Universitatea
Transilvania
din Brașov

INTERDISCIPLINARY DOCTORAL SCHOOL

Faculty of Electrical Engineering and Computer Science

Horia Alexandru MODRAN

Systems based on Artificial Intelligence in Advanced Signal Processing

Sisteme bazate pe Inteligență Artificială în procesarea avansată a semnalelor

SUMMARY

Scientific supervisor

Prof. univ. dr.fiz. Doru URSUȚIU

BRAȘOV, 2023

Content

INTRODUCTION

Artificial Intelligence (AI) is the most fascinating and challenging technology of the current decade, and at the heart of the fourth Industrial Revolution *(Industry 4.0)*. Although it has had its ups and downs over the years, it has now become very popular, thanks to its ability to mimic human intelligence in many common tasks or to support it.

This current thesis follows an approach that presents different applications where the use of such technologies has been implemented or is proposed for future implementation; it also provides a brief survey of the advanced signal processing and artificial intelligence modalities that are used in such applications. Thus, the PhD thesis focuses on the application of Artificial Intelligence and Machine Learning techniques in general signal processing, with particularization in audio, biomedical and physiological signal processing. For conducting this research, the bibliographical sources indicated at the end of the thesis were used and four main directions were followed for the practical experiments carried out:

1. development of an intelligent data acquisition and processing system based on the PSoC 6 chip developed by Infineon
2. biomedical and physiological signal processing
    a. predicting cardiovascular disease using Artificial Intelligence techniques
    b. Fatigue detection via a smart watch using machine learning
    c. gesture classification using *edge* processing techniques
3. audio signal processing - connected with the subjects taught in the Master of Music Therapy
    a. recognition of emotions conveyed by music using Deep Learning techniques
    b. analysis of audio signals using deep learning (music genre classification)
    c. detection of musical notes, speech and noise by using edge processing
4. noise reduction in signals
    a. real-time audio noise reduction using Deep Learning
    b. signal noise reduction autoencoder

Scientific objectives of the PhD thesis

The aim of the present work is to develop a system for acquiring and processing data obtained from measurements/experiments and then making predictions or filtering based on Artificial Intelligence techniques to develop machine learning models with wide efficiency and applicability.

This PhD thesis presents a complex software-hardware approach to the realization of a prediction system based on hybrid Artificial Intelligence algorithms, with applications in both biomedical and audio signal processing, as well as general processing and noise reduction of electronic signals.

The objective of the thesis is to design, implement and validate an intelligent signal acquisition system, through which Machine Learning models can be run on the acquired data, such as:

1. prediction model for cardiovascular disease, fatigue and various human activities

2. model for predicting the mood conveyed by music and reducing noise in audio signals
3. noise reduction model for electronic signals

The present work aims at the application of intelligent signal processing, which generated the idea of developing a complete system, including both an advanced data acquisition and processing system, represented by signals, and their analysis using Artificial Intelligence and Machine Learning techniques. For this, Programmable system on a chip (PSoC) hardware systems - namely Infineon's PSoC 6 architectural platform and smart clocks - were used. On the software side, solutions such as the LabVIEW development environment, together with the Deep Learning (DeepLTK) module and Python API and related libraries were used.

The thesis focuses on the development of a system for intelligent analysis and processing of signals, both biomedical, audio and electronic. From this general objective of the thesis the following specific objectives result (Fig. 1):

1. development of an intelligent data acquisition and processing system - capable of taking biomedical, audio and electronic signals, processing them digitally and performing intelligent *edge* prediction
2. creating models based on Artificial Intelligence for the processing of biomedical and physiological data, to be optimized for the developed system
3. building AI-based models for processing audio signals compatible with the intelligent system
4. development of signal noise reduction models optimised for the developed module

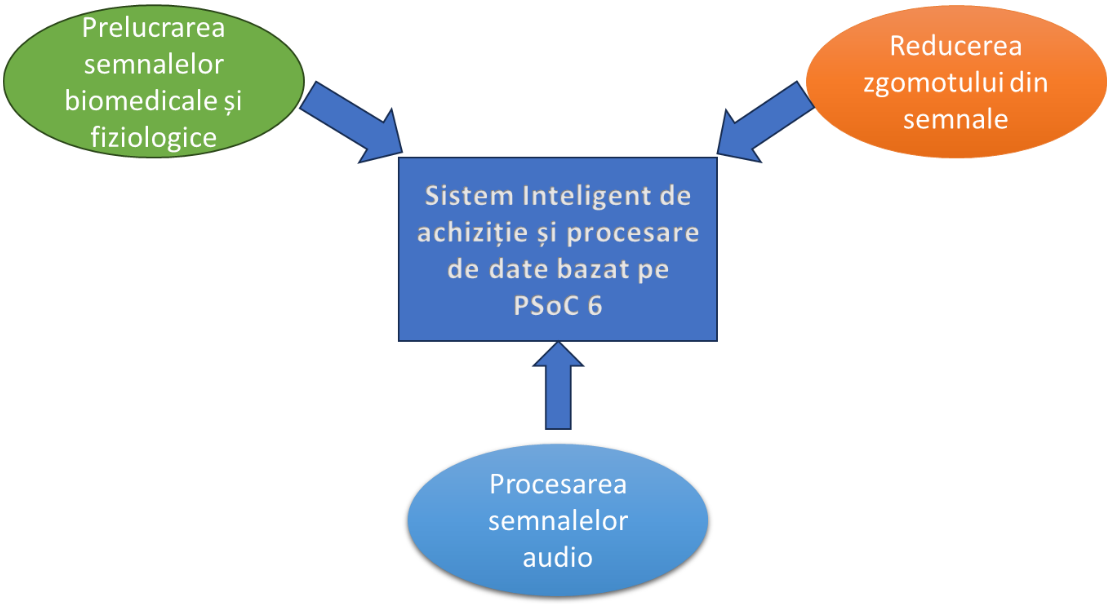


Fig. 1*.* Research Objectives.

Structure of the PhD thesis

The thesis consists of an introduction, followed by 5 Chapters, dealing with the proposed objectives, and the last section of the paper is devoted to conclusions, original contributions and potential for further development, including the dissemination of research results in international publications and conferences.

The introductory part presents an introduction to the topic, the timeliness and topicality of the chosen theme and the scientific objectives of the research undertaken.

- Chapter 1, "**State of the Art in Artificial Intelligence and Signal Processing**" reviews the main existing technologies and solutions for applying Artificial Intelligence in advanced signal processing.
- Chapter 2, "**Development of an intelligent data acquisition and processing system**" presents the development of an intelligent data acquisition and processing system using Artificial Intelligence methods, based on PSoC 6, which is software reconfigurable.
- Chapter 3, "**Applications of Artificial Intelligence in Biomedical and Physiological Signal Processing**" describes the application of Artificial Intelligence algorithms in biomedical and physiological signal processing, focusing on the detection of cardiovascular disease, fatigue using a smart watch, and activity using *edge* processing and prediction techniques.
- Chapter 4, "**Application of Artificial Intelligence in Audio Signal Processing**" illustrates the application of deep learning models in audio signal processing, in particular for the detection moods conveyed by music, music genre classification, as well as musical notes, speech and noise recognition, using edge processing techniques.
- Chapter 5, entitled "**Reducing Signal Noise Using Artificial Intelligence**", presents experiments conducted to suppress noise in audio signals and develop a model for reducing noise and fluctuations in signals.
- The last chapter of the thesis presents the overall conclusions of the work, the author's original contributions, the potential for further development of the original contributions and the dissemination of the scientific results in international journals and conferences, in which the author has published the results of research during his doctoral studies.
- The bibliography includes 102 works in the field of the PhD thesis.

1. DEVELOPMENT OF A RECONFIGURABLE INTELLIGENT DATA ACQUISITION AND PROCESSING SYSTEM

The current chapter presents the development of an intelligent data acquisition and processing system using Artificial Intelligence methods. The developed hardware system is software reconfigurable, combining the software configurability provided by the Programmable System on a Chip (PSoC) ecosystem from Infineon (formerly Cypress).

1.1. First prototype - PSoC 6 circuit printed using Voltera V-One

In the first stage of development, the aim was to produce a small printed circuit board (PCB) that integrates the PSoC 6 BLE Prototyping Kit (CY8CPROTO-063-BLE) platform and offers the possibility to acquire various signals through it [74].

The circuit was developed using Proteus software. The basic features of the circuit are as follows (Fig. 2):

- 6 PSoC connectors (21 pins for the left side and 21 for the right side)
- 2 resistors (10k and 100k respectively) and 2 BNC input connectors
- 10x and 100x amplifier with a switch between them

The role of this circuit is to integrate the PSoC for data acquisition via BNC connectors and to amplify the acquired data by 10, respectively 100 times. After designing the circuit in Proteus, Gerber files were generated in order to print the circuit using Voltera V-One.

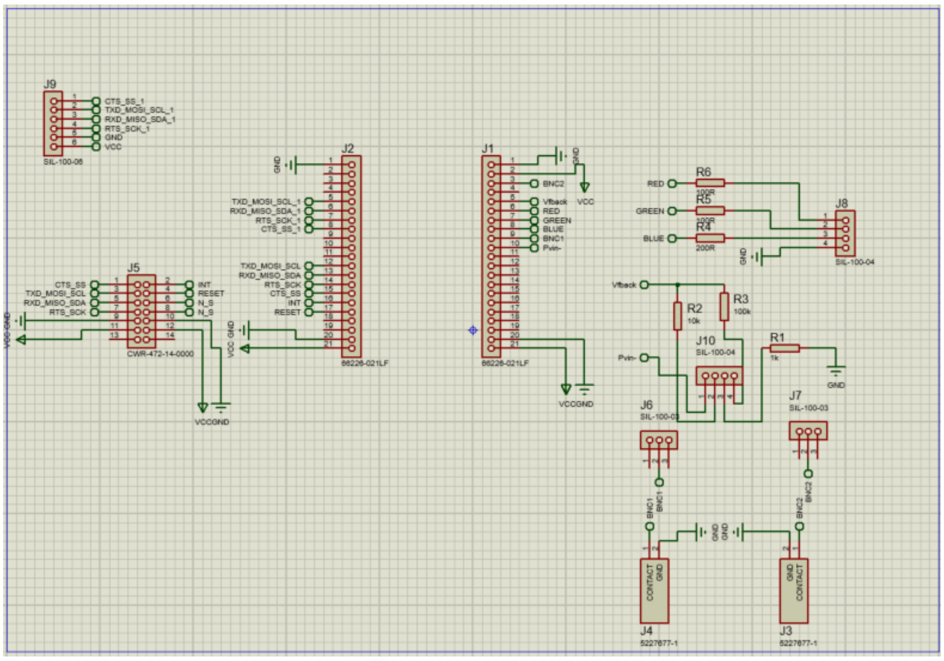


Fig. 2. Design of the PCB.

After generating the Gerber file, the first step in making the circuit is the *drilling* step, which is the precursor to the printing step. Holes are drilled all over the printed circuit board and are used for internal electrical interconnection or used to position components on the board.

