FACTA UNIVERSITATIS
Series: **Mechanical Engineering** Vol. 17, N^o 3, 2019, pp. 285 - 308
https://doi.org/10.22190/FUME190327035A

Original scientific paper

RESULTS AND CHALLENGES OF ARTIFICIAL NEURAL NETWORKS USED FOR DECISION-MAKING AND CONTROL IN MEDICAL APPLICATIONS

Adriana Albu, Radu-Emil Precup, Teodor-Adrian Teban

Politehnica University Timisoara, Dept. Automation and Applied Informatics, Romania

Abstract. *The aim of this paper is to present several approaches by which technology can assist medical decision-making. This is an essential, but also a difficult activity, which implies a large number of medical and technical aspects. But, more important, it involves humans: on the one hand, the patient, who has a medical problem and who requires the best solution; on the other hand, the physician, who should be able to provide, in any circumstances, a decision or a prediction regarding the current and the future medical status of the patient. The technology, in general, and particularly the Artificial Intelligence (AI) tools could help both of them, and it is assisted by appropriate theory regarding modeling tools. One of the most powerful mechanisms that can be used in this field is the Artificial Neural Networks (ANNs). This paper presents some of the results obtained by the Process Control group of the Politehnica University Timisoara, Romania, in the field of ANNs applied to modeling, prediction and decision-making related to medical systems. An Iterative Learning Control-based approach to batch training a feedforward ANN architecture is given. The paper includes authors' concerns in this domain and emphasizes that these intelligent models, even if they are artificial, are able to make decisions, being useful tools for prevention, early detection and personalized healthcare.*

Key Words: *Artificial Neural Networks, Medical Diagnosis, Medical Prediction, Prosthetic Hands, Recurrent Neural Networks*

Received March 27, 2019 / Accepted July 15, 2019

Corresponding author: Radu-Emil Precup
Politehnica University Timisoara, Department of Automation and Applied Informatics, Bd. V. Parvan 2, 300223 Timisoara, Romania
E-mail: radu.precup@upt.ro

1. INTRODUCTION

Decision-making and control in medical domain aim to improve the healthcare system and to help physicians by assisting them and offering suggestions or a second opinion. Several reasons that justify and encourage the use of these mechanisms should be mentioned here:

- there are medical processes that can be modeled by these artificial intelligent tools, with relevant benefits for the patients,
- some medical conditions are hardly detected by humans; a suggestion or an alert that is provided by an automated system could be a solid argument for a physician, and,
- there is a large number of simple and routine activities that are time consuming and that overload medical staff; these could be easily performed by a machine.

Artificial Intelligence (AI) domain provides a series of techniques, approaches and algorithms that can be used to solve problems which require an intelligent behavior. In order to do this, it is necessary to understand the way humans think and act, and then to build intelligent tools able to reproduce humans' functionalities. One of the key properties of the human brain activity is learning. Although, different kinds of systems having this property have been developed in recent years, the Artificial Neural Networks (ANNs) are still one of the most popular and efficient forms of learning systems [1]. They follow the structure of the human brain, using a simplified architecture, which is made of basic processing units (artificial neurons), interconnected and working in parallel.

ANNs can be used to solve various problems of classification and generalization. Their properties (the ability to execute parallel distributed computation, to tolerate noisy inputs, to learn new associations, new patterns and new functional dependencies, according to [1] and [2]) make them suitable for applications that can be developed to support medical activity. The most common issues in this domain are related to decision-making area, where the problems are difficult to be solved using traditional computation. ANNs are able to supplement the processing power of the digital computers with the ability to make sensible decisions and to learn by ordinary experience, as humans do [2].

The justification of ANNs is not difficult. AI domain contains more of the other mechanisms that have been used to create automated systems for medical decision-making (e.g. knowledge-based systems and statistical inference, which seem to be the most wide-spread). The authors of the present paper have been interested in applying them to medical systems [3], [4] and modeling with focus on medical [5], [6] and non-medical process control [7]–[14] including novel training algorithms formulated in the framework of Iterative Learning Control (ILC) [7]–[11] and model predictive control [12]. Their conclusions have been similar to those emphasized by the main researchers in AI domain [1], [2]: these conventional mechanisms have remarkable advantages, but they are suitable only for a small area of medical decision-making problems. Knowledge-based systems can be easily implemented for simple rules, but it is possible to encounter unexpected results if the problem that has to be solved involves many interconnected premises; therefore, fuzzy logic can be included. In addition to this, the transition from implicit knowledge (of human medical experts) to explicit rules (necessary for symbolic processing) can damage the information content. On the other hand, the most common method of statistical inference, which is Bayes' theorem, assumes that the diseases are mutually exclusive (sometimes this can be true) and, more restrictive, that the considered symptoms of diseases are statistically independent (more often, it is difficult to fulfill this constraint).

ANNs provide an alternative approach by training them to associate diagnoses (or other decisions) to the input data (symptoms, laboratory tests results, images, signals). During the training process, a set of examples consisting in input values, together with the already established decision for them, are repeatedly presented to the network (which has a specific architecture and some features that can be modified in this phase). Once the training is completed, the ANN should be able to generalize, offering a correct decision for a new set of input data.

Even if the literature contains an impressive number of papers that describe different ways of using ANNs, there is enough room for further developments. Technology becomes faster and faster. Biological neurons are switching at speeds that are million times slower than a computer gate [2]. Nevertheless, humans (and even animals) are more efficient in speech recognition and visual information processing than the fastest computer. Therefore, the research in ANNs field is still a challenge, as long as the understanding of human neural system is not completed, yet.

This paper describes, in several sections, the research results obtained by the authors in the field of ANNs developed for medical domain. It continues the work underlined in [15], adding some new results. Section 2 presents a literature review, where some current articles are considered, in order to prove that these aspects are still in top of researchers' concerns. Then, the paper continues, in Section 3, with the description of three common ANNs that analyze regular medical data (symptoms and laboratory tests results); one of them is used for detection, suggesting a diagnosis in the field of skin diseases, and the other two were trained to make predictions regarding the evolution of hepatitis B and the risk of stroke. ANNs are suitable for applications that use medical images in decisional process; Section 4 contains a detailed description of such a system. Section 5 presents a network of ANNs that was developed to make predictions regarding several different aspects in the same time. Then, Section 6 describes results concerning a variable structure Recurrent Neural Network (RNN) and a non-recurrent ANN with similar size, which replicate the nonlinear mechanism of a prosthetic hand based on surface myoelectric sensors and produce ANNs the midcarpal joint angles based on myoelectric signals obtained from surface myoelectric sensors. Building upon our results reported in [8], Section 7 offers an ILC-based approach to batch training a feedforward ANN architecture. Conclusions are pointed out in Section 8.

2. RELATED WORK

A quick look at recent research papers shows that the influence of AI on human life is continuously increasing. Medicine is part of this trend ([16]-[23]) and it could incorporate in the future much more applications based on AI mechanisms in general and, particularly, on machine learning techniques.

ANNs, which are machine learning models inspired from the architecture of the human brain, have been extensively used in various applications developed for medical field. Several examples are hereby presented.

In [16], the authors are using ANNs to predict acute rejection in liver transplant patients. The novelty and the advantage of their method is that it uses routine laboratory data only, being non-invasive (while conventional tests need biopsy, which is an invasive procedure).

The efficiency of ANNs is also proved in [17], where the authors present an overview regarding the use of ANNs in lung cancer research. They also underline that even if the literature shows that these tools are suitable for clinical decision support, a strict cooperation between physician and biostatistician is mandatory in order to avoid inaccurate use of ANNs.

Another recent example is provided in [18], where ANNs are used to analyze stomach images, classifying them as normal, benign or malign. The original images are processed in order to reduce their dimension. The accuracy of classification results places this method above others, in the authors' opinion.

Medical images are intensively used together with neural networks in decision-making process. For instance, an automated classification of skin lesions is performed by deep convolutional neural networks (CNNs) that analyze images in [19]. The results demonstrate that such tools are able to classify skin cancer with accuracy comparable to dermatologists.

The use of machine learning in stroke imaging is emphasized in [20]. The authors focus on technical principles, applications and perspectives, stating that these techniques may play an important role in setting the adequate therapeutic method or in predicting the prognosis for stroke patients.

As stated in the Introduction, ANN is just one technique besides many others that are part of AI domain. The comparison between these techniques is debated by researchers for decades. For instance, a study implemented in 2007 on a database of 1069 cases with coronary heart disease, indicated that ANN is the best identifier [21]. In 2017, several researchers included ANNs together with other classifiers in the same project [22], making use of the benefits provided by each of them in order to identify breast cancer.

A relevant study is provided in [23], where the authors forecast how medicine (particularly cardiovascular medicine) will incorporate AI in the future. They consider the most common machine learning techniques, including CNNs and RNNs.

Even if AI tools have already been used in medicine, the physicians still need to analyze a multitude of variables and relationships between these variables in order to identify the medical status of a patient. Meanwhile, the machine learning domain provides a series of algorithms used to represent data structures and to make predictions or classifications. Therefore, medicine is still a proper field for automated tools that may support decision-making process [15].

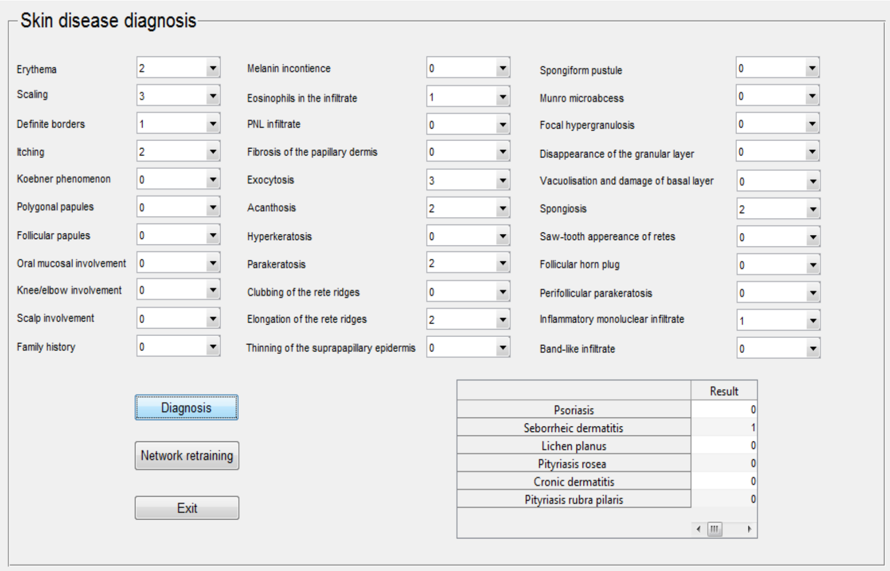
3. Artificial Neural Networks Used in Diagnosis and Prediction

ANNs offer support in decision-making process, so that the physicians can benefit of a faster diagnosis for diseases with various and confusing symptoms [15]. This section contains three relevant examples for this area [24]–[26].

