	
07/2020	Male	56	Yes	Yes	160.05	69.56	
Raúl		INPUT DATA				MODEL OUTPUT	
date analytic	Sex	age	hypertension	heart_disease	Avg_glucose_level (mg/dl)	Prediction (%)	
02/2014	Male	66	No	No	81.10	38.08	
03/2019	Male	71	No	Yes	102.36	69.59	
02/2021	Male	73	Yes	Yes	98.6	76.11	
Carla		INPUT DATA				MODEL OUTPUT	
Analytic Date	Sex	age	hypertension	heart_disease	Avg_glucose_level (mg/dl)	Prediction (%)	
03/2021	Female	55	Yes	Yes	210.40	66.43	
06/2021	Female	55	No	Yes	190.47	63.26	
09/2021	Female	55	No	Yes	180.40	63.08	
Toni		INPUT DATA				MODEL OUTPUT	
Analytic Date	Sex	age	hypertension	heart_disease	Avg_glucose_level (mg/dl)	Prediction (%)	
02/2011	Male	78	No	No	78.03	41.01	
05/2017	Male	84	No	Yes	80.12	41.91	
06/2021	Male	88	No	Yes	94.64	65.87	
Ángela		INPUT DATA				MODEL OUTPUT	
Analytic Date	Sex	age	hypertension	heart_disease	Avg_glucose_level (mg/dl)	Prediction (%)	
05/2019	Female	71	Yes	No	195.25	60.62	
06/2020	Female	72	Yes	No	180.63	61.21	
08/2021	Female	73	Yes	No	161.21	62.10	



Figure 11. Model output for patients.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest regarding the publication of this paper.

REFERENCES

1. Díaz Novás, J., Gallego Machado, B. and León González, A. (2006) El diagnóstico médico: Bases y procedimientos. *Revista Cubana de Medicina General Integral*, **22**, 11 p. <http://scielo.sld.cu/pdf/mgi/v22n1/mgi07106.pdf>
2. Goodfellow, I., Bengio, I. and Courville, A. (2016) Deep Learning (Adaptive Computation and Machine Learning series). The MIT Press, Cambridge.
3. Lee, K.H. (2002) Diagnosis of Cerebrovascular Disease. *Journal of the Korean Medical Association*, **45**, 1432-1439. <https://doi.org/10.5124/jkma.2002.45.12.1432>
4. See, E.S.K. and Yeung, D.W.C. (2019) The Use of Neural Network Analysis of Brain ¹⁸F-FDG PET in Diagnosis of Dementia Subjects. *Journal of Biomedical Science and Engineering*, **12**, 111-120. <https://doi.org/10.4236/jbise.2019.122009>
5. Ghwanmeh, S., Mohammad, A. and Al-Ibrahim, A. (2013) Innovative Artificial Neural Networks-Based Decision Support System for Heart Diseases Diagnosis. *Journal of Intelligent Learning Systems and Applications*, **5**, 176-183. <https://doi.org/10.4236/jilsa.2013.53019>
6. Uka, K.K., Oguoma, S.I., Chukwu, C.A. and Emele, C.I. (2020) Adoption of Artificial Intelligence for Diagnosis and Treatment of *Staphylococcus aureus* Infections Disease on Humans. *E-Health Telecommunication Systems and Networks*, **9**, 1-15. <https://doi.org/10.4236/etsn.2020.91001>
7. Liao, Y.L. (2020) Applying Neural Network in Classifying Parkinson's Disease. *Journal of Computer and Communications*, **8**, 19-23. <https://doi.org/10.4236/jcc.2020.810003>
8. Bijar, A., Khanloo, M.M., Peñalver Benavent, A. and Khayati, R. (2011) Segmentation of MS Lesions Using Entropy-Based EM Algorithm and Markov Random Fields. *Journal of Biomedical Science and Engineering*, **4**, 552-561. <https://doi.org/10.4236/jbise.2011.48071>
9. Sriram, A., *et al.* (2021) COVID-19 Prognosis via Self-Supervised Representation Learning and Multi-Image

Prediction.

<https://ai.facebook.com/blog/new-ai-research-to-help-predict-covid-19-resource-needs-from-a-series-of-x-rays/>

10. Chin, C.L., Lin, B.J., Wu, G.R., Weng, T.C., Yang, C.S., Su, R.C. and Pan, Y.J. (2017) An Automated Early Ischemic Stroke Detection System Using CNN Deep Learning Algorithm. *Proceedings of the 2017 IEEE 8th International Conference on Awareness Science and Technology*, Taichung, 8-10 November 2017, 368-372. <https://doi.org/10.1109/ICAWSST.2017.8256481>
11. Liu, L.L., Chen, S.W., Zhang, F.H., Wu, F.X., Pan, Y. and Wang, J.X. (2019) Deep Convolutional Neural Network for Automatically Segmenting Acute Ischemic Stroke Lesion in Multi-Modality MRI. *Neural Computing and Applications*, **32**, 6545-6558. <https://doi.org/10.1007/s00521-019-04096-x>
12. Dourado, C.M., da Silva, S.P.P., da Nobrega, R.V.M., Barros, A.C.D.S., Reboucas Filho, P.P. and de Albuquerque, V.H.C. (2019) Deep Learning IoT System for Online Stroke Detection in Skull Computed Tomography Images. *Computer Networks*, **152**, 25-39. <https://doi.org/10.1016/j.comnet.2019.01.019>
13. Choi, Y.A., Park, S.J., Jun, J.A., Pyo, C.S., Cho, K.H., Lee, H.S. and Yu, J.H. (2021) Deep Learning-Based Stroke Disease Prediction System Using Real-Time Bio Signals. *Sensors*, **21**, Article No. 4269. <https://doi.org/10.3390/s21134269>
14. Fedesoriano. Stroke Prediction Dataset. <https://www.kaggle.com/fedesoriano/stroke-prediction-dataset>
15. Gonzalez, P.L. (2019) Conjunto de datos desbalanceados. Aprende IA. <https://aprendeia.com/conjunto-de-datos-desbalanceado/>
16. Mokhtar, E. (2020) La matriz de Correlación en Python. <https://likegeeks.com/es/matrix-correlacion-python/>
17. Jolliffe, I.T. (2002) Principal Component Analysis. Springer Series in Statistics. Springer, New York..