After the drilling, printing and drying stages, the components are placed by hand. The following components were placed on the PCB circuit board (Fig. 3):

- prototyping module CY8CPROTO-063-BLE
- 2 BNC connectors
- 2 switches
- 4 Pmod connectors

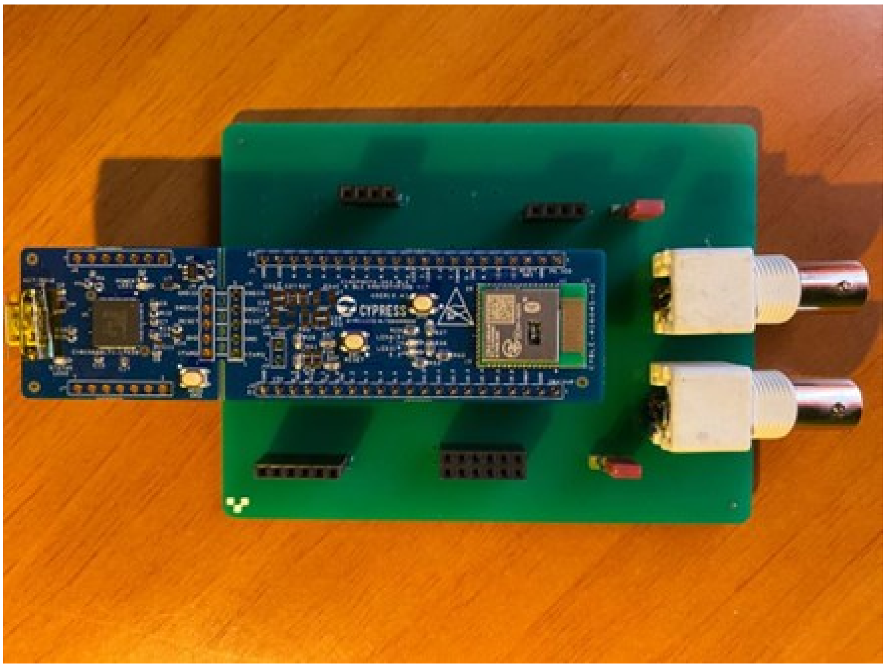


Fig. 3. Placing components on the circuit.

To test the developed prototype, the PSoC6 microcontroller was programmed with a data acquisition application using an internal amplifier and an analog-to-digital converter.

Basic data acquisition was illustrated by using an unamplified and an amplified input signal, which were connected through the BNC connectors of the circuit. The signal was fed into an ADC block and to a Universal Asynchronous *Receiver-Transmitter* (abbreviated UART). The PSoC board communicated with the computer via the COM serial port that was assigned to the device. The design realized in PSoC Creator is illustrated in Fig. 4. The developed application is intended to acquire biomedical and other data, as well as transmit it both via the serial port to the computer and via Bluetooth 5.0 to other IoT devices.

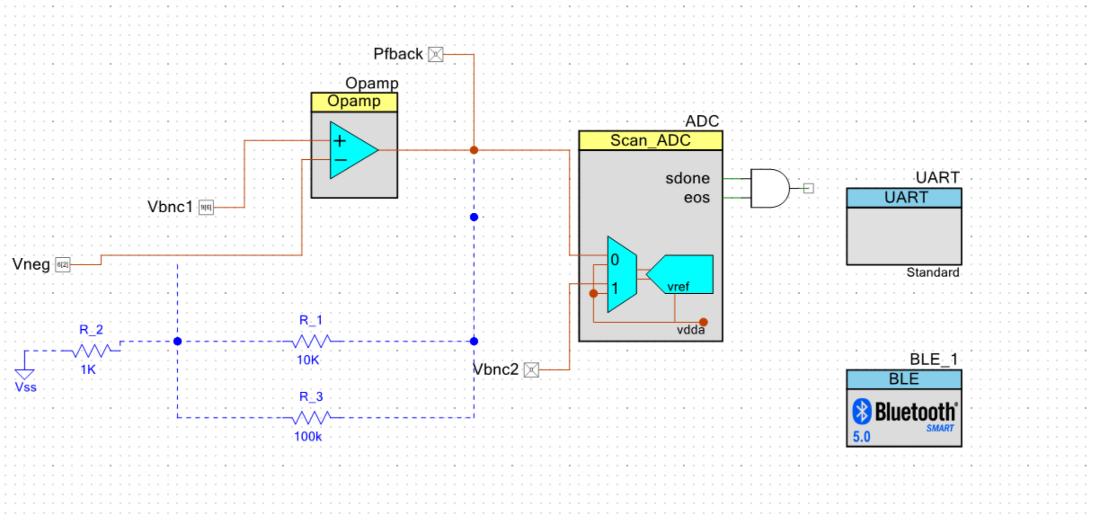


Fig. 4. PSoC Creator diagram.

This application is developed for use with the PCB. Switching between the two types of amplification is done with a switch and the input data can be taken via the two BNC connectors (*Vbnc1* and *Vbnc2*).

For testing the values before and after amplification, a Virtual Instrument (VI) was developed in LabVIEW that acquires the signal from PSoC6 via the Serial port and displays it graphically. A non-inverting amplifier circuit was created by connecting an internal amplifier component to an external resistor network. The NI ELVIS prototyping board was used to generate two comparable sinusoidal signals. The acquired signal was translated by the ADC component and the value was tested for both the unamplified and amplified signals. The front panel, containing the values read and plotted in a Waveform Chart structure for 100x amplification are shown in Fig. 5.

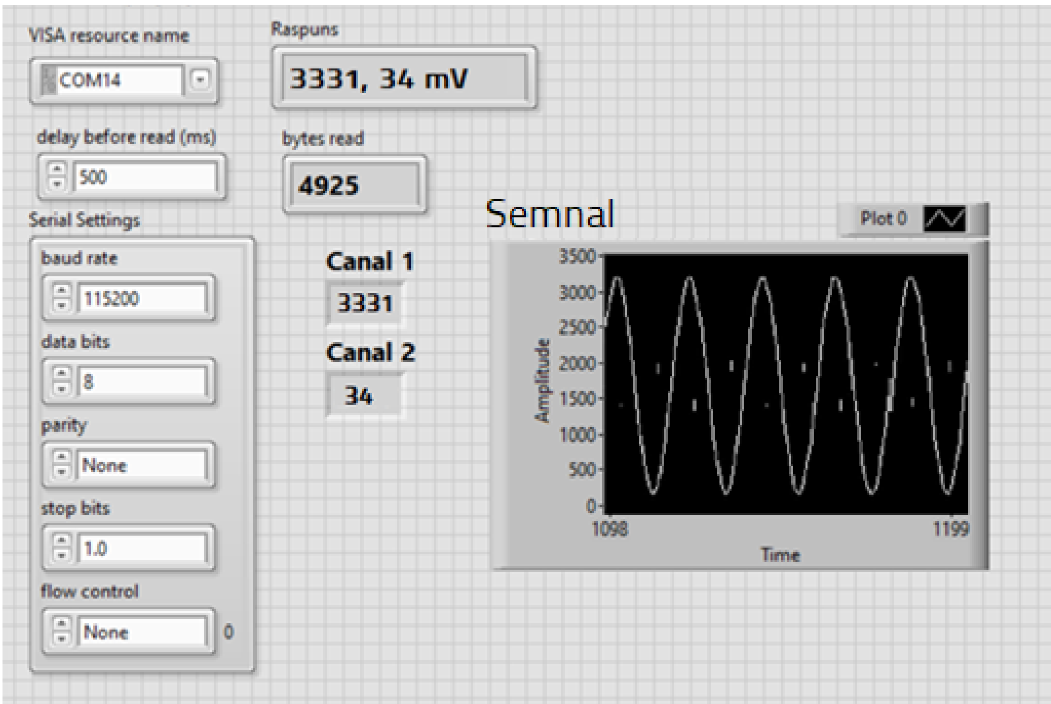


Fig. 5. Front panel of the signal acquisition VI.

The acquired data can then be transmitted using Bluetooth 5.0 technology to a smartphone or any other IoT device that can use the Bluetooth communication protocol.

To communicate with the IoT device, an Android application based on the Model - View - Controller (MVC) architecture was developed. There are seven classes from the Bluetooth library that were used: BluetoohManager, BluetoothAdapter, BluetoothLeScanner, BluetoothGatt, BluetoothGattCallback, BluetoothGattCharacteristic and finally BluetoothGattDescriptor.

The role of classes is as follows [74]:

1. BluetoothManager is used to find the Bluetooth system service for the phone running the application

2. BluetoothAdapter is the Bluetooth adapter. This object is used to interact with the PSoC device and is found using the getAdapter method in the BluetoothManager.

3. BluetoothLeScanner is used to find BLE (Bluetooth Low Energy) devices that the phone can interact with. It is a listener for BLE advertising packets and then notifies of their existence.

4. BluetoothGatt is used to enable communication with a BLE device. This object is created when connecting to a BLE device or BLE peripheral.

5. BluetoothGattCallback is used to record events on the BLE device, such as connection status changes and new available data.

6. BluetoothGattCharacteristic is used for each of the features contained in the BLE device profile.

7. BluetoothGattDescriptor is used for additional feature attributes. In our case, it will be used to contain the notification descriptor for the feature that transmits the values obtained by PSoC6 from the data acquisition channels.

The user interface of the Android app on the phone is illustrated in Fig. 6.

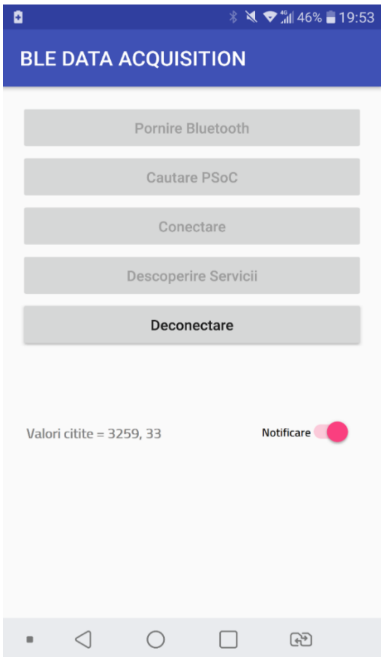


Fig. 6. UI interface of the Android app.

1.2. Second experimental module - Intelligent system

In this second phase, a smart configurable board based on the PSoC 6 CYBLE-416045-02 chip has been developed with various peripherals connected. This system provides a robust intelligent data acquisition and processing platform built around the scalable and reconfigurable PSoC CYBLE-416045-02 EZ-BLE module.

Proteus software was used to design the system. The power supply of the board is from Li-Ion battery, including a 5V and a 3.3V regulator. In addition, a power supply test component has been included to check that the power supply is working properly. The USB Type-C connector offers the possibility to program the PSoC 6 microcontroller and allows a serial connection from the board to the computer, through which acquired data can be transmitted to the PC. Programming the microcontroller also required the inclusion of a PSoC5 chip to act as a KitProg2 - which is basically a programmer/debugger with USB-I2C and USB-UART bridge functionality. A microBNC connector and an external TSV7722IDT operational amplifier (OpAmp) were integrated for signal acquisition. It has a high-precision input offset voltage, wide supply voltage range (1.8V - 5V) and a rail-to-rail output. The CYBLE-416045-02 chip has also been integrated, around which virtually the entire system has been developed. Via the USB-C connection it can be programmed and reconfigured, and is capable of being programmed with various Artificial Intelligence models using the ModusToolbox ecosystem.

After completion of the board prototype, Gerber files were generated for both the front of the board (Fig. 7) and the back of the board (Fig. 8), and all existing connections on the board were checked in detail.