3.1. Skin Diseases Diagnosis

The ANN described in [15] and [26] has been created and trained to suggest a diagnosis regarding skin diseases from erythemato-squamous class. It makes the difference between six such diseases: psoriasis, seborrheic dermatitis, lichen planus, pityriasis rosea, chronic dermatitis, and pityriasis rubra pilaris. Setting a diagnosis for these diseases is difficult and sometimes inaccurate because patients have multiple and vague symptoms.

There are 11 clinical features that are evaluated when a patient is suspected to have an erythemato-squamous disease [26]. These are enumerated in the left column of the graphical user interface provided by Fig. 1. Some of the erythemato-squamous diseases can be detected using these clinical features only. Nevertheless, a biopsy is also required for a correct diagnosis; 22 histopathological features can be evaluated from skin samples [26]. They are listed in the last two columns of Fig. 1.


Skin disease diagnosis

Erythema	2	Melanin incontience	0	Spongiform pustule	0
Scaling	3	Eosinophils in the infiltrate	1	Munro microabcess	0
Definite borders	1	PNL infiltrate	0	Focal hypergranulosis	0
Itching	2	Fibrosis of the papillary dermis	0	Disappearance of the granular layer	0
Koebner phenomenon	0	Exocytosis	3	Vacuolisation and damage of basal layer	0
Polygonal papules	0	Acanthosis	2	Spongiosis	2
Follicular papules	0	Hyperkeratosis	0	Saw-tooth appereance of retes	0
Oral mucosal involvement	0	Parakeratosis	2	Follicular horn plug	0
Knee/elbow involvement	0	Clubbing of the rete ridges	0	Perifollicular parakeratosis	0
Scalp involvement	0	Elongation of the rete ridges	2	Inflammatory monoluclear infiltrate	1
Family history	0	Thinning of the suprapapillary epidermis	0	Band-like infiltrate	0

	Result
Psoriasis	0
Seborrheic dermatitis	1
Lichen planus	0
Pityriasis rosea	0
Cronic dermatitis	0
Pityriasis rubra pilaris	0

Fig. 1 The graphical user interface for the skin diseases diagnosis system [15]

An analysis of these six diseases from medical point of view shows up that they have similar features. Therefore, the diagnosis is difficult for a human. An automated diagnosis system based on ANN can be a real help for the dermatologist. ANNs have the ability to overcome these problems because they learn by examples and then they can generalize and identify patterns.

Thus, a feedforward ANN with back-propagation learning algorithm has been created. Since an ANN learns by examples, a database of 366 patients with erythemato-squamous diseases has been used [26], [27]. Each patient has 33 symptoms and an already established diagnosis (one of the six considered diseases). Therefore, the input matrix of ANN has 366x33 elements, representing the symptoms of all the patients, and the target matrix has 366x6 elements, representing the diagnosis of all the patients. Between the 33 input elements and the 6 output neurons there is a hidden layer with 10 neurons.

Once the ANN is trained on a representative set of data, it can be used to diagnose a new patient. Through a user interface (Fig. 1 [15]), the physician enters the clinical and the histopathological features of his patient. Some of these features are binary values (1 – present, 0 – not present), others have an intensity degree between 0 (not present) and 3 (acute), with discrete intermediate values 1 and 2.

Based on these features, the ANN generates the diagnosis. At most one of the six outputs of the network is 1, the rest of them being 0. Table 1 contains several examples of inputs and outputs from the dataset [27] that has been used in ANN's training. A relevant difference with respect to [15] is that the current examples emphasize the idea that apparently similar features are defining different diseases, making the diagnosis process more difficult. The superscript T in Table 1 indicates matrix transposition.

Table 1 Examples of records from the skin diseases database

Record	Input vector	Diagnosis
1	$[2\ 2\ 2\ 1\ 0\ 0\ 0\ 0\ 2\ 2\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 0\ 2\ 2\ 3\ 2\ 0\ 0\ 0\ 2\ 0\ 0\ 0\ 0\ 0\ 2\ 0]^T$	psoriasis
2	$[2\ 2\ 2\ 3\ 2\ 3\ 0\ 2\ 0\ 0\ 0\ 2\ 0\ 0\ 0\ 2\ 2\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 3\ 0\ 2\ 0\ 2\ 0\ 0\ 2\ 3]^T$	seborrheic dermatitis
3	$[2\ 2\ 3\ 3\ 2\ 3\ 0\ 1\ 0\ 0\ 0\ 2\ 0\ 0\ 0\ 3\ 2\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 2\ 0\ 3\ 0\ 3\ 0\ 0\ 2\ 3]^T$	lichen planus
4	$[2\ 2\ 2\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 2\ 2\ 2\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 2\ 0]^T$	pityriasis rosea
5	$[2\ 2\ 0\ 2\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 2\ 2\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 1]^T$	chronic dermatitis
6	$[2\ 2\ 1\ 0\ 0\ 0\ 3\ 0\ 2\ 0\ 1\ 0\ 0\ 0\ 0\ 2\ 2\ 1\ 2\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 0\ 3\ 3\ 2\ 0]^T$	pityriasis rubra pilaris

The system's performance is next discussed. Its precision (93.7%) reported in [15] and [26] is comparable to that of a human expert; but it is infinitely faster. However, two features of this system should be improved. One refers to the fact that when two diagnoses are plausible, the system identifies just one of them because the outputs are binary values and do not have associated a plausibility score. The other one is connected to uncommon diseases, which are difficult to be identified by the system (because of the small number of examples in the training set).

Therefore, the research should continue in this field in order to find better solutions. One could be a hybrid expert system that combines the power of ANN with the sensitivity of other AI mechanisms.

3.2. Hepatitis B Prediction

The medical status of a patient can be improved if the physician would have some information about a possible future evolution of that patient. The system hereby presented has been developed for liver diseases. An ANN has been used to make suggestions regarding the evolution of patients infected with hepatitis B virus [25]. There are six possible severity levels of this infection: easy, medium, serious, prolonged, cholestatic, and comatose. Each patient can have a specific evolution, which can be predicted using this mechanism.

A function fitting neural network (which is a specialized version of feedforward neural network) has been used. The patient's current medical status is defined by 22 symptoms, which can be divided into: general features of the patient, subjective signs (that can be seen or easily detected), and objective signs (laboratory test results). An example of these features is given in Table 2. ANN receives as inputs these values in their initial form.

Table 2 Example of input values for hepatitis B predictions

<table>
<tr><td colspan="8">General features of the patient</td></tr>
<tr><td>Feature</td><td>Gender</td><td>Age</td><td>Living conditions</td><td>Contact</td><td>Transfusion</td><td>Parenteral treatments</td><td>Debut time</td></tr>
<tr><td>Value</td><td>Male</td><td>31</td><td>Urban</td><td>No</td><td>No</td><td>Yes</td><td>4</td></tr>
<tr><td colspan="8">Subjective signs</td></tr>
<tr><td>Feature</td><td>Anorexia</td><td>Nausea</td><td>Queasiness</td><td>Asthenia</td><td>Fever</td><td colspan="2">Hyperchromic urine</td></tr>
<tr><td>Value</td><td>Yes</td><td>Yes</td><td>No</td><td>Yes</td><td>No</td><td colspan="2">Yes</td></tr>
<tr><td>Feature</td><td>Tegument jaundice</td><td>Arthralgia</td><td>Myalgia</td><td>Skin eruption</td><td>Jaundice</td><td colspan="2">Liver dimension</td></tr>
<tr><td>Value</td><td>Yes</td><td>No</td><td>No</td><td>No</td><td>3</td><td colspan="2">2</td></tr>
<tr><td colspan="8">Objective signs</td></tr>
<tr><td>Feature</td><td colspan="3">Biliary retention</td><td colspan="2">Thymol</td><td colspan="2">Alanine transaminase (ALT)</td></tr>
<tr><td>Value</td><td colspan="3">4.81</td><td colspan="2">8</td><td colspan="2">840</td></tr>
</table>

ANN has an output layer, which produces the result. This is a number defining one of the six severity levels of the disease: 1 – easy, 2 – medium, 3 – serious, 4 – prolonged, 5 – cholestatic, 6 – comatose. In the training phase, besides the input values, the network also needs, for each patient, the severity level that was already set by a physician and that is assumed to be correct.

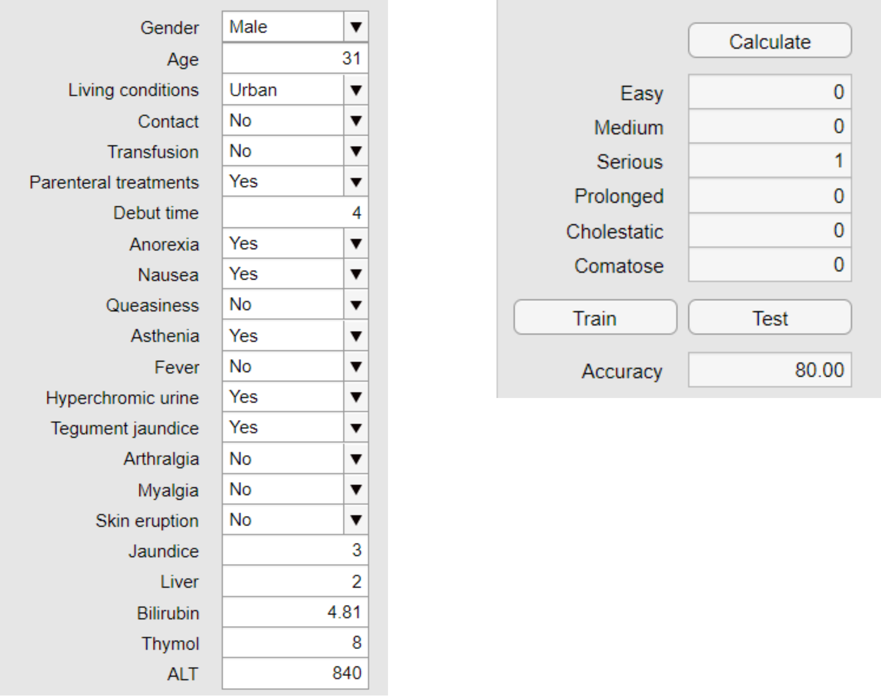
Between the inputs and the output there is a hidden layer of 10 neurons. This is an empirically established number, which usually is between the number of inputs (22, this case) and the number of outputs (one, for this system).

The Levenberg-Marquardt back-propagation algorithm is used for training. It is the most suitable for this type of problems, and also the fastest.

Each neuron of ANN reacts to its weighted inputs, producing an output. This is internally performed by an activation function. From this point of view, the current system contains two types of neurons: for those belonging to the hidden layer, the activation function is hyperbolic tangent sigmoid, while the neuron from the output layer has a linear activation function.

The graphical user interface is divided in several parts, and two of them are presented in Fig. 2. The input panel allows the user to enter the features of a new patient. The results are displayed by the output panel, where the accuracy of the system is also provided.

The system is using a database that contains 165 patients from the Clinical Hospital of Infectious Diseases No. 4 “Victor BABES”, Timisoara. 150 of them are used for training and 15 for testing.



Gender	Male
Age	31
Living conditions	Urban
Contact	No
Transfusion	No
Parenteral treatments	Yes
Debut time	4
Anorexia	Yes
Nausea	Yes
Queasiness	No
Asthenia	Yes
Fever	No
Hyperchromic urine	Yes
Tegument jaundice	Yes
Arthralgia	No
Myalgia	No
Skin eruption	No
Jaundice	3
Liver	2
Bilirubin	4.81
Thymol	8
ALT	840

Easy	0
Medium	0
Serious	1
Prolonged	0
Cholestatic	0
Comatose	0
Accuracy	80.00

a) The input panel b) The output panel

Fig. 2 The graphical user interface for hepatitis B prediction

The accuracy of the system is 80%. In order to obtain better results, the database could be improved in two directions: more records and a higher number of fields [25]. On the one hand, any classifier needs as many records as possible, in order to establish the influence of inputs (symptoms) on an output (disease or other prediction). On the other hand, a larger number of variables that describe a patient is desirable, but only if this action significantly increases the accuracy. These features should be relevant for the diseases that are classified. Otherwise, the overall performance of the system is depreciated because of the execution time, which is increasing [25].

3.3. Stroke Risk Prediction

As presented in the previous section, ANNs can be trained to make predictions regarding the future medical status of a patient. Another example is the system created for stroke, also called cerebrovascular accident, risk prediction [15], [26]. According to the World Health Organization [28], stroke is one of the two main frequent causes of death worldwide. It is also responsible for long-term disabilities, and the perspective is getting worse. Therefore, the concern of developing automatic tools for predictions of vascular events is fully justified.

The risk factors responsible for stroke [26], [29] can be grouped in two distinct categories: controllable (high blood pressure, diabetes, heart diseases, smoking, sedentariness, obesity) and not controllable (age, gender, race, prior stroke, family history). However, it is difficult to present a well-defined influence of these factors on stroke, so that the risk cannot be accurately established. A computational model, able to recognize special aspects, predicting a stroke risk may become crucial in prognostication, identifying the high-risk patients, and may have a future role in treatment selection [30]. ANNs are able to acquire, to store and to use experiential knowledge [2]. These features make them suitable for the exposed problem.

The prediction system [26] has been trained to recognize four levels of stroke risk. The training set contains information about 108 patients, which has been collected from the Municipal Emergency Hospital “Clinicile Noi” Timisoara, Romania, between 2015-2016. There are eleven features for each patient into the dataset and also the risk level, which is a value between 1 and 4 (the worst case is denoted by value 4). The fields of this dataset and some examples of values for several patients with different risk levels are presented in Table 3. More than in [15], these examples are indicating that it is difficult to determine the influence that each feature has on the stroke risk level.

Table 3 The fields of the database for stroke risk prediction (with examples)

Medical factor	1	Age	33	21	37	56
	2	Gender	1	1	2	2
	3	Dyslipidemia	100	90	169	149
	4	Abdominal Circumference	77	95	80	90
	5	HLV on Echocardiogram	90	57	105	88
	6	HLV EKG Sokolov-Lyon	20	24	42	48
	7	GFR	30	43	48	18
	8	LV Wall Thickness	0.2	0.7	0.9	0.9
	9	Postprandial Glycaemia	166	250	231	310
	10	A Jeun Glycaemia	102	113	118	150
	11	Previous Condition	0	0	1	3
Risk value			**1**	**2**	**3**	**4**

A feedforward ANN has been created for stroke risk prediction [26]. It has 11 inputs (the patient’s medical features involved in decisional process – these are the risk factors responsible for stroke) and one output (the value predicted for stroke risk). It was considered that one hidden layer should be enough (usually, this is sufficient for a large variety of problems). The number of the neurons in the hidden layer is one of the most important considerations when the architecture of the ANN is established. Nevertheless, there is no conclusive solution appropriate for any task [2]. The current system has a hidden layer of 10 neurons. The back-propagation algorithm has been used for the training process. It is one of the most commonly used algorithms for feedforward ANNs training.

The user interacts with the application through a graphical interface, where he enters the necessary information (the features of the patient that are requested for the decisional process) or he can load this information from an external file (if it is available). Then the system provides the stroke risk value and the physician can use it in the decisional process that follows.

Therefore, having this valuable information regarding a potential evolution of a patient, the human expert can choose the actions that are further required to avoid a stroke. This ANN is a relevant example of using automated decisional systems to improve patients’ medical condition.

Further developments can also be performed in this field. For instance, the risk of a second stroke or even the risk of death after a stroke can be predicted using ANNs [31].

4. Artificial Neural Networks Applied to Medical Image Analysis

Setting a diagnosis is the first step that should be accomplished when a person has a medical problem. But, as already stated in this paper, this could be a difficult action. Sometimes the symptomatic and analytic information presented by the patient are not relevant. For specific diseases, the solution can be provided by medical images, emphasizing internal problems of the human body that cannot be found analyzing the symptoms or the laboratory tests results.

The liver represents a great challenge to the radiologists because of the difficulty to appreciate the morphological changes induced by the illness [3]. For this reason, an ANN was developed in order to suggest a diagnosis regarding the liver diseases that can be detected analyzing information extracted from medical images obtained by Computed Tomography (CT). The system is able to detect three hepatic diseases: hepatomegaly, steatosis and tumors. In addition to the previous work described in [15], Fig. 3 contains three sequential slides from abdominal tomography of patients with these diseases, together with a healthy liver [3]. In this way the features of each disease can be observed more easily.

ANN is trained to distinguish between these four types of images. But, as can be seen in Fig. 3, a slice of an abdominal tomography contains, besides the liver, some other elements (parts of vertebral column and ribs, a portion of lung, etc.) that are not relevant for the problem that must be solved. For this reason, it is necessary to process a medical image before using it by an ANN.

The first step is segmentation and it is necessary to locate and to extract an object (the liver, in this case) from the initial image. Fig. 4 presents an image before and after segmentation process. Then, the processing continues, extracting some features that are relevant for liver’s diseases. One of these is texture and Fig. 3 emphasizes the differences of texture between the four states of the liver.

The texture features can be extracted using grey level spatial dependence matrices, also called co-occurrence matrices [3], which define the distribution of co-occurring pixel values for a specified offset. The offset is determined by an angle and a distance between pixels. The most used angles are 0°, 45°, 90°, and 135°. The co-occurrence matrices consist of the following elements expressed in a computer programming-like notation (with arguments instead of subscripts):

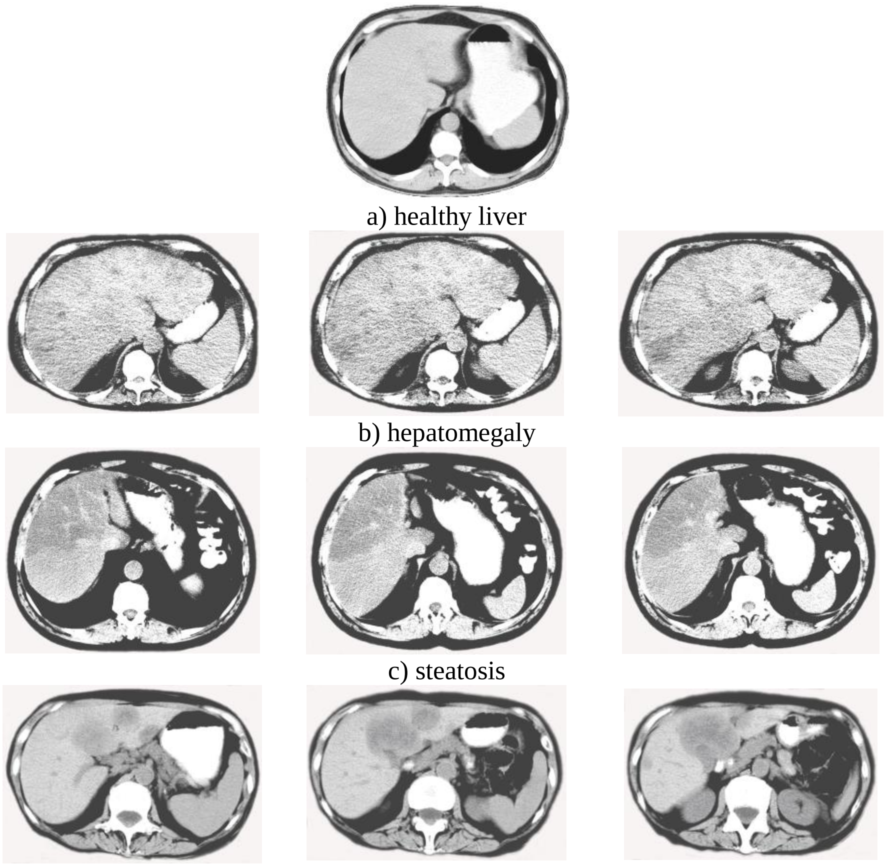
$$\mathrm{C}_{0^\circ,d}(\mathrm{i,j})=|\{((\mathrm{k,l}),(\mathrm{m,n}))\in I\text{: } \mathrm{k}-\mathrm{m}=0,\ |\mathrm{l}-\mathrm{n}|=d,\ I(\mathrm{k,l})=\mathrm{i},\ I(\mathrm{m,n})=\mathrm{j}\}|, \tag{1}$$