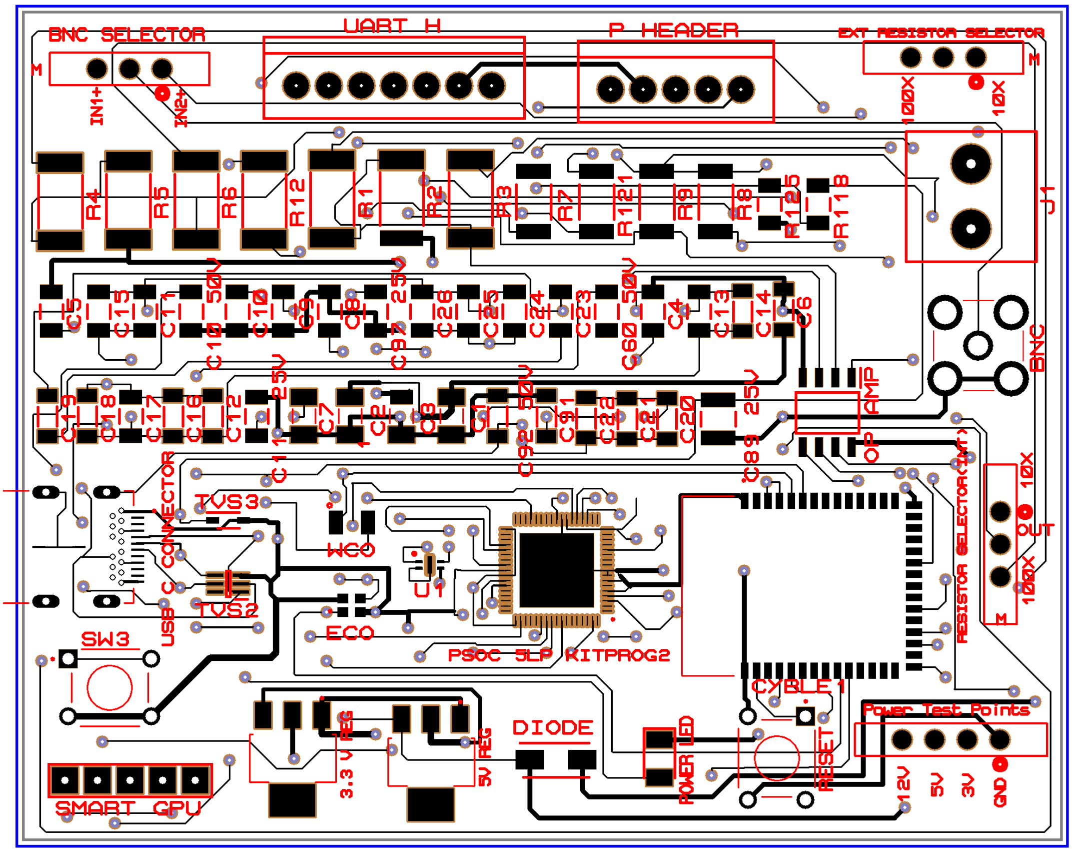


Fig. 7. PCB Prototype (front).

First the top layer of the plate was exposed and checked (Fig. 6), then the bottom layer (Fig. 7).

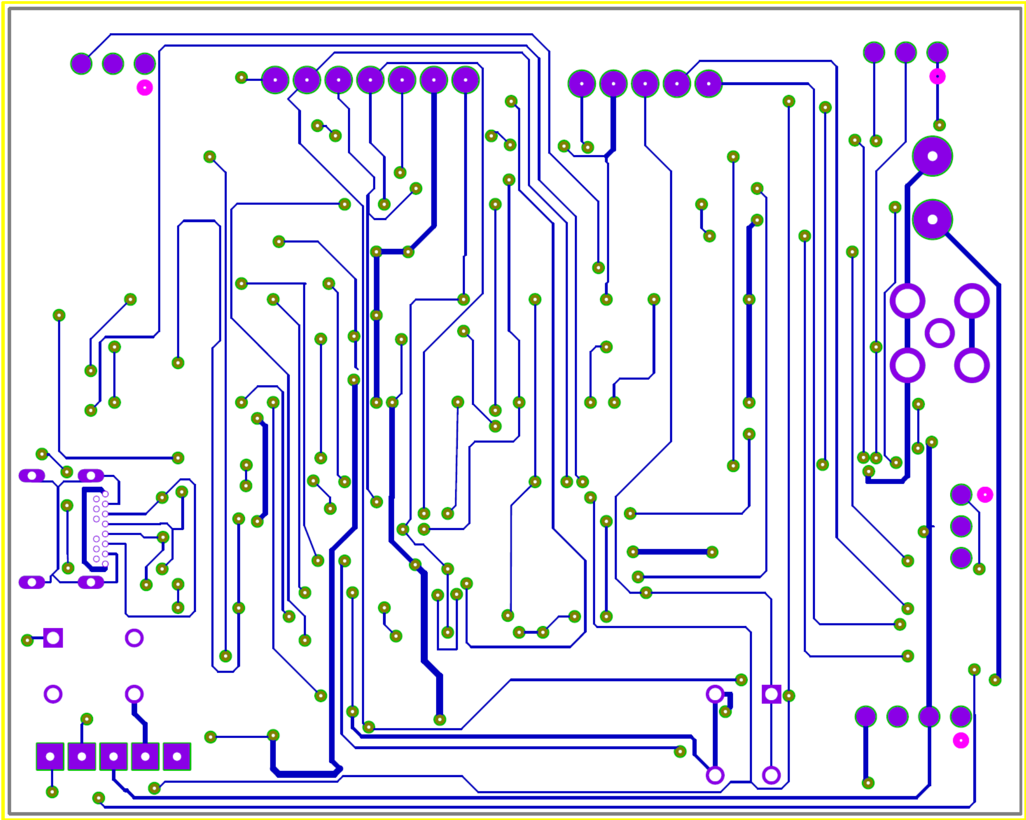


Fig. 8. PCB Prototype (reverse).

After making the board, all components were then placed on the board and it was then powered and programmed to check all its functions (Fig. 9).

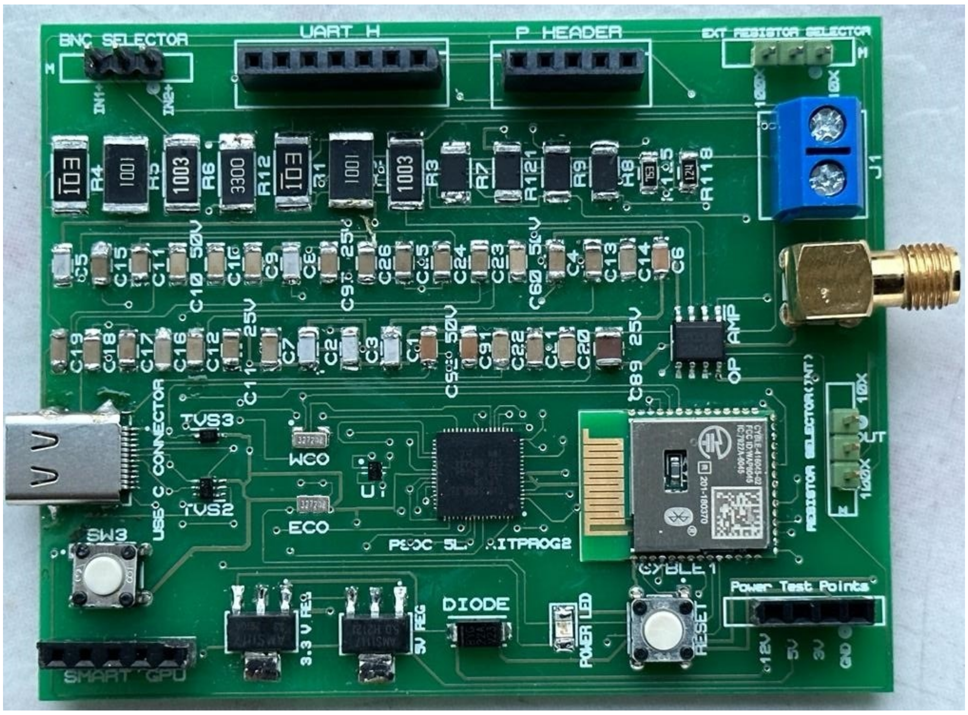


Fig. 9. Final version of the system.

A flexible data acquisition application has been developed for this board using the ModusToolbox ecosystem. This application uses the ADC HAL (Hardware Abstraction Library) libraries to perform signal acquisition. The Analog-to-Digital Converter component of the HAL library is configured to periodically sample the input voltage on a single differential channel and display the sampled voltage on the UART. In addition, the application is flexible and both the sampling rate and the number of points to be acquired can be configured.

The user can select the number of points to be purchased in a single reading, with the following options:

- 128 points (key 1)
- 256 points (key 2)
- 512 points (key 3)
- 1024 points (key 4)

The input voltage is sampled at each user-specified interval by calling *cyhal_adc_read_uv*, then each read value is converted to millivolts (Fig. 10). Also, depending on the number of samples selected the result will be concatenated into a string, which at the end of reading all values will be transmitted through the UART component.

```
void adc_single_channel_acquire(char res[])
{
    for(int i = 0; i < NO_OF_SAMPLES; i++) {
        //variabilă pentru a stoca valoarea citită
        int32_t adc_result_0 = 0;

        //citire voltage, convertire la milivolts
        adc_result_0 = cyhal_adc_read_uv(&adc_chan_0_obj) / MICRO_TO_MILLI_CONV_RATIO;
        char val[100] = {};
        //concatenare valoare citită
        sprintf(val, "%d,", adc_result_0);

        strcat(res, val);

        //asteptare rată de esantionare
        cyhal_system_delay_ms(ADC_SCAN_DELAY_MS);
    }
}
```

Fig. 10. Pins connected to the ADC component.

A LabVIEW Virtual Instrument was also developed through which data could be acquired and both the number of points and the sampling rate could be specified via the front panel.

1.3. Chapter summary. Dissemination

The objective of this chapter was to develop a smart board, reconfigurable in terms of software and functionality. Being based on the PSoC 6 chip, the CYBLE-416045-02 takes full advantage of the benefits it offers, namely small size, dual-core architecture, Bluetooth and serial connection, security and compatibility with a wide range of peripherals. Few examples of flexible experimental data acquisition and processing applications have been presented in this chapter to illustrate the reconfigurability of the system.

In the process of developing the final version of the board there were two different stages of development: a first prototype integrating the PSoC 6 BLE Prototyping Kit platform (CY8CPROTO-063-BLE), and a second stage of development a PCB board including only the PSoC6 chip.

The final version of the developed system includes:

- CYBLE-416045-02 dual-core microcontroller
- USB type C connection
- Bluetooth BLE communication
- Pmod connector, for connection to various peripheral modules or sensors
- software reconfigurability
- microBNC connector for signal acquisition
- serial connection via UART (to USB-C port)
- external operational amplifier
- battery power

A mobile application on the Android operating system has also been developed, using the MVC architecture, which can communicate via Bluetooth with this board. In addition, Artificial Intelligence models presented in the following chapters, being mostly developed using the Tensorflow library, can be exported, programmed and run on this board.