$$\mathrm{C}_{45^\circ,d}(\mathrm{i,j})=|\{((\mathrm{k,l}),(\mathrm{m,n}))\in I\text{: } (\mathrm{k}-\mathrm{m}=d,\mathrm{l}-\mathrm{n}=-d)\ \text{OR}\ (\mathrm{k}-\mathrm{m}=-\mathrm{d},\mathrm{l}-\mathrm{n}=d),\ I(\mathrm{k,l})=\mathrm{i},\ I(\mathrm{m,n})=\mathrm{j}\}|, \tag{2}$$

$$\mathrm{C}_{90^\circ,d}(\mathrm{i,j}) = |\{((\mathrm{k,l}),\ (\mathrm{m,n})) \in I\text{: } |\mathrm{k}-\mathrm{m}|=d,\ \mathrm{l}-\mathrm{n}=0,\ I(\mathrm{k,l})=\mathrm{i},\ I(\mathrm{m,n})=\mathrm{j}\}|, \tag{3}$$

$$C_{135°,d}(i,j)=|\{((k,l),(m,n))\in I\text{: } (k-m=d,l-n=d) \text{ OR } (k-m=-d,l-n=-d),\ I(k,l)=i,\ I(m,n)=j\}|, \quad (4)$$

where *I* is the image (in fact the matrix of gray level elements) and *d* is a distance between pixels.



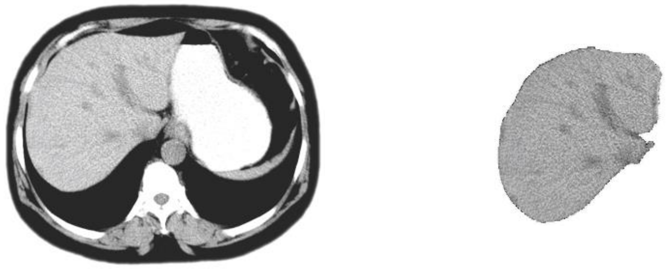
a) healthy liver

b) hepatomegaly

c) steatosis

d) tumors

Fig. 3 Slices of abdominal tomography



a) an entire slice

b) the liver

Fig. 4 Segmentation

The co-occurrence matrices have (each one) 256x256 elements, too many to be analyzed by an ANN. Therefore, the process of analyzing the image continues, extracting a set of six texture features from the co-occurrence matrices: energy, entropy, contrast, maximum element, inverse difference moment, and correlation.

These spatial grey tone co-occurrence texture features are usually called Haralick features. They were introduced by R. Haralick four decades ago, and they are still successfully used in image texture analysis.

An image is now described by 24 elements (six features for each of the four matrices). These are the inputs of the ANN. It is a feedforward ANN, with back-propagation training algorithm. It also has one hidden layer with 10 neurons and one output layer suggesting the diagnosis.

This ANN has a fixed architecture, therefore, during the training process, the weights of the connections between neurons are modified. This training process has two phases [3]: a preliminary phase, where the parameters receive their initial values, and a main phase, which is iterative, where the parameters are adjusted. The performance of an ANN depends not only on the way of modifying its weights, but also on their initial values. For this reason, in order to choose the best neural network, 500 ANNs with different initial values of the weights have been created and trained. Then, the ANN with the best accuracy has been used to provide the prediction regarding the healthy level of the liver. The training of this system was performed using abdominal CT images obtained from the "The Modeling Centre for Prosthesis and Surgical Interventions on the Human Skeleton" Multiple Users Research Base of the Politehnica University Timisoara.

Liver diseases can be a real danger for patients' lives because frequently, they have perceptible symptoms barely in advanced stages. An early detection using an automated diagnosis system that puts together the ability of ANNs and the accuracy of medical images might save lives.

5. A NETWORK OF ARTIFICIAL NEURAL NETWORKS

This section also treats an application in the area of liver diseases. Hepatitis C is a serious and frequent one. There is no vaccine against its virus, the treatment is very expensive (in some countries it is supported by national programs), and, more important, this treatment is not always efficient, sometimes causing severe adverse effects.

There are three treatments for hepatitis C [3]: Simple InterFeron (IFN), Peg interferon α-2a, and Peg interferon α-2b. At the beginning of the treatment, for both physician and patient, it would be good to know which of these treatments will have benefits (if there is one), in order to avoid undesired side effects.

The treatments for hepatitis C influence four biological indicators (TGP, TGO, GGT, and ARN VHC). The system described here (and also mentioned in [3] and [15]) is able to offer, for each of these four biological indicators, predictions regarding the evolution during the next 12 months, indicating its growing tendency, its stabilizing or its decreasing tendency. The physician could use these predictions in order to estimate the evolution of the patient during each possible treatment and to decide which the most suitable treatment for a specific patient is.

This system is, in fact, a network of strongly interconnected ANNs (Fig. 5). There are four sections that provide predictions regarding the evolution of the biological indicators after 3, 6, 9, and 12 months of treatment. Inside each section there are four ANNs, one for each biological indicator. All the 16 ANNs are feedforward neural networks and they are trained using back-propagation algorithm.

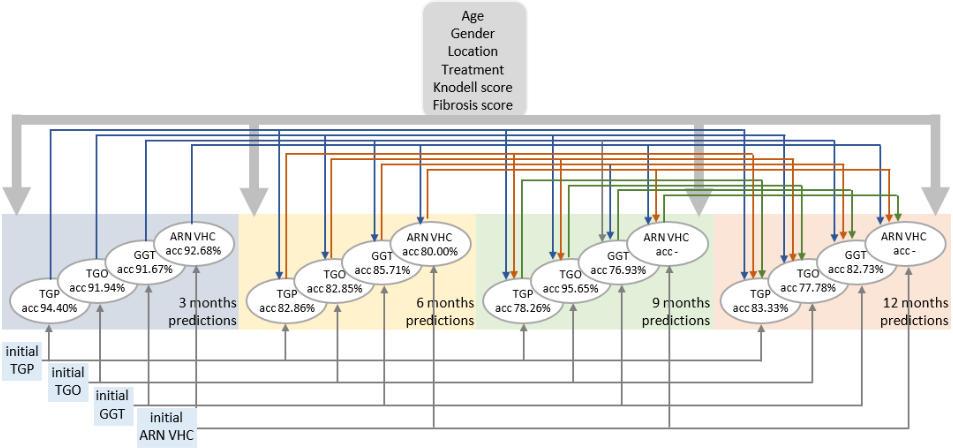
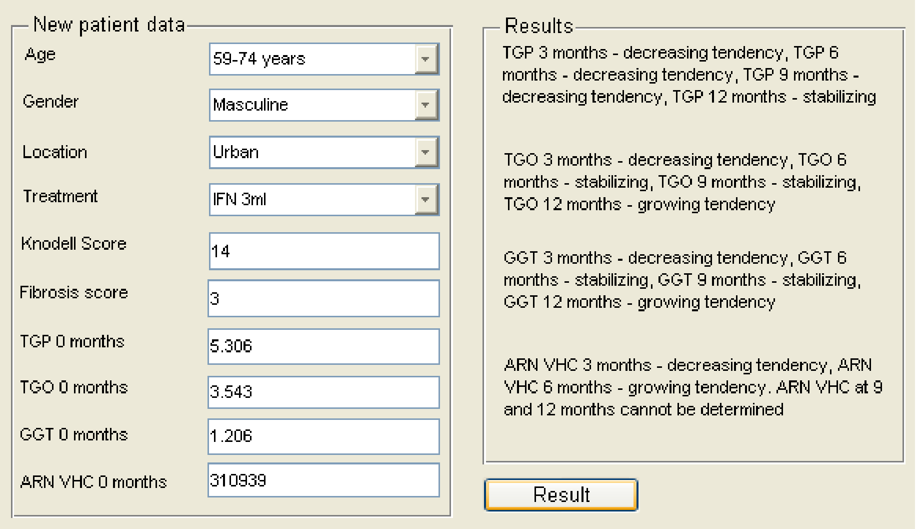


Fig. 5 The network of ANNs used for hepatitis C prediction

Each ANN has 10 hidden neurons, an output neuron (which predicts the evolution tendency of that biological indicator after 3, 6, 9, or 12 months) and a variable number of inputs. The networks belonging to the first section (the ones that predict the evolution of each biological indicator after the first 3 months of treatment) receive as inputs: the patient's age, the gender, the location where he/she lives (rural/urban), the treatment that is evaluated, the Knodell score, the hepatic fibrosis score and the value of the biological indicator for which the prediction is made, at the initial moment (before the treatment starts). The output of these networks is the value of that biological indicator after 3 months of treatment. The networks in the following sections have a similar structure, but they have as additional inputs the outputs of the networks in the previous sections (referring to the same biological indicator); therefore, the networks in the last section will have 10 inputs (the initial inputs and the values of biological indicators after 3, 6, and 9 months of treatment).

The human expert enters, through a graphical user interface (Fig. 6), the five features of the patient, the initial values of biological indicators and chooses one of the three treatments. Then the system provides the predictions regarding the evolution of the biological indicators during that treatment.

In this study, 193 patients have been under observation for 12 months in order to establish the treatment's influence on the evolution of the biological indicators. The information about these 193 patients (120 women and 73 men, between 14 and 67 years of age) has been collected from the Gastroenterology Department of the Emergency Clinical Hospital, Timisoara, Romania.