The experiment described in this chapter presented at the International Conference on Interactive Mobile Communication Technologies and Learning (IMCL) 2021, held in Thessaloniki (Greece) on 4-5 November 2021, and published in the conference volume -SpringerLink, which is ISI indexed:

- **H. A. Modran**, D. Ursutiu, C. Samoila, T. Chamunorwa: "Intelligent IoT Biomedical Bluetooth Data Acquisition System", In New Realities, Mobile Systems, and Applications - Proceedings of the International Conference on Interactive Mobile Communication, Technologies, and Learning (IMCL) 2021, http://dx.doi.org/10.1007/978-3-030-96296-8_88

2. APPLICATIONS OF ARTIFICIAL INTELLIGENCE IN BIOMEDICAL AND PHYSIOLOGICAL SIGNAL PROCESSING

2.1. Detecting cardiovascular pathologies using IoT and AI

The approach used in this study is based on the development of a blood pressure data collection and processing system using machine learning algorithms [75]. The acquired data are obtained using a powerful and inexpensive IoT device based on the PSoC6 microcontroller integrated circuit from Infineon company. The data is then processed inside a virtual instrument (VI) created in the Laboratory for Virtual Engineering Instrumentation (LabVIEW) using a machine learning model developed in Python. The most widely used machine learning algorithms that are suitable for a classification problem were compared and the one with the best evaluation metrics was chosen. Thus, the developed system uses a multilayer perceptron neural network to predict whether a person has a 5-year risk of developing cardiovascular disease [76]. A dataset consisting of 70,000 data was used to train the model. Compared to similar studies [77], the developed machine learning model achieved higher performance indicators and accuracy. The subjects on whom the system was tested signed an IRB agreement according to the General Data Protection Regulation (GDPR) and the data was not stored.

The PSoC6 BLE microcontroller was programmed to obtain blood pressure data in PSoC Creator using the C programming language. For the schematic (Fig. 11) a BLE component was used to send data using Bluetooth 5.0 and a UART component was used to print data via the serial port of the computer. The processors communicate with each other using a mutex to control access to a shared variable. With the built-in security provided by PSoC6, the IoT system is thus protected.

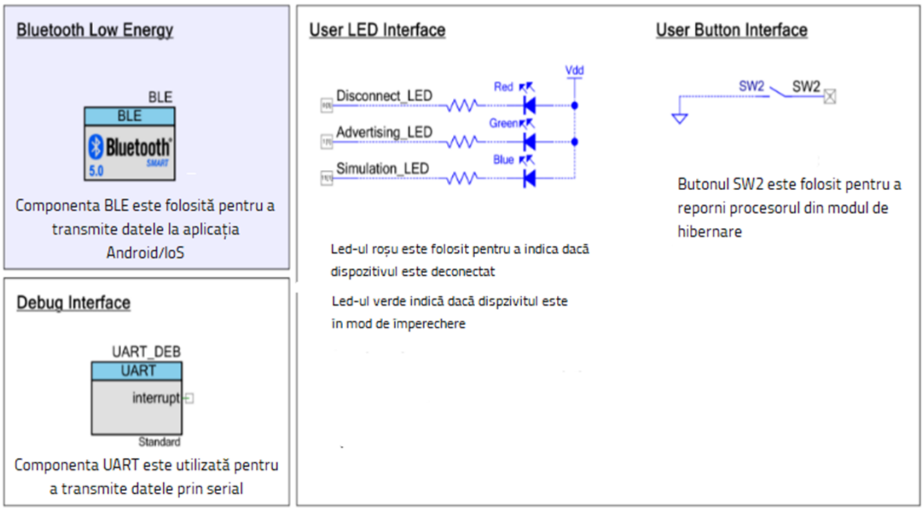


Fig. 11. PSoC schematic of the IoT system.

The IoT system can be controlled via Bluetooth using an Android or IoS smartphone, while systolic and diastolic blood pressure acquired by the PSoC 6 device can be tracked on the smartphone in real time (Fig. 12).

Fig. 12. Android app for system control.

The data used to build the model can sometimes come from multiple distributions. There are three main datasets used at different stages of building a machine learning model: the training dataset, the validation dataset and the testing dataset, therefore the dataset has been divided into these 3 categories [79]. The size of each of these three datasets can be chosen depending on the typology of the problem [80], but as a common approach, it is recommended that the training dataset should contain approximately 60% of the original dataset, while the validation and training subsets should each contain 20%.

The data acquisition process can be controlled via the mobile app connected via Bluetooth to the PSoC microcontroller. The PSoC6 kit has been programmed to acquire blood pressure data and display it graphically using the Universal Asynchronous Receiver-Transmitter (UART) debugging component of the PSoC Creator scheme at regular intervals via the serial port. To enable the debug port, the DEBUG_UART_ENABLED macro was set to ENABLED.

To see the data displayed on the terminal, the user should connect to the serial port using Putty and it will display the acquired values in the terminal. This data can also be available in the LabVIEW environment, for which a Virtual Instrument (VI) has been developed which uses the VISA function to connect to the serial port of the computer.

For the development of the machine learning model, the Python programming language was used together with the NumPy and Pandas libraries. Since the model is trained with a dataset containing labeled data, a supervised learning algorithm should be used. To choose the appropriate supervised learning algorithm for this model, the most commonly used algorithms were compared. Since neural networks have better prediction speed than SVMs, the multilayer perceptron neural network (MLP) was selected for the model (Fig. 13). The model that is used contains several features, equal to the number of features of the training dataset - the most important of which are age, systolic and diastolic blood pressure, and 3 hidden layers.

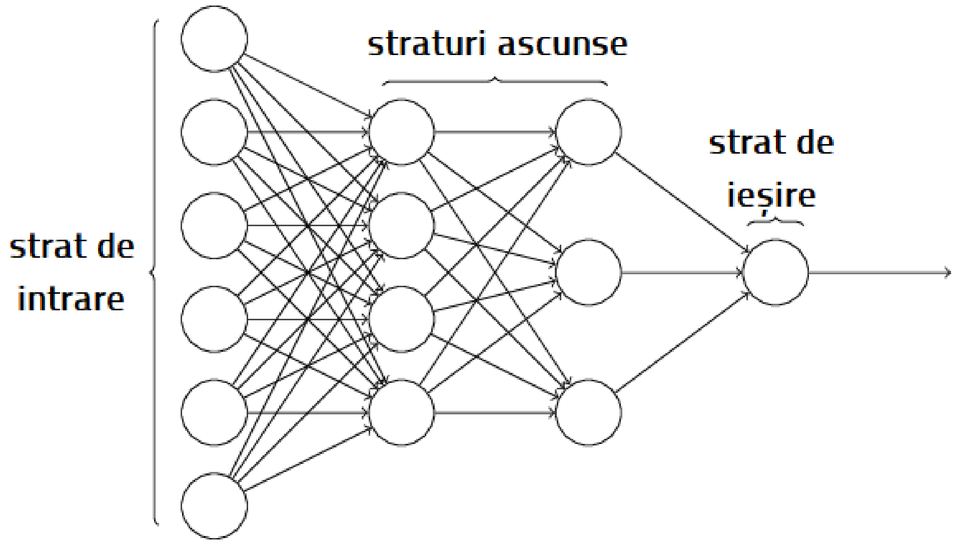


Fig. 13. MLP neural network architecture.

To import the machine learning model into LabVIEW, a subVI was created using the Python node to get a Python session. This subVI was then used in the main virtual tool which aims to acquire data from the IoT device, send it to the Python module and display the prediction made by the machine learning model. While the default threshold for interpreting probabilities for labels in classification problems is usually 0.5 [83], in medical diagnostics it is very important that predictions are as accurate as possible. Therefore, in this model the confidence threshold for prediction was set to 0.7. The front panel of this VI (Fig. 14) also displays the response received from the PSoC6 chip and the bytes read from them for debugging purposes. Furthermore, the right side also displays systolic and diastolic blood pressure.

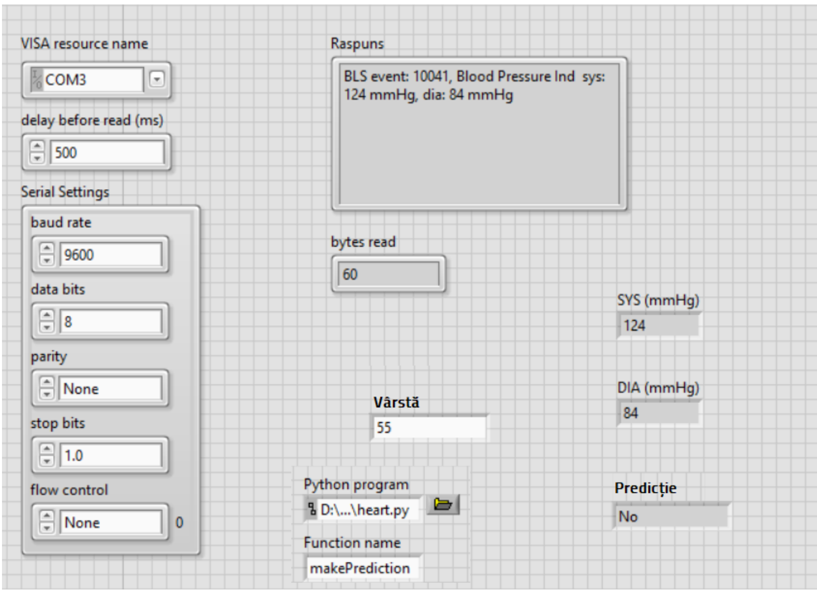


Fig. 14. Front panel of the prediction IV.

2.2. Fatigue detection using wearable devices

Fatigue is a common non-specific condition that is identified with reduced work capacity and motivation to perform certain tasks. Accumulation of fatigue can cause chronic fatigue syndrome, disorders, dysfunctions or diseases on human health. Fatigue is classified as:

a) physical when it affects muscle movement performance
b) mental if it is related to overwork or work under pressure and stress
c) emotional fatigue, which is known as long-term or chronic fatigue, with an inability to maintain mental focus [84].

Machine learning (ML) techniques are applied to recognize daily human activities and assess tremor severity, as well as to interpret tremor data and predict fatigue. For tremor compensation, distinct and accurate recognition and separation of involuntary hand movements in real time are required.

The current study aims to develop a non-invasive model for estimating fatigue in real time, based on typical signs of fatigue such as tremor and heart rate. The objectives of the research are to distinguish early signs of fatigue and exhaustion in order to prevent their adverse effects using movement and physiological data obtained by using a *smartwatch as* a specialized sensor. Advanced machine learning techniques will be applied to create a mobile app that will warn the user when the first signs of fatigue appear.

A group of 10 volunteers of various ages took part in this study. They wore the watch for a week during their daily activities and also kept a daily diary of their activities. An IRB agreement was concluded with them in advance and the data acquired and stored complies with the General Data Protection Regulation (GDPR).

A compact, high-precision accelerometer integrated into the smartwatch was used to capture physiological low-frequency tremor. The optical sensor on the smartwatch collected heart rate data. The collected data is saved directly to the watch as a CSV file as well as to the cloud, and a machine learning model will be developed that predicts fatigue based on the values of tremor and heartbeat. To collect accelerometer and heart rate data continuously in real time, the TicWatch E[1] and Polar M600[2] smartwatches, which combine both sensors, were used. The Android app developed for the smartwatch records and graphically displays the data.

An Android app has been developed to collect physiological data (heart rate and tremor) that can also be run on a smartwatch. This app is intended to display graphically on the screen and store locally, in CSV file, and on the cloud the acquired data [88].

The developed app can be used both on smartwatches with round screen (TicWatch) and square screen (Polar) (Fig. 15).

---

[1] https://www.mobvoi.com/af/pages/ticwatchse
[2] https://support.polar.com/en/support/m600

Fig. 15. Android App on TicWatch.