New patient data

Age	59-74 years
Gender	Masculine
Location	Urban
Treatment	IFN 3ml
Knodell Score	14
Fibrosis score	3
TGP 0 months	5.306
TGO 0 months	3.543
GGT 0 months	1.206
ARN VHC 0 months	310939

Results

TGP 3 months - decreasing tendency, TGP 6 months - decreasing tendency, TGP 9 months - decreasing tendency, TGP 12 months - stabilizing

TGO 3 months - decreasing tendency, TGO 6 months - stabilizing, TGO 9 months - stabilizing, TGO 12 months - growing tendency

GGT 3 months - decreasing tendency, GGT 6 months - stabilizing, GGT 9 months - stabilizing, GGT 12 months - growing tendency

ARN VHC 3 months - decreasing tendency, ARN VHC 6 months - growing tendency. ARN VHC at 9 and 12 months cannot be determined

Fig. 6 The graphical user interface for hepatitis C prediction system [15]

Regarding the accuracy of this system, each of the 16 ANNs has its own value. In fact, for each of them have been created and trained 500 neural networks with the same architecture, but different initial parameters. The network with the best accuracy was chosen to be used for predictions. The results are encouraging: most of ANNs having accuracies around 85% (according to [3] and [15]), as “acc” parameter of each ANN from Fig. 3 shows. It can be observed that the accuracy of all the networks from the first section is greater than 90% and that in most of the cases the accuracy is decreasing along the chain of networks. One reason could be the fact that not only the prediction is propagating through the network of ANNs, but also the error.

The concern regarding the hepatitis C virus infection is still a major issue in medical domain. Even if the treatments for hepatitis C are continuously improved and becoming more and more efficient, the patient evolution during the treatment has to be carefully observed in order to react if something goes wrong. The prediction system presented here provides an overview for 12 months of treatment. This contains information that can be used by the physician in the decisional process of choosing the best treatment for a patient.

6. Artificial Neural Networks Applied to Modeling Finger Dynamics

This section treats the use of neural networks in modeling the finger dynamics of an amputated hand and the control of a prosthetic hand based on the architecture specified in [6] and [32]. The inputs of the system are 8 myoelectric sensors placed:

- four sensors on the flexor digitorum superficialis (Fig. 7a),
- two sensors on the extensor digitorum and one on extensor digiti minimi (Fig. 7b),
- one sensor on abductor pollicis longus (Fig. 7c).

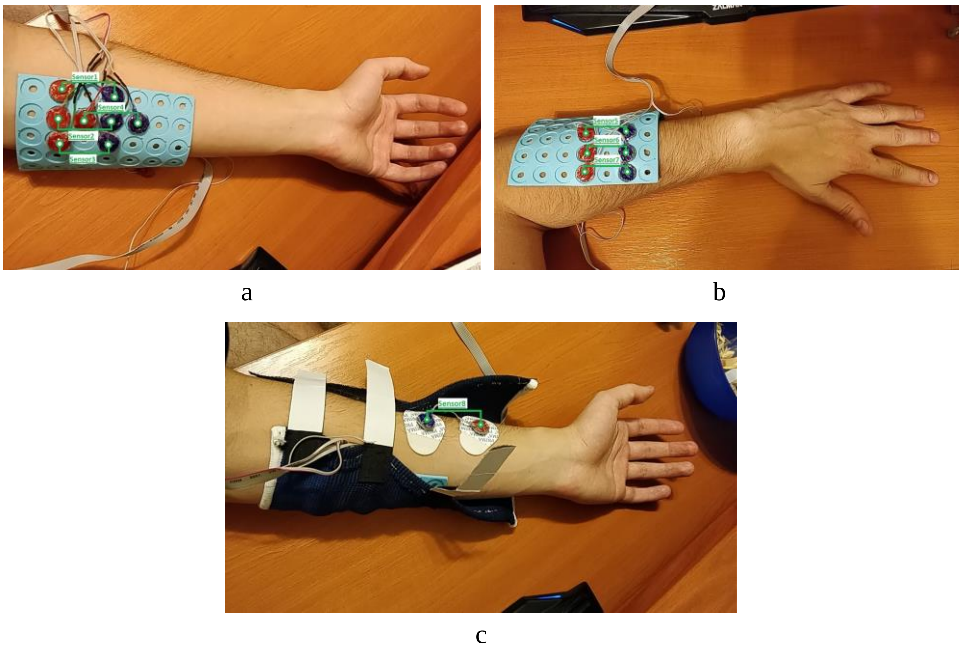


Fig. 7 Placement of sensors 1 to 4 (a), sensors 5 to 7 (b) and sensor 8 (c) on the hand [6]

The main RNN considered in [6] is a Long-Short Term Memory (LSTM) network [33], [34], which is next extended so as to meet the finger dynamics modeling purpose. The RNN is structured on 4 players as follows: an input layer, an LSTM layer, a hidden layer and an output layer.

The input is composed of 24 values [6] and is connected to the neural network. Each input is sampled at 10 ms and is structured as follows in terms of input vector **I**:

$$\mathbf{I} = [z_{1,n} \ \ldots \ z_{8,n} \ \ a_{1,n} \ \ldots \ a_{8,n} \ \ s_{1,n} \ \ldots \ s_{8,n}]^T \in \Re^{24}, \tag{5}$$

where $z_{j.n}$ is the output of myoelectric sensor j, j=1…8, the average of the past ten samples (100 ms) of myoelectric sensor j, j = 1…8, is $a_{j,n}$:

$$a_{j,n} = \frac{1}{10}\sum_{i=0}^{9} z_{j,n-i}, \tag{6}$$

and the average of the past 100 samples (1 s) of myoelectric sensor j, j=1…8, is $s_{j.n}$:

$$s_{j,n} = \frac{1}{100}\sum_{i=0}^{99} z_{j,n-i}. \tag{7}$$

The LSTM layer is composed of 300 neurons and connects the input layer to the hidden layer but also memorizes the past 100 time steps (1 second) in order to provide how the signals change in time.

The hidden layer contains 300 neurons and is connected to the current LSTM layer for a new abstractization of the inputs. This layer of abstractization helps the LSTM to output values in the first second without having to wait for the first 100 samples to be received

The output layer contains five neurons and produces output vector **O**:

$$\mathbf{O} = [o_{1,n} \;\; o_{2,n} \;\; o_{3,n} \;\; o_{4,n} \;\; o_{5,n}]^T, \tag{8}$$

which outputs the flex percentage of each finger at time stamp t_n.

This model contains a total number of 1,112,705 parameters which were trained. They are distributed as follows: 390,000 in the first LSTM layer, 721,200 in the hidden LSTM layer and 1,505 in the output layer.

The RNN structure is described Fig. 8 (a Multi Input-Single Output system structure) for normal operation.

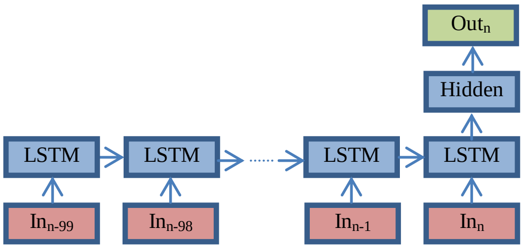


Fig. 8 LSTM network after the first second [6]

The RNN model results are compared to the real hand movements (captured by the flex sensor) using the percent Root Mean Square Error (RMSE) to express the model accuracy. The number of data samples in the dataset is 18,490 for the validation dataset. As shown in [6], the training of the networks was done on a training data set of 110,374 samples, equivalent to 1,103.74 s. The RMSE value for the RNN on the validation data is 8 to 9% depending on the training performance. The other two ANNs used for comparison are the regular ANN (i.e., non-recurrent) with an RMSE of 13 to 14% and the same RNN architecture with the validation data used also for training, which generated an RMSE of 2%.

The regular ANN consists of 3 layers, similar in size with the RNN but without the memory, arranged as an input layer of 300 neurons, a hidden layer of 300 neurons and an output layer of 5 neurons. The total number of trained parameters is 99,305 distributed across the layers as follows: input – 7,500, hidden – 90,300 and output – 1,505. The activation function for all the layers is sigmoid and the training was done using the Adam optimizer in which a dropout of 0.2 was used for a better generalization of the training.

The training of the RNN is the same with the regular ANN except that the used optimizer is Root Mean Square Propagation, which had a better performance than Adam with a 1% increased accuracy.

The results are exemplified in Fig. 9 for only one finger (index) to simplify the figure. The different lines are a comparison between the expected result (the real finger angle), blue the RNN output (green), the regular ANN output (red), and the output of RNN when the validation data was also used for training the RNN (magenta).

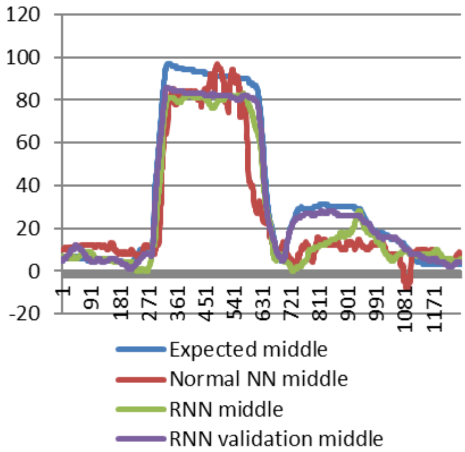


Fig. 9 Results for index finger: expected output (blue), regular ANN output (red), RNN output (green) and output of RNN trained with validation data (magenta)

The main increase of accuracy for both ANNs was made by several solutions like changing the number of neurons per layer, the number of layers or the type of network, but the main increase is given by the data preprocessing before the network training.

7. An ILC-based Approach to Artificial Neural Network Batch Training

As shown in [8], a feedforward ANN architecture is considered, which consists of one hidden layer with a hyperbolic tangent activation function and a single linear neuron. The nonlinear input-output map specific to ANN is:

$$y(k+1) = \mathbf{W}^T(k)\boldsymbol{\sigma}(\mathbf{V}(k), \mathbf{x}(k)), \tag{9}$$

where k indicates the discrete time moment, $k = 0...N$, and also the data sample index, and it is also dropped out as follows if the simplification of notations is intended, y is the output, $\mathbf{W}$ is the vector of output layer weights, $\boldsymbol{\sigma}$ is the vector of outputs of hidden layer neurons:

$$\begin{aligned} \mathbf{W} &= [w_0 \quad w_1 \quad ... \quad w_H]^T \in \Re^{H+1}, \\ \boldsymbol{\sigma} &= [1 \quad \sigma_1(\mathbf{V}_1^T\mathbf{x}) \quad ... \quad \sigma_H(\mathbf{V}_H^T\mathbf{x})]^T, \end{aligned} \tag{10}$$

and $\sigma_m(x) = \tanh(x), m = 1...H$, are hyperbolic activation tangent activation functions. Other activation functions can be considered as well.