The heart rate data collected are recorded once per minute (Fig. 16 a) and the tremor data at a frequency of 50Hz (Fig. 16 b) and are stored in separate CSV files together with the date and time they were recorded by the subject on a normal working day.

1	Data	Ritm cardiac
2	14-05-2020 11:23:02	91
3	14-05-2020 11:24:01	113
4	14-05-2020 11:25:01	96
5	14-05-2020 11:27:00	98
6	14-05-2020 11:28:02	99
7	14-05-2020 11:29:02	102
8	14-05-2020 11:30:01	116
9	14-05-2020 11:31:02	111
10	14-05-2020 11:32:01	110
11	14-05-2020 11:33:01	102
12	14-05-2020 11:34:01	100
13	14-05-2020 11:35:02	90
14	14-05-2020 11:36:01	102
15	14-05-2020 11:37:01	112
16	14-05-2020 11:38:01	109
17	14-05-2020 11:39:01	128
18	14-05-2020 11:40:01	112
19	14-05-2020 11:41:02	119
20	14-05-2020 11:42:02	103
21	14-05-2020 11:44:02	99

(a) Heart Rate data

	A	B	C	D	E
1	Data	X	Y	Z	
2	14-05-2020 09:23:37	0.54	-10.48	3.56	
3	14-05-2020 09:23:37	0.45	-10.15	2.47	
4	14-05-2020 09:23:37	1.09	-9.01	3.15	
5	14-05-2020 09:23:37	0.93	-7.54	6.44	
6	14-05-2020 09:23:37	0.93	-8.73	4.85	
7	14-05-2020 09:23:37	0.76	-9.02	4.69	
8	14-05-2020 09:23:37	0.43	-7.66	4.75	
9	14-05-2020 09:23:38	0.73	-7.72	5.64	
10	14-05-2020 09:23:38	0.69	-7.48	5.07	
11	14-05-2020 09:23:38	0.44	-9.97	3.38	
12	14-05-2020 09:23:38	0.6	-8.3	5.83	
13	14-05-2020 09:23:38	1.08	-8.56	4.15	
14	14-05-2020 09:23:38	1.27	-6.35	5.54	
15	14-05-2020 09:23:38	0.64	-7.67	4.58	
16	14-05-2020 09:23:38	0.03	-7.5	6.5	
17	14-05-2020 09:23:42	0.21	-8.06	5.53	
18	14-05-2020 09:23:42	0.21	-5.78	6.59	
19	14-05-2020 09:23:42	0.4	-9.05	5.42	
20	14-05-2020 09:23:42	1.91	-8.34	2.46	

(b) Tremor data

Fig. 16. CSV files of heart rate and tremor data.

The app uses a *ForegroundService* to be able to run and acquire data in the background as well, increasing the battery life of the device. It contains a *Heartbeat* service, which measures heart rate data and stores it in a local file and on the Cloud. In addition, it includes an *AccelerometerService* that acquires tremor data and saves it. Because it requires permission to access the device's sensors, the app will prompt the user to grant this permission, only on the first use.

After collecting medical and physiological parameters for several days, the two files containing these data were processed. The data was processed using the Python programming environment and then displayed graphically for both heart rate (Fig. 17) and accelerometer data (Fig. 18). These graphs provide an opportunity to visualize and analyze the processed data, and are a preparatory step before the process of cleaning and splitting the dataset. The tremor data collected by the accelbackground was analysed graphically, both displaying in one graph all 3 axes and in three separate graphs (one for each axis - x, y, z).

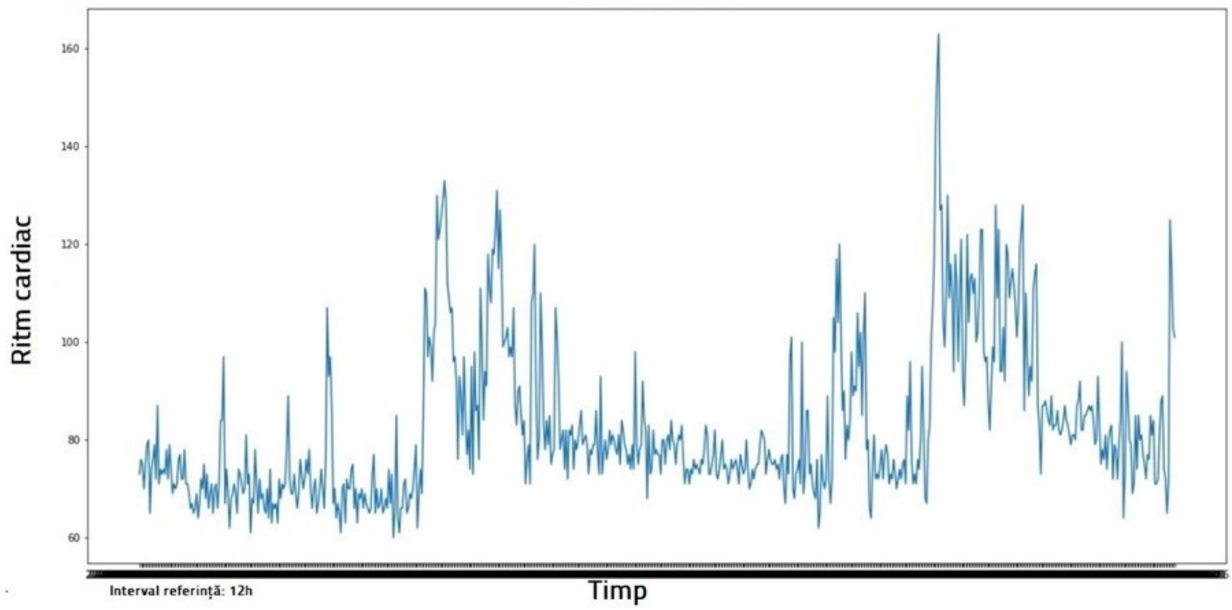


Fig. 17. Heart rate data.

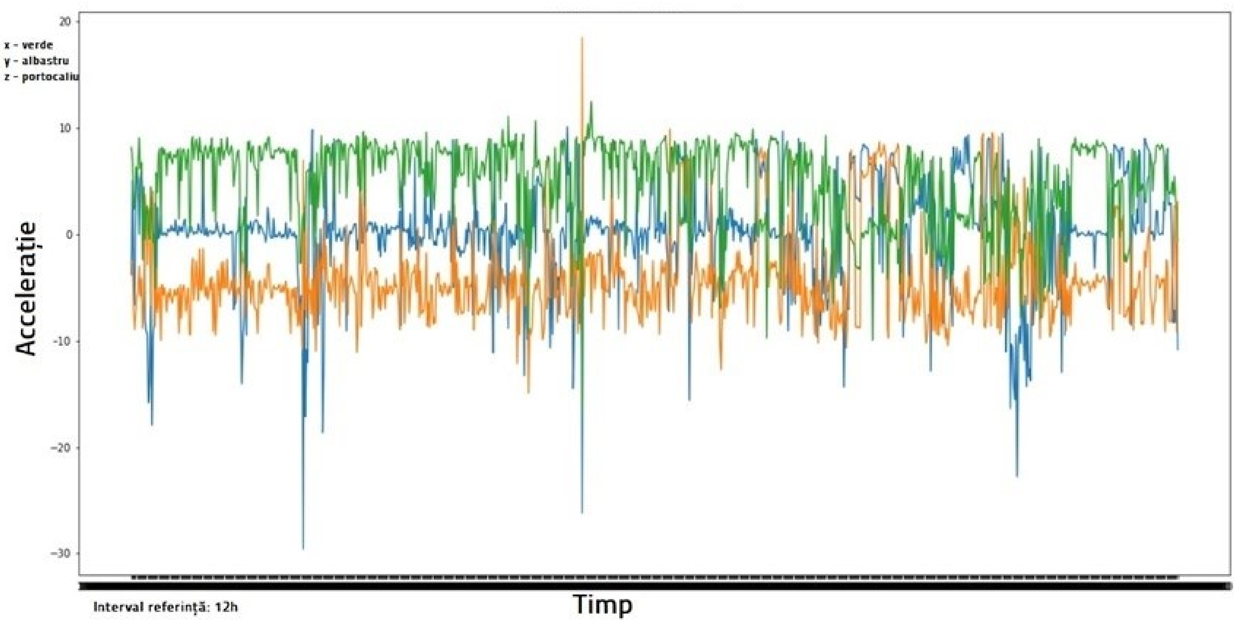


Fig. 18. Accelerometer data.

2.3. Classification of human activities by edge techniques

Typically, the process of developing a human activity recognition model consists of four main steps (Fig. 19):

1. acceleration signal acquisition
2. data pre-processing
3. activity recognition (based on deep learning techniques)
4. user interface for transmission and display of the prediction

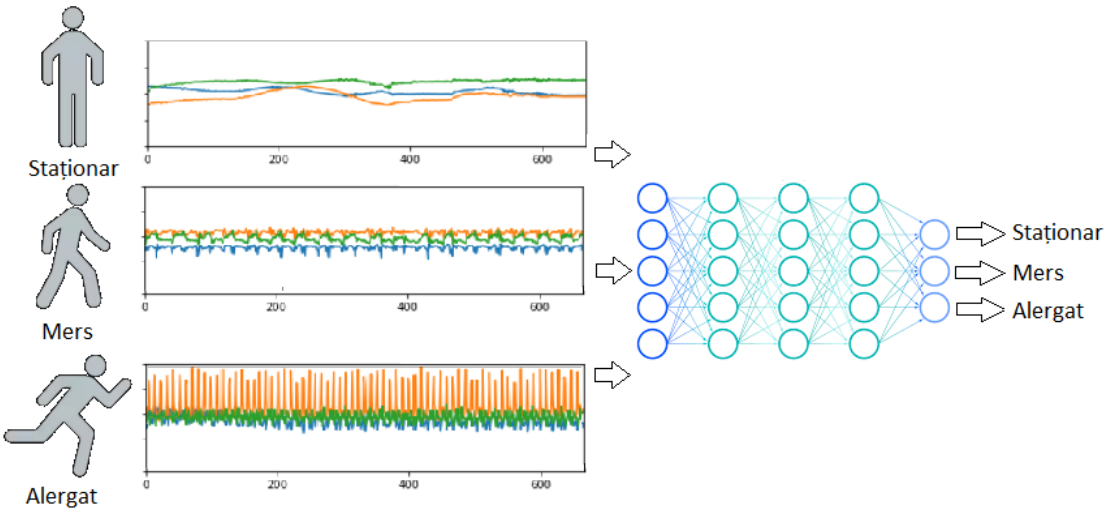


Fig. 19. Stages in the development of a human activity recognition model.

This experiment classifies human activities based on motion sensor data (accelerometer and gyroscope). The model programmed on the IoT device was pre-trained on the computer using Keras and classifies several common activities: stationary, walking and running.

The operation of the application is described in Fig. 20. In an infinite loop, the IoT device reads data from a motion sensor (BMX160) attached to the PSoC6 to detect activities. The dataset consists of 3-axis orientation data from both the accelerometer and the gyroscope. A timer is configured to interrupt at 128 Hz. The interrupt operator reads all 6 axes via SPI and signals a data processing task when the internal buffer has 128 new samples. It performs an IIR filter and min-max normalization on 128 samples simultaneously. This processed data is then passed to the inference processor. The inference processor determines and returns the prediction confidence for each activity class. If the confidence exceeds 80%, the predicted activity is displayed on the UART terminal.