The input vector is

$$\mathbf{x} = [x_0 \quad x_1 \quad ... \quad x_{nu}]^T \in \Re^{nu}, \tag{11}$$

and the first term in $\boldsymbol{\sigma}$ given in (10) corresponds to the bias of the output neuron. Each neuron in the hidden layer neuron is parameterized by the vector of weights $\mathbf{V}^m$, $m = 1...H$, and vector $\mathbf{V}$ in (9) is

$$(\mathbf{V}^m)^T = [(\mathbf{V}^m)^T \quad (\mathbf{V}^2)^T \quad ... \quad (\mathbf{V}^H)^T]^T, \\ \mathbf{V}^m = [v_m^0 \quad v_m^1 \quad ... \quad v_m^{nu}]^T \in \Re^{nu+1}. \tag{12}$$

Each vector $\mathbf{V}^m$ includes weight $\overset{0}{v_m}$ of the bias of m^{th} neuron. The number of ANN inputs is $nu + 1$, and the number of neurons in the hidden layer is H.

The nonlinear map specific to ANN expressed in (9) is next organized as a nonlinear Multi Input-Multi Output dynamical system considered in the iteration domain [8]:

$$\mathbf{W}_{j+1} = \mathbf{W}_j + \mathbf{u}_j^w, \\ \mathbf{V}_{j+1}^i = \mathbf{V}_j^i + \mathbf{u}_j^{v^i}, i = 1...H, \\ \mathbf{Y}_j(k+1) = \mathbf{W}_j{}^T \boldsymbol{\sigma}(\mathbf{V}_j^i, \mathbf{x}(k)), k = 0...N, \tag{13}$$

where the vector expressions are:

$$\mathbf{u}_j^w = [u_j^{w0} \quad ... \quad u_j^{wH}]^T \in \Re^{H+1}, \\ \mathbf{u}_j^{v^i} = [u_j^{v^{i0}} \quad ... \quad u_j^{v^{inu}}]^T \in \Re^{nu+1}, \\ \mathbf{Y}_j = [y_j(1) \quad ... \quad y_j(N+1)]^T \in \Re^{N+1}, \\ \mathbf{X}_j = [\mathbf{x}_j^T(0) \quad ... \quad \mathbf{x}_j^T(N)]^T \in \Re^{(N+1)(nu+1)}, \tag{14}$$

j is the iteration index, $\mathbf{u}_j^w, \mathbf{u}_j^{v^i}$ are the input vectors, and weight vectors $\mathbf{W}_j, \mathbf{V}_j^i$ are the state vectors of the dynamical system given in (13). Vector $\mathbf{X}_j$ is a trial-repetitive time-series disturbance input and also a time-varying parameter vector of (13). Vector $\mathbf{Y}_j$ is the output of the nonlinear dynamical system given in (13).

Using ILC framework, the dynamical system given in (13) is transformed into an input-output static map, which, in fact, ensures an ANN. The general goal of ILC is to reduce or minimize the tracking error expressed as the difference between the actual output and a desired output. The desired output vector specific to ANN, where the system (or process if the control is targeted) modeling is carried out, is $\mathbf{Y}_d$:

$$\mathbf{Y}_d = [y_d(1) \;\; ... \;\; y_d(k) \;\; ... \;\; y_d(N+1)]^T \in \Re^{N+1}, \tag{15}$$

where $y_d(k)$ is the desired system (process) output. In this regard, the batch training of ANN can be considered as a supervised learning approach, where the goal is to minimize the tracking error vector (in terms of ILC), also called training error vector (in terms of ANN) $\mathbf{E}_j$:

$$\mathbf{E}_j = \mathbf{Y}_j - \mathbf{Y}_d. \tag{16}$$

However, the input at each iteration can be derived in the framework of norm-optimal ILC as the solution to the optimization problem:

$$(\mathbf{u}_j^{w^*}, \mathbf{u}_j^{v^{i^*}}) = \arg\min_{\mathbf{u}_j^w, \mathbf{u}_j^{v^i}} \left\| \mathbf{E}_{j+1}^T \, \mathbf{R} \, \mathbf{E}_{j+1} + \mathbf{U}_j^T \, \mathbf{Q} \, \mathbf{U}_j \right\|_2^2, \tag{17}$$

where the stacked vector of inputs is $\mathbf{U}_j$:

$$\mathbf{U}_j = [(\mathbf{u}_j^w)^T \quad (\mathbf{u}_j^{v^1})^T \quad \ldots \quad (\mathbf{u}_j^{v^H})^T]^T \in \Re^{H+1+H(nu+1)}, \tag{18}$$

weighting matrices $\mathbf{R} = \mathbf{R}^T \succ 0$ and $\mathbf{Q} = \mathbf{Q}^T \succ 0$ are diagonal ones (for the sake of simplicity), $\mathbf{E}_{j+1} = \mathbf{Y}_{j+1} - \mathbf{Y}_d$ is the tracking error at iteration $j+1$, and $\|\bullet\|$ is the Euclidean norm of vector $\bullet$. The second term in the objective function in (17), which weights vector $\mathbf{U}_j$, is added in order to prevent over-fitting.

The popular nonlinear least-squares approach is applied to obtain the analytical solution to the optimization problem defined in (17). The linearization of the nonlinear input-output map specific to ANN of type (9) at the iteration *j*+1:

$$y_{j+1}(k+1) = \mathbf{W}_{j+1}{}^T \boldsymbol{\sigma}(\mathbf{V}_{j+1}^i, \mathbf{x}(k)), k = 0 \ldots N \tag{19}$$

is carried out around $\mathbf{W}_j, \mathbf{V}_j^i$ for small variations of $\mathbf{u}_j^w, \mathbf{u}_j^{v^i}$ by considering the output as a nonlinear function of the weight vectors as follows:

$$y_{j+1}(k+1) = f(\mathbf{W}_{j+1}, \mathbf{V}_{j+1}^i, \mathbf{x}(k)), k = 0 \ldots N, \tag{20}$$

and input vector $\mathbf{x}(k)$ as a parameter vector. The Taylor series expansion of (19) in the vicinity of an arbitrary operating point is:

$$\begin{aligned} y_{j+1}(k+1) = {} & \mathbf{W}_j{}^T \boldsymbol{\sigma}(\mathbf{V}_j^i, \mathbf{x}(k)) + [1 \quad \tanh(\mathbf{V}_j^{1^T} \mathbf{x}(k)) \quad \ldots \quad \tanh(\mathbf{V}_j^{H^T} \mathbf{x}(k))] \mathbf{u}_j^w \\ & + w_j^1 \frac{4}{(e^{\mathbf{V}_j^{1^T} \mathbf{x}(k)} + e^{-\mathbf{V}_j^{1^T} \mathbf{x}(k)})^2} \mathbf{x}^T(k) \mathbf{u}_j^{v^1} + \ldots + w_j^H \frac{4}{(e^{\mathbf{V}_j^{H^T} \mathbf{x}(k)} + e^{-\mathbf{V}_j^{H^T} \mathbf{x}(k)})^2} \mathbf{x}^T(k) \mathbf{u}_j^{v^H} + h.o.t., \end{aligned} \tag{21}$$

where *h.o.t.* indicates higher-order terms, which are next neglected.

Using (19) with *j* instead of *j*+1, introducing the notations:

$$\begin{aligned} g_i(k) &= \frac{4}{(e^{\mathbf{V}_j^{i^T} \mathbf{x}(k)} + e^{-\mathbf{V}_j^{i^T} \mathbf{x}(k)})^2}, \\ \boldsymbol{\sigma}_j(k) &= [1 \quad \tanh(\mathbf{V}_j^{1^T} \mathbf{x}(k)) \quad \ldots \quad \tanh(\mathbf{V}_j^{H^T} \mathbf{x}(k))]^T, \end{aligned} \tag{22}$$

and neglecting *h.o.t.* in (21), the result is:

$$y_{j+1}(k+1) = y_j(k+1) + \boldsymbol{\sigma}_j{}^T(\mathbf{x}(k)) \mathbf{u}_j^w + w_j^1 g_1(k) \mathbf{x}^T(k) \mathbf{u}_j^{v^1} + \ldots + w_j^H g_H(k) \mathbf{x}^T(k) \mathbf{u}_j^{v^H}. \tag{23}$$

Stacking the $N+1$ outputs in (23) over the time argument *k* leads to [8]:

$$\mathbf{Y}_{j+1} = \mathbf{Y}_j + \boldsymbol{\Psi}_j \mathbf{U}_j, \boldsymbol{\Psi}_j \in \Re^{(N+1)\times(H+1+H(nu+1))},$$
$$\boldsymbol{\Psi}_j = \begin{bmatrix} \boldsymbol{\sigma}_j^T(\mathbf{x}(0)) & w_j^1 g_1(0)\mathbf{x}^T(0) & \ldots & w_j^H g_H(0)\mathbf{x}^T(0) \\ \boldsymbol{\sigma}_j^T(\mathbf{x}(1)) & w_j^1 g_1(1)\mathbf{x}^T(1) & \ldots & w_j^H g_H(1)\mathbf{x}^T(1) \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{\sigma}_j^T(\mathbf{x}(N)) & w_j^1 g_1(N)\mathbf{x}^T(N) & \ldots & w_j^H g_H(N)\mathbf{x}^T(N) \end{bmatrix}. \tag{24}$$

But the tracking error vector at iteration j+1 can also be expressed as:

$$\mathbf{E}_{j+1} = \mathbf{Y}_{j+1} - \mathbf{Y}_d = \mathbf{Y}_j + \boldsymbol{\Psi}_j \mathbf{U}_j - \mathbf{Y}_d = \mathbf{E}_j + \boldsymbol{\Psi}_j \mathbf{U}_j, \tag{25}$$

which determines the transformation of the optimization problem defined in (17) into the following quadratic one:

$$\mathbf{U}_j^* = \arg\min_{\mathbf{U}_j} \left\| \mathbf{U}_j^T \, \mathbf{X} \, \mathbf{U}_j + 2\, \mathbf{Z} \, \mathbf{U}_j + \mathbf{E}_j^T \, \mathbf{R} \, \mathbf{E}_j \right\|_2^2, \tag{26}$$

where:

$$\mathbf{X} = \boldsymbol{\Psi}_j^T \, \mathbf{R} \, \boldsymbol{\Psi}_j + \mathbf{Q}, \; \mathbf{Z} = \mathbf{E}_j^T \, \mathbf{R} \, \boldsymbol{\Psi}_j. \tag{27}$$