This application uses FreeRTOS. Within it, a task has been defined and run in the system - the activities task, which processes the incoming data and passes it to the ML model.

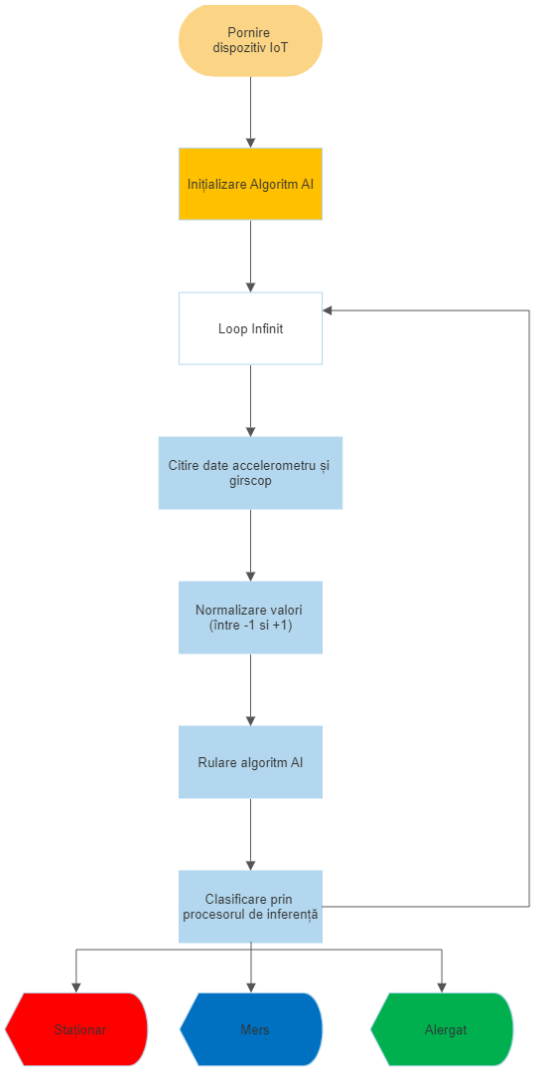


Fig. 20. Different stages in developing the ML model.

Data for training the machine learning model was collected from multiple users during several types of activities via the BMX160 sensor attached to the PSoC6, then tagged by activity and saved to a CSV file.

After collecting the data, a model was developed using that data - a step that includes both training and calibrating the model. For this problem a neural network model was developed in Python using Keras.

The model was then trained using previously collected data using the following features:

- *Adam* optimizer
- learning rate of 0.0001
- metrics: accuracy, confusion matrix
- 20 *epochs* and 1000 steps in each epoch

The confusion matrix was displayed graphically to visualize the classification performance of the model (Fig. 21).

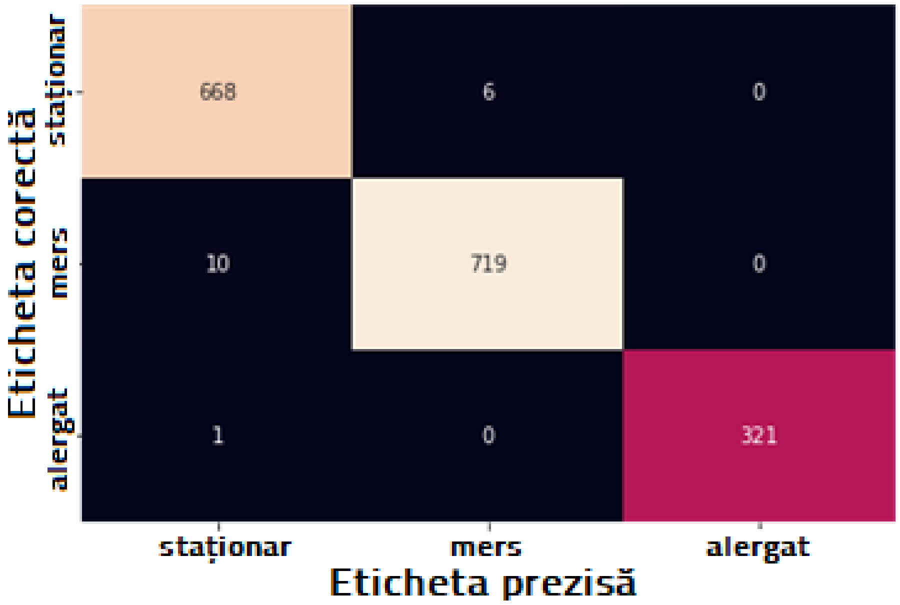


Fig. 21. Model Confusion Matrix.

The model weights and structure were then saved to a file for use in programming the IoT device, and its validation and calibration data were saved to a separate file. Model performance indicators are illustrated in Table 1.

Table 1. Model performance indicators.

Indicator	Value
Loss	0.0169
Accuracy	0.9971
Precision	0.9975
Recall	0.9947
Scor F1	0.9961

The convolutional neural network (CNN) model consists of two convolutional blocks and two fully connected layers. Each convolutional block includes convolutional operations, including rectified linear unit (*ReLU*) and dropout layer, with the addition of a batch flattening (*Flatten*) layer after the first block. The convolutional layers act as feature extractors and provide abstract representations of the input sensor data in the feature map. They capture short-term dependencies (spatial relationships) of the data. In the developed network, features are extracted and then used as inputs to a fully connected network using *Softmax* activation for classification.

After this, the generated model was programmed onto the device for testing. The ML Configurator also shows the resources consumed by the model produced, which are suitable for any microcontroller. The model was validated on the PC, with very good results for both 8x8 and float quantization. A relative accuracy of 100% was obtained and the misprediction error was only 0.001, while the memory used is very small (approx. 16 kB).

Before being programmed on the PsoC6 device, the model must be checked to validate that it is optimised for the hardware available on that device. Following the validation performed in ML Configurator, 100% accuracy and 0.01 prediction error was obtained (Fig. 22).

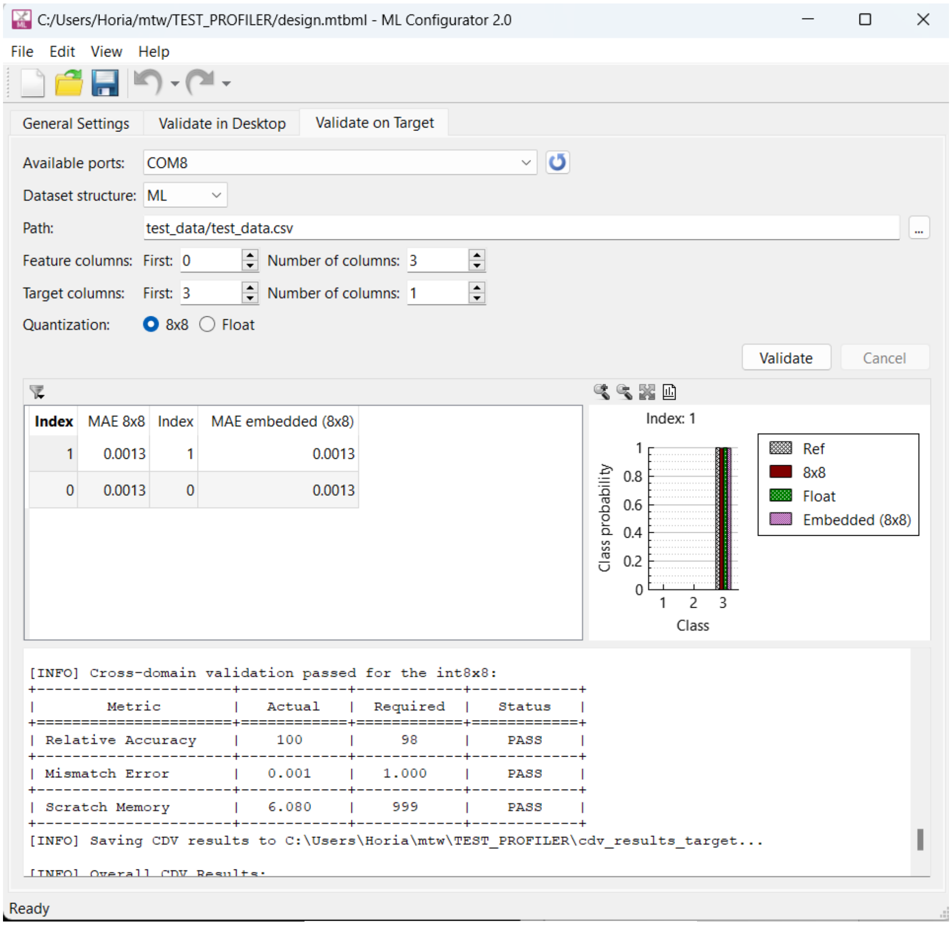


Fig. 22. Model validation on PSoC6.

After programming the IoT device, via the serial connection to the computer, it displays on the UART terminal the real-time prediction together with the percentage of confidence for each possible class (Fig. 23).

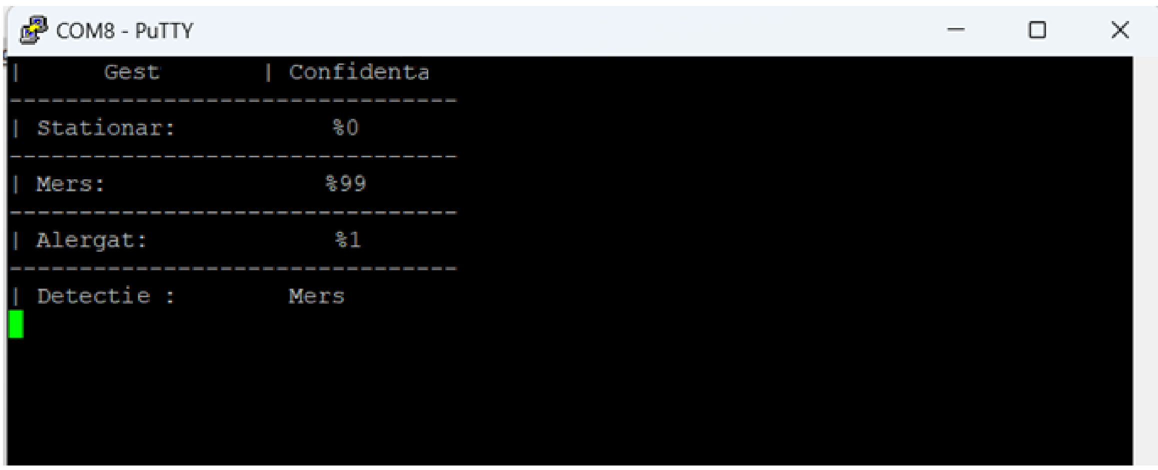


Fig. 23. Model prediction.

2.4. Chapter summary. Dissemination

This chapter described the application of Artificial Intelligence algorithms in biomedical and physiological signal processing, focused on the detection of cardiovascular diseases, fatigue using a smart watch and human activities, through *edge* processing and prediction techniques.

The first experiment is based on the development of a blood pressure data acquisition and processing system using a machine learning model. Blood pressure data was acquired using a very affordable IoT device - the PSoC6 microcontroller. The model uses a multilayer perceptron (MLP) neural network to accurately predict the 5-year risk of developing BCV in LabVIEW using the Python Node module. Compared to similar studies for predicting BCV, the developed machine learning model obtained higher accuracy performance indicators.

The second study aims to develop a non-invasive proactive model for real-time fatigue estimation. Data from several people have been collected and processed and Artificial Intelligence and machine learning algorithms are to be applied on this collected data. For this, the relationship between tremor, heart rate and SpO2 on the one hand and fatigue onset on the other hand will be used. The expected results are to define a relevant warning framework for the presence of early indications of fatigue.

The last experiment classifies human activities based on motion sensor data (accelerometer and gyroscope). The model programmed on the IoT device was pre-trained on the computer using the Keras library and classifies several common activities: stationary, walking and running. In an infinite loop, the IoT device reads data from a motion sensor (BMX160) attached to the PSoC6 to detect activities. The main advantage of this experiment is that both data acquisition, data processing and prediction are performed directly at the *edge*, on the PsoC 6 chip. This application can run directly on the smart system described in Chapter 2.

The practical experiments presented in this chapter have been presented at international conferences and published or forthcoming in ISI-indexed SpringerLink conference volumes:

- **H. A. Modran**, D. Ursuțiu, C. Samoilă, T. Chamunorwa: "Artificial Intelligence System for predicting cardiovascular diseases using IoT devices and Virtual Instrumentation", In Online Engineering and Society 4.0 - Proceedings of the 18thth International Conference on Remote Engineering and Virtual Instrumentation (REV) 2021, http://dx.doi.org/10.1007/978-3-030-82529-4_28.

  Web of Science Number: WOS:000772185600028

- **H. A. Modran**, T. Chamunorwa, D. Ursuțiu, C. Samoilă: "Fatigue Estimation using Wearable Devices and Virtual Instrumentation", In Open Science in Engineering - Proceedings of the 20th International Conference on Remote Engineering and Virtual Instrumentation (REV) 2023 (accepted and submitted, in publication).

3. APPLICATION OF ARTIFICIAL INTELLIGENCE IN AUDIO SIGNAL PROCESSING

3.1. Recognition of the state transmitted by music using AI

In the current experiment, the dominant emotion conveyed by a given musical sequence was estimated using an Artificial Intelligence model [89]. The basic emotion wheel [91] describes the types of emotions into which songs are categorized. A categorical approach was used, music was divided into groups and each group was described with an adjective (sad, happy, boring, etc.).

The pipeline of the experiment involves the following steps (Fig. 24):

- extract audio features
- exploratory data analysis
- data set cleaning
- initial model training
- model evaluation
- design and development of a machine learning classifier

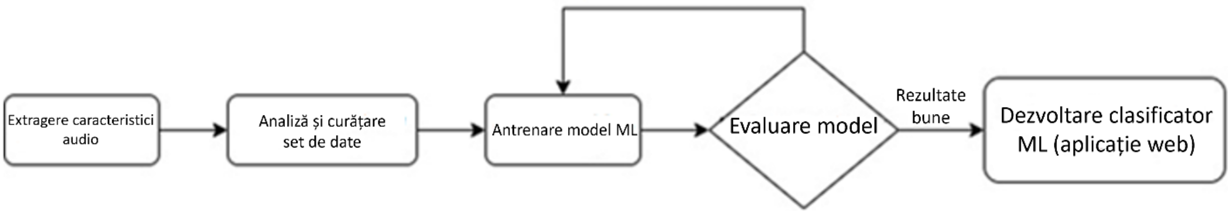


Fig. 24. Structure of the experiment.

Because everything in life has a vibration, music therapy focuses on specific frequencies. Six of these frequencies, known as *solfeggio* frequencies, are specific tones known since ancient times to have a beneficial effect on the mind and body.

A typical audio processing process involves data acquisition and extraction of acoustic features relevant to the problem, followed by decision making schemes involving knowledge detection, and classification techniques.

The musical features used in this experiment are as follows:

1. spectral characteristics:
   - spectral centroid, i.e. the magnitude-weighted average signal frequency
   - spectral roll-off - how many frequencies are concentrated below a certain threshold
   - spectral flux - how much the frequency varies over time
   - mel frequency cepstral coefficients (MFCCs)
2. temporal characteristics
   - zero-crossing rate - the number of zero crossings in the time domain in a frame
   - time centroid - the timestamp of a signal that represents a temporal balance point of the energy of the sound event
   - log attack time - the time required to reach the maximum amplitude of a signal from a minimum time threshold

3. melodic/harmonic characteristics:
    - pitch class profile - enhanced pitch distribution feature consisting of sequences of feature vectors describing pitch
    - key clarity
    - harmonic change - the rate at which chords change in a melody
    - music mode
4. rhythmic characteristics:
    - beat histogram
    - average tempo (measured in beats per minute)

The exploratory data analysis step was performed after extracting audio features from the dataset. The main purpose of this step was to determine which features could be used as relevant indicators to make an accurate prediction. The distribution of each variable was plotted and then analysed. The 10 most relevant audio features were selected after analysis of each feature and the others were removed from the dataset.

Since classification problems use labeled data, these labels were used for each audio file. Songs were divided into 4 categories, represented by different colours, namely "Energetic", "Calm", "Happy" and "Sad". These categories were selected based on the article written by M. Nuzzolo [97], who explains which is the best way to classify music by mood.

In the process of normalizing the features, a *MinMaxScaler* was used to scale the values between a range of [0,1] and preserving the shape of the original distribution. Finally, the dataset was split, with 80% allocated for training the network and the remaining 20% for testing.

The Keras library was used to build the model, as this library is designed to allow fast design of deep neural networks. Several possible values were tested and compared for selecting input and output layers and activation functions.

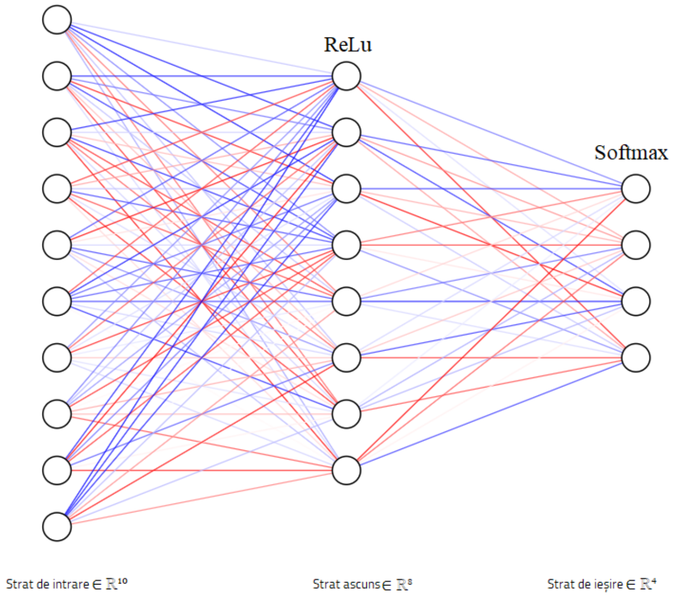


Fig. 25. Neural network structure.

The layers of the developed neural network are as follows (Fig. 25):

- An input layer with 10 audio features as input

- A fully connected multi-node hidden layer with a Rectified Linear Unit (*ReLU*) activation function
- An output layer containing four outputs (one for each category) with a Softmax activation function. Therefore, a classifier with the role of estimator was also needed

The estimator was evaluated using K-Fold cross-validation. After trying with different possible values, the number of splits was set to K = 10. The overall accuracy of the model was 91.49%. In the current experiment, the model was trained on 8000 samples. Finally, to assess the accuracy of the model, the confusion matrix was plotted (Fig. 26). The model accuracy score was also calculated.

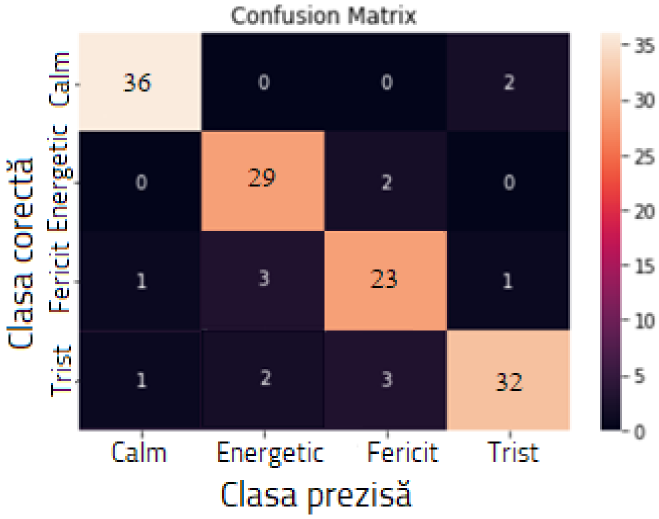


Fig. 26. Confusion matrix.

With a final accuracy score of 94% and a confusion matrix analysis, it was observed that the model classified calm, happy and energetic songs very well, but the accuracy for sad songs was slightly lower at 85%. It also tried to improve the accuracy of the model by changing some parameters such as batch size, number of epochs and aggregating or deleting some features used for training the model.

The classification is based on a dataset whose features have already been extracted by the algorithm presented in Subchapter 4.1.2, using a web application developed in HTML with JavaScript. In order to perform the classification, it is necessary to provide the files containing the extracted features, generated by the Python application, in either CSV or JSON format. The data is also validated and, if necessary, will be normalized, in order for the algorithm to perform optimally

The user interface used to configure the parameters is illustrated in Fig. 27. It allows setting the learning rate, the percentage for the validation dataset, the number of epochs and the number of neurons in the hidden layer. Pressing the Classification button will classify the songs and display the results in the console. When running the classifier, performance parameters for each step are also displayed in the browser console.

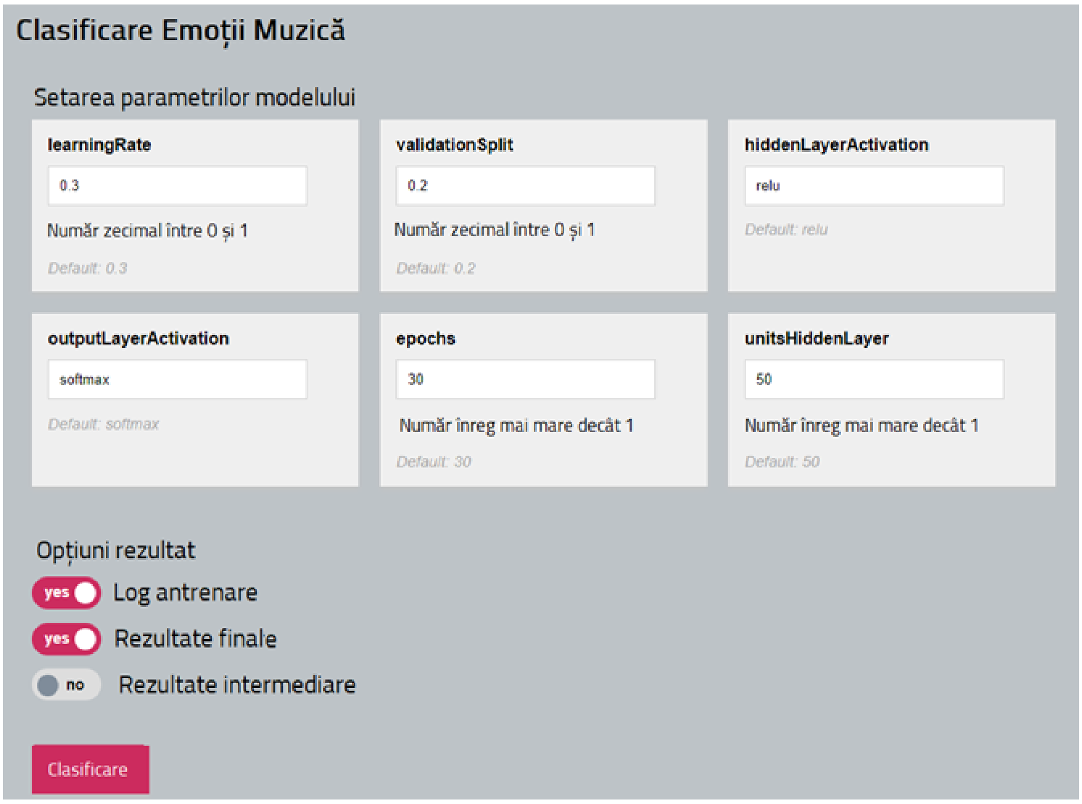


Fig. 27. Web Application UI.

Another web application has been developed, allowing the music therapist to select some characteristics for the patient and a song, and based on the machine learning model described above, the application will be able to indicate whether that specific song will have a therapeutic effect for the patient. Figure 28 shows the user interface of the application.

Fig. 28. Classification Web Application.

A study to validate the Machine Learning solution is ongoing and the system developed has already been tested on a large number of people. The participants were either CVCT members or students of the Master of Music Therapy programme at Transilvania University of Brasov, all with different musical dispositions and tastes. These subjects used the application on different days and in different moods, and the Machine Learning model predicted and chose correctly about 91.6% of the time.

3.2. Analysis of audio signals using DeepLearning

In this experiment, an artificial neural network (ANN) was built to classify audio files according to musical genre. Spectrograms were displayed using the Librosa library in Python.

Each audio signal consists of several features. However, only the features relevant to the problem should be used. The spectral features (frequency-based features), which are obtained by converting the time-based signal into the frequency domain using the Fourier Transform, are the fundamental frequency, frequency components, spectral centroid, spectral flux, spectral density, spectral drift.

The dataset used consists of 1000 audio tracks of 30 seconds each. It contains 10 genres, each genre being represented by 100 tracks. The tracks are all 16-bit 22,050 Hz monophonic audio files in *.wav* format. The genres selected are Blues, Classical, Country, Disco, Hiphop, Jazz, Metal, Pop, Raggae and Rock. First, the audio files must be converted to PNG images (spectrograms). From these spectrograms, significant features must be extracted, i.e. MFCCs, Spectral centroid, Zero crossing rate, Chroma frequencies, Spectral roll-off. The first step is to convert the audio files into spectrograms (Fig. 29).

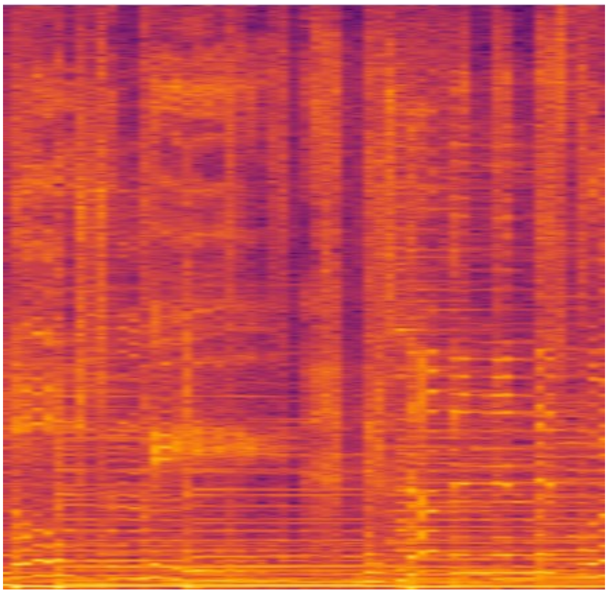


Fig. 29. Example of a spectrogram of the Country music genre.

From these spectrograms the following characteristics of the audio signals were extracted:

- Mel frequency cepstral coefficients (MFCC)
- Centroid spectral
- Zero pass rate
- Chroma frequencies
- Spectral roll-off

The next step is data preprocessing, which involves loading CSV data, encoding labels, scaling features and splitting the initial dataset into training dataset and test dataset.

A fully connected neural network with 4 layers and *adam* optimizer was used for the model:

- dense layer with 256 neurons and ReLU activation function
- 128-unit dense layer and ReLU activation function
- dense layer with 64 neurons and ReLU activation function
- 10-unit dense layer and Softmax activation function

After 100 epochs, the accuracy achieved by the algorithm was over 99.8%, thus achieving a much higher accuracy than that shown in other similar existing studies. As illustrated in Fig. 30, in the last epochs the accuracy did not improve anymore, but only slightly the loss function.

```
Epoch 94/100
7/7 [==============================] - 0s 4ms/step - loss: 0.0256 - accuracy: 0.9987
Epoch 95/100
7/7 [==============================] - 0s 5ms/step - loss: 0.0209 - accuracy: 0.9987
Epoch 96/100
7/7 [==============================] - 0s 5ms/step - loss: 0.0234 - accuracy: 0.9987
Epoch 97/100
7/7 [==============================] - 0s 4ms/step - loss: 0.0221 - accuracy: 0.9987
Epoch 98/100
7/7 [==============================] - 0s 5ms/step - loss: 0.0175 - accuracy: 0.9987
Epoch 99/100
7/7 [==============================] - 0s 5ms/step - loss: 0.0187 - accuracy: 0.9987
Epoch 100/100
7/7 [==============================] - 0s 5ms/step - loss: 0.0212 - accuracy: 0.9987
```

Fig. 30. Neural network accuracy improvement.

After training and validating the model in the Python ecosystem, obtaining very good performance indicators (almost 100% prediction accuracy), it was exported as a Keras model for use in ModusToolbox and validated on the PSoC 6 microcontroller. Since the model was optimised in terms of memory consumption to run on PSoC 6, the model was re-validated on the computer and subsequently on PSoC 6, using both int 8x8 and float weight quantization. The results of this validation are shown in Table 2.

Table 2. Model validation results on PSoC 6

Performance indicator	Obtained	Needed	Status
Relative accuracy	100	98	PASS
Mismatch error	0.010	1.000	PASS
Scratch memory	5.280	999	PASS

3.3. Detecting musical notes, speech and noise

The current study proposes real-time automatic detection of musical notes, speech and background noise using a deep learning model based on a fully connected neural network. In this experiment, the SensiML plugin was used, which helps to collect data from PSoC6 through attached sensors and also provides methods to label the data that was captured. When the dataset is collected and annotated, a model can be generated and trained that is optimized for PsoC 6. The experiment consists of the following steps:

- audio data acquisition and annotation
- application of signal pre-processing techniques on collected data
- design and training of a classification algorithm
- Implementing a smart model optimised for IoT device resources

Data was acquired using the microphone on the CY8KIT-028-TFT shield. It contains a digital microphone with a single-bit Pulse-Width Modulation (PDM) output, allowing any sound captured by the microphone to be converted to a digital signal. The PSoC6 converts this digital signal into a quantized 16-bit Pulse-Core Modulation (PCM) value. An interrupt is triggered when there is sufficient data to be processed, i.e. at least 128 samples.

Numerous segments were saved for each of the following musical notes: D, E, F, G, A, B. After data acquisition was completed, these were tagged in the Data Capture Lab. In addition to the musical notes, several files containing audio data on speech and ambient noise were acquired and tagged. Also, after acquisition, the data is automatically uploaded to the Cloud. Through the Cloud portal the data can be viewed and all tags and their distribution can be analysed. As illustrated in Fig. 31, for each musical note there are a similar number of segments, while for speech and noise more segments have been made, as they have a larger variety and more complex features.

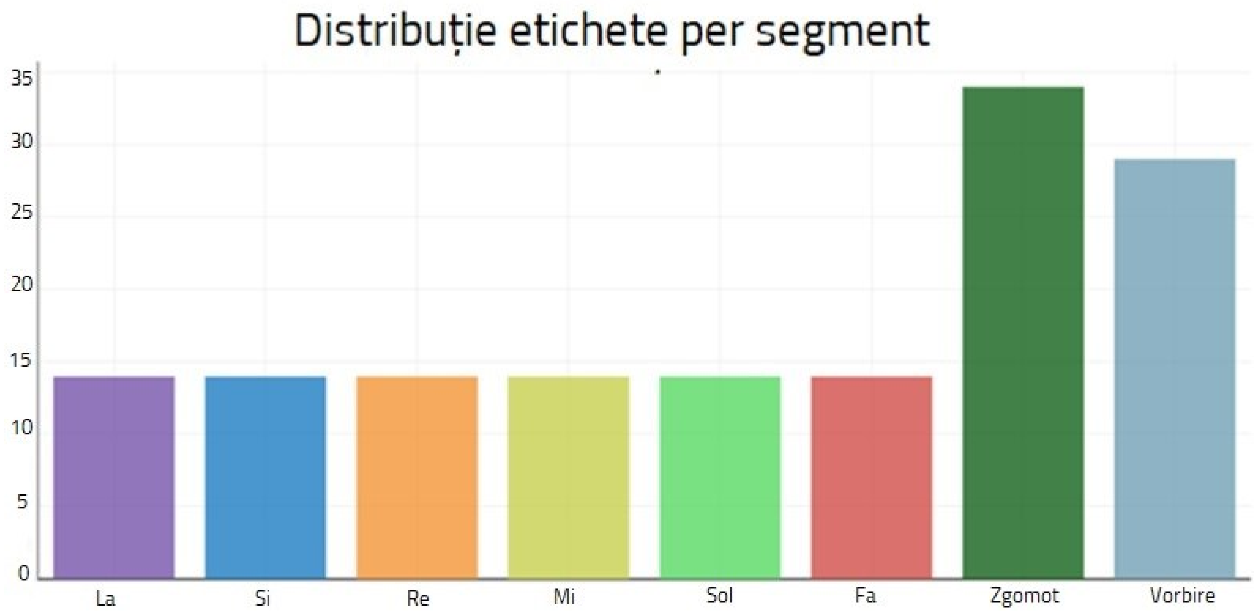


Fig. 31. Number of segments for each label.

The next step is to add a filter and set the whole Pipeline. In this step the following items have been configured (Fig. 32):

- 400 size segmenting *windowing* - takes input from the sensor's transform/filter step and stores data until a segment is found
- frequency domain feature generator - a collection of feature generators processes the data segment to extract meaningful information
- data balancing: Undersample Majority Classes - creates a balanced dataset by undersampling majority classes using random sampling without replacement
- quantization of features: *min-max scaler* - normalizes and scales the data to integer values between min_bound and max_bound, leaving the specified pass columns unscaled
- outlier filter: zscore filter - filters feature vectors that have values outside a threshold limit (threshold has been set to 3)
- classifier: TensorFlow Lite for Microcontrollers - takes a feature vector as input and returns a classification based on a predefined model
- Training algorithm: a fully connected neural network with the following features:
    - Dense layers with size (number of neurons) 128, 64, 32, 16, 8
    - learning rate of 0.01
    - a number of 4 epochs (*epochs*)
    - activation for Softmax end layer
    - threshold of 0.8 (if the classified value is lower, the prediction will be *Unknown*)
    - categorical cross entropy loss function
- validation parameters - accuracy, F1 score, sensitivity