Using the matrix derivation rules and using the fact that **X** is symmetric because **R** and **Q** are symmetric, the analytic solution to the optimization problem defined in (25) is:

$$\mathbf{U}_j^* = -(\mathbf{X}^T)^{-1} \mathbf{Z}^T = -(\boldsymbol{\Psi}_j^T \, \mathbf{R} \, \boldsymbol{\Psi}_j + \mathbf{Q})^{-1} \boldsymbol{\Psi}_j^T \, \mathbf{R} \, \mathbf{E}_j = -\mathbf{K}_j \, \mathbf{E}_j, \tag{28}$$

where:

$$\mathbf{K}_j = (\boldsymbol{\Psi}_j^T \, \mathbf{R} \, \boldsymbol{\Psi}_j + \mathbf{Q})^{-1} \boldsymbol{\Psi}_j^T \, \mathbf{R}. \tag{29}$$

Matrix $\mathbf{K}_j$ can be obtained easily because matrices $\boldsymbol{\Psi}_j, \mathbf{E}_j$ can be computed at the current iteration. The optimal solution (vector) $\mathbf{U}_j^*$ contains the optimal increments of ANN weights. Using the following partitioning of $\mathbf{K}_j$:

$$\mathbf{K}_j = [\mathbf{K}_j^{w^T} \quad \mathbf{K}_j^{v^{1^T}} \quad \ldots \quad \mathbf{K}_j^{v^{H^T}}]^T, \tag{30}$$

with $\mathbf{K}_j^w \in \Re^{(H+1)\times(N+1)}, \mathbf{K}_j^{v^i} \in \Re^{(nu+1)\times(N+1)}$, the first two equations in (13) are transformed into:

$$\begin{aligned} \mathbf{W}_{j+1} &= \mathbf{W}_j - \mathbf{K}_j^w \mathbf{E}_j, \\ \mathbf{V}_{j+1}^i &= \mathbf{V}_j^i - \mathbf{K}_j^{v^i} \mathbf{E}_j. \end{aligned} \tag{31}$$

The two equations in (31) are update laws in the ILC framework, and they depend on the error at current iteration j. Concluding, solving the optimization problem defined in (17) at each iteration, results in the expression (15) of ANN training equations. This ANN training approach is general because the formulation of the objective function in (17) brings a degree of freedom in the learning process.

8. CONCLUSIONS

The amount of medical information that describes a patient and that has to be processed in order to generate a decision is continuously increasing. Therefore, it is becoming obvious that medicine needs to incorporate some automated tools able to provide valuable support in decision-making process.

As shown in the invited paper [15], the current paper continues the presentation of the authors' results offered in [15] and also shows that ANNs are suitable for such applications, being reliable tools, which are adapting to the informational environment specific to the problem they are solving. They are providing accuracies that are comparable to those of human experts. More than this, an ANN has the advantage that it is faster, more efficient and the experiments it is used for can be easily repeated.

The results presented by this paper are encouraging. Of course, they can be improved. One of the key factors that influence the accuracy of these tools, increasing their reliability, is the number of available patients used for training. From this point of view, the content of the databases available for these experiments is appropriate (366 patients for skin diseases diagnose, 165 for hepatitis B virus infection, 108 for stroke, 193 for hepatitis C treatment suggestion). However, better results will be obtained if a larger number of records will be used in the training phase.

Even if medical experts became aware that AI is a necessity and even if they are already using such tools, computer-aided decision-making systems still have enough room for improvements. All these applications lead to a favorized topic nowadays: personalized medicine. The authors intend to continue the research in this direction in order to develop new tools and methods able to improve the healthcare system, in general, and particularly, the patients' lives. It is still challenging to create a reliable prognostic model able to achieve the requested confidence in order to be used in real clinical circumstances. Future research will deal with the combination with fuzzy modeling and the transfer of results to various nonlinear models using results from other modeling and control applications [35]–[40], fuzzy [32], [41]–[44] and other models [45]–[52] applied to the medical field, and the consideration of other nonlinear models that proved to be successful in different fields [53]–[57] including various ANN architectures [58]–[61] and optimization techniques [62]–[68].

Therefore, ANNs, which are probably the most common AI techniques could lead to a significant improvement of the medical environment. Also, due to the increase number of inputs and the interdependencies between them, deep ANNs start to surpass the accuracy of the shallow ANNs and will probably become a necessity in the medical field. The physicians will be able to analyze in a greater depth a much larger quantity of data and the patients will have the chance to benefit from a higher quality healthcare system.

Acknowledgements: *This work was supported by the CEEPUS network "BG-0722 Computer Aided Design of automated systems for assembling" and by the valuable activity of the team from the Faculty of Mechanical Engineering of the University of Niš, Serbia. The support of Dr. Mircea-Radac in the development of the ILC-based training approach is duly acknowledged.*

REFERENCES

1. Russell, S.J., Norvig, P., 2010, *Artificial Intelligence: A Modern Approach*, 3rd ed., Pearson, Upper Saddle River, NJ, USA.
2. Zurada, J.M., 2012, *Introduction to Artificial Neural Systems*, Jaico Publishing House, Mumbai, India.
3. Albu, A., 2009, *Decisional methods applied in medical domain*, Proc. 5th International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romania, pp. 123-128.
4. Albu, A., Stanciu, L., 2015, *Benefits of using artificial intelligence in medical predictions*, Proc. 5th IEEE International Conference on E-Health and Bioengineering, Iasi, Romania, pp. 1-4.
5. Teban, T.-A., Precup, R.-E., de Oliveira, T.E.A., Petriu, E.M., 2-016, *Recurrent dynamic neural network model for myoelectric-based control of a prosthetic hand*, Proc. 2016 IEEE International Systems Conference, Orlando, FL, USA, pp. 1-6.
6. Teban, T.-A., Precup, R.-E., Lunca, E.-C., Albu, A., Bojan-Dragos, C.-A., Petriu, E.M., 2018, *Recurrent neural network models for myo-electric-based control of a prosthetic hand*, Proc. 22nd International Conference on System Theory, Control and Computing, Sinaia, Romania, pp. 1-6.
7. Radac, M.-B., Precup, R.-E., Petriu, E.M., Preitl, S., 2014, *Iterative data-driven controller tuning with actuator constraints and reduced sensitivity*, Journal of Aerospace Information Systems, 11(9), pp. 551-564.
8. Radac, M.-B., Precup, R.-E., Petriu, E.M., Preitl, S., 2014, *Iterative data-driven tuning of controllers for nonlinear systems with constraints*, IEEE Transactions on Industrial Electronics, 61(11), pp. 6360-6368.
9. Radac, M.-B., Precup, R.-E., Petriu, E.M., 2015, *Constrained data-driven model-free ILC-based reference input tuning algorithm*, Acta Polytechnica Hungarica, 12(1), pp. 137-160.
10. Radac, M.-B., Precup, R.-E., 2015, *Data-based two-degree-of-freedom iterative control approach to constrained non-linear systems*, IET Control Theory & Applications, 9(7), pp. 1000-1010.
11. Radac, M.-B., Precup, R.-E., 2016, *Three-level hierarchical model-free learning approach to trajectory tracking control*, Engineering Applications of Artificial Intelligence, 55, pp. 103-118.
12. Radac, M.-B., Precup, R.-E., Roman, R.-C., 2017, *Model-free control performance improvement using virtual reference feedback tuning and reinforcement Q-learning*, International Journal of Systems Science, 48(5), pp. 1071-1083.
13. Radac, M.-B., Precup, R.-E., 2018, *Data-driven model-free slip control of anti-lock braking systems using reinforcement Q-learning*, Neurocomputing, 275, pp. 317-329.
14. Radac, M.-B., Precup, R.-E., Roman, R.-C., 2018, *Data-driven model reference control of MIMO vertical tank systems with model-free VRFT and Q-learning*, ISA Transactions, 73, pp. 227-238.
15. Albu, A., Precup, R.-E., Teban, T.-A., 2018, *Medical applications of artificial neural networks*, Proc. XIV International SAUM Conference on Systems, Automatic Control and Measurements, Nis, Serbia, pp. 1-11.
16. Zare, A., Zare, M.-A., Zarei, N., Yaghoobi, R., Zare, M.-A., Salehi, S., Geramizadeh, B., Malekhosseini, S.-A., Azarpira, N., 2017, *A neural network approach to predict acute allograft rejection in liver transplant recipients using routine laboratory data*, Hepatitis Monthly, 17(12), paper e55092.
17. Bertolaccini, L., Solli, P., Pardolesi, A., Pasini, A., 2017, *An overview of the use of artificial neural networks in lung cancer research*, Journal of Thoracic Disease, 9(4), pp. 924-931.
18. Korkmaz, S.-A., Binol, H., Akcicek A., Korkmaz, M.-F., 2017, *An expert system for stomach cancer images with Artificial Neural Network by using HOG Features and Linear Discriminant Analysis: HOG_LDA_ANN*, Proc. IEEE 15th International Symposium on Intelligent Systems and Informatics, Subotica, Serbia, pp. 327-332, 2017.
19. Esteva, A., Kuprel, B., Novoa, R.-A., Ko, J., Swetter, S.-M., Blau, H.-M., Thrun, S., 2017, *Dermatologist-level classification of skin cancer with deep neural networks*, Nature, 542(7639), pp. 115-118.
20. Lee, E.-J., Kim, Y.-H., Kim, N., Kang, D.-W., 2017, *Deep into the brain: artificial intelligence in stroke imaging*, Journal of Stroke, 19(3), pp. 277-285.
21. Chen, J.X., Xing, Y.W., Xi, G.C., Chen, J., Yi, J.Q., Zhao, D.B., Wang, J., 2007, *A comparison of four data mining models: Bayes, neural network, SVM and decision trees in identifying syndromes in coronary heart disease*, Proc. 4th International Symposium on Neural Networks, Nanjing, China, pp. 1274-1279.
22. Udayakumar, E., Santhi, S., Vetrivelan, P., 2017, *An investigation of Bayes algorithm and neural networks for identifying the breast cancer*, Indian Journal of Medical and Paediatric Oncology, 38(3), pp. 340-344.
23. Johnson, K.-W., Soto, J.-T., Glicksberg, B.-S., Shameer, K., Miotto, R., Ali, M., Ashley, E., Dudley, J.-T., 2018, *Artificial intelligence in cardiology*, Journal of the American College of Cardiology, 71(23), pp. 2668-2679.
24. Filimon, D.-M., Albu, A., 2014, *Skin diseases diagnosis using artificial neural networks*, Proc. 9th IEEE International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romania, pp. 189-194.

25. Albu, A., Pasca, M.-S., Zimbru, C.-G., 2019, *Medical predictions: Bayes' theorem vs artificial neural networks*, Proc. 13th IEEE International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romania, pp. 1-4.

26. Tanasoiu, I., Albu, A., 2017, *A connectionist model for cerebrovascular accident risk prediction*, Proc. 6th IEEE International Conference on E-Health and Bioengineering, Sinaia, Romania, pp. 45-48.

27. UCI Machine Learning Repository - Dermatology Data Set, http://archive.ics.uci.edu/ml/datasets/Dermatology, last accessed 2018.

28. World Health Organization, http://www.who.int, last accessed 2019.

29. Avram, R., 2012, *Elemente de clinică medicală: aparat cardiovascular*, Editura Orizonturi Universitare, Timisoara (in Romanian).

30. Asadi, H., Dowling, R., Yan, B., Mitchell, P., 2014, *Machine learning for outcome prediction of acute ischemic stroke post intra-arterial therapy*, PLoS ONE, 9(2), paper e88225.

31. Lukić, S., Ćojbasić, Ž., Perić, Z., Milošević, Z., Spasić, M., Pavlović, V., Milojević, A., 2012, *Artificial neural networks based early clinical prediction of mortality after spontaneous intracerebral hemorrhage*, Acta Neurologica Belgica, 112(4), pp. 375-382.

32. Precup, R.-E., Teban, T.-A., Petriu, E.M., Albu, A., Mituletu, I.-C., 2018, *Structure and evolving fuzzy models for prosthetic hand myoelectric-based control systems,* Proc. 26th Mediterranean Conference on Control and Automation, Zadar, Croatia, pp. 625-630.

33. Hochreiter, S., Schmidhuber, J., 1997, *Long short-term memory,* Neural Computation, 9(8), pp. 1735-1780.

34. Gers, F.A., Schmidhuber, J., Cummins, F., 2000, *Learning to forget: continual prediction with LSTM*, Neural Computation, 12(10), pp. 2451-2471.

35. Precup, R.-E., Preitl, S., 1997, *Popov-type stability analysis method for fuzzy control systems*, Proc. Fifth European Congress on Intelligent Technologies and Soft Computing, Aachen, Germany, 2, pp. 1306-1310.

36. Angelov, P., Victor, J., Dourado, A., Filev, D., 2004, *On-line evolution of Takagi-Sugeno fuzzy models*, IFAC Proceedings Volumes, 37(16), pp. 67-72.

37. Precup, R.-E., Preitl, S., Balas, M., Balas, V., 2004, *Fuzzy controllers for tire slip control in anti-lock braking systems,* Proc. IEEE International Conference on Fuzzy Systems, Budapest, Hungary, 3, pp. 1317-1322.

38. Precup, R.-E., Tomescu, M.L., Preitl, S., Petriu, E.M., Fodor, J., Pozna, C., 2013, *Stability analysis and design of a class of MIMO fuzzy control systems*, Journal of Intelligent and Fuzzy Systems, 25(1), pp. 145-155.

39. Blažič, S., Škrjanc, I., Matko, D., 2014, *A robust fuzzy adaptive law for evolving control systems*, Evolving Systems, 5(1), pp. 3-10.

40. Andoga, R., Fozo, L., 2017, *Near magnetic field of a small turbojet engine*, Acta Physica Polonica A, 131(4), pp. 1117-1119.

41. Baranyi, P., Tikk, D., Yam, Y., Patton, R.J., 2003, *From differential equations to PDC controller design via numerical transformation*, Computers in Industry, 51(3), pp. 281-297.

42. Precup, R.-E., Tomescu, M.L., Preitl, S., 2007, *Lorenz system stabilization using fuzzy controllers*, International Journal of Computers Communications and Control, 2(3), pp. 279-287.

43. Navarro, G., Umberger, D.K., Manic, M. 2017, *VD-IT2, Virtual Disk cloning on disk arrays using a type-2 fuzzy controller*, IEEE Transactions on Fuzzy Systems, 25(6), pp. 1752-1764.

44. Alvarez Gil, R.P., Johanyák, Z.C., Kovács, T., 2018, *Surrogate model based optimization of traffic lights cycles and green period ratios using microscopic simulation and fuzzy rule interpolation*, International Journal of Artificial Intelligence, 16(1), pp. 20-40.

45. Rotariu, C., Manta, V., Costin, H., 2012, *Wireless remote monitoring system for patients with cardiac pacemakers*, Proc. 2012 IEEE International Conference and Exposition on Electrical and Power Engineering, Iasi, Romania, pp. 845-848.

46. Haidegger, T., Kovács, L., Precup, R.-E., Benyó, B., Benyó, Z., Preitl, S., *Simulation and control for telerobots in space medicine*, Acta Astronautica, 181(1), pp. 390-402.

47. Costin, H., 2013, *Fuzzy rules-based segmentation method for medical images analysis*, International Journal of Computers Communications and Control, 8(2), pp. 196-205.

48. Takács, Á., Kovács, L., Rudas, I.J., Precup, R.-E., Haidegger, T., 2015, *Models for force control in telesurgical robot systems*, Acta Polytechnica Hungarica, 12(8), pp. 95-114.

49. Belean, B., Streza, M., Crisan, S., Emerich, S., 2017, *Dorsal hand vein pattern analysis and neural networks for biometric authentication*, Studies in Informatics and Control, 26(3), pp. 305-314.

50. Kovács, L., 2017, *A robust fixed point transformation-based approach for type 1 diabetes control*, Nonlinear Dynamics, 89(4), pp. 2481-2493.

51. Melin, P., Miramontes, I., Prado-Arechiga, G., 2018, *A hybrid model based on modular neural networks and fuzzy systems for classification of blood pressure and hypertension risk diagnosis*, Expert Systems with Applications, 107, pp. 146-164.
52. Precup, R.-E., Teban, T.-A., Albu, A., Szedlak-Stinean, A.-I., Bojan-Dragos, C.-A., 2018, *Experiments in incremental online identification of fuzzy models of finger dynamics*, Romanian Journal of Information Science and Technology, 21(4), pp. 358-376.
53. Korondi, P., Hashimoto, H., Gajdar, T., Suto, Z., 1996, *Optimal sliding mode design for motion control*, Proc. 1996 IEEE International Symposium on Industrial Electronics, Warsaw, Poland, pp. 277-282.
54. Filip, F.G., 2008, *Decision support and control for large-scale complex systems*, Annual Reviews in Control, 32(1), pp. 61-70.
55. Antić, D., Nikolić, S., Milojković, M., Danković, N., Jovanović, Z., Perić, S., 2011, *Sensitivity analysis of imperfect systems using almost orthogonal filters*, Acta Polytechnica Hungarica, 8(6), pp. 79-94.
56. Osaba, E., Yang, X.-S., Diaz, F., Onieva, E., Masegosa, A., Perallos, A., 2017, *A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy*, Soft Computing, 21(18), pp. 5295-5308.
57. Nikolić, V., Milovančević, M., Petković, D., Jocić, D., Savić, M., 2018, *Parameters forecasting of laser welding by the artificial intelligence techniques*, Facta Universitatis, Series: Mechanical Engineering, 16(2), pp. 193-201.
58. Dumitrache, I., Constantin, N., Drăgoicea, M., 1996, *Retele neurale: identificarea si conducerea proceselor*, Matrix Rom, Bucharest (in Romanian).
59. Alique, A., Haber, R.E., Haber, R.H., Ros, S., Gonzalez, C., 2000, *Neural network-based model for the prediction of cutting force in milling process. A progress study on a real case*, Proc. 15th IEEE International Symposium on Intelligent Control, Patras, Greece, pp. 121-125.
60. Azadeh, A., Babazadeh, R., Asadzadeh, S.M., 2013, *Optimum estimation and forecasting of renewable energy consumption by artificial neural networks*, Renewable and Sustainable Energy Reviews, 27, pp. 605-612.
61. Tran, T.V., Wan, Y.N., 2017, *Artificial chemical reaction optimization algorithm and neural network based adaptive control for robot manipulator*, Control Engineering and Applied Informatics, 19(2), pp. 61-70.
62. Pozna, C., Precup, R.-E., Tar, J.K., Škrjanc, I., Preitl, S., 2010, *New results in modelling derived from Bayesian filtering*, Knowledge-Based Systems, 23(2), pp. 182-194.
63. Precup, R.-E., Preitl, S., 2003, *Development of fuzzy controllers with non-homogeneous dynamics for integral-type plants*, Electrical Engineering, 85(3), pp. 155-168.
64. Niu, B., Fan, Y., Wang, H., Li, L., Wang, X., 2011, *Novel bacterial foraging optimization with time-varying chemotaxis step*, International Journal of Artificial Intelligence, 7(A11), pp. 257-273.
65. Khmelev, A., Kochetov, Y., 2015, *A hybrid local search for the split delivery vehicle routing problem*, International Journal of Artificial Intelligence, 13(1), pp. 147-164.
66. Mls, K., Cimler, R., Vaščák, J., Puheim, M., *2017, Interactive evolutionary optimization of fuzzy cognitive maps*, Neurocomputing, 232, pp. 58-68.
67. Li, Y.-Q., Hou, Z.-S., Feng, Y.-J., Chi, R.-H., 2017, *Data-driven approximate value iteration with optimality error bound analysis*, Automatica, 78, pp. 79-87.
68. Precup, R.-E., David, R.-C., Szedlak-Stinean, A.-I., Petriu, E.M., Dragan, F., 2017, *An easily understandable grey wolf optimizer and its application to fuzzy controller tuning*, Algorithms, 10(2), pp. 1-15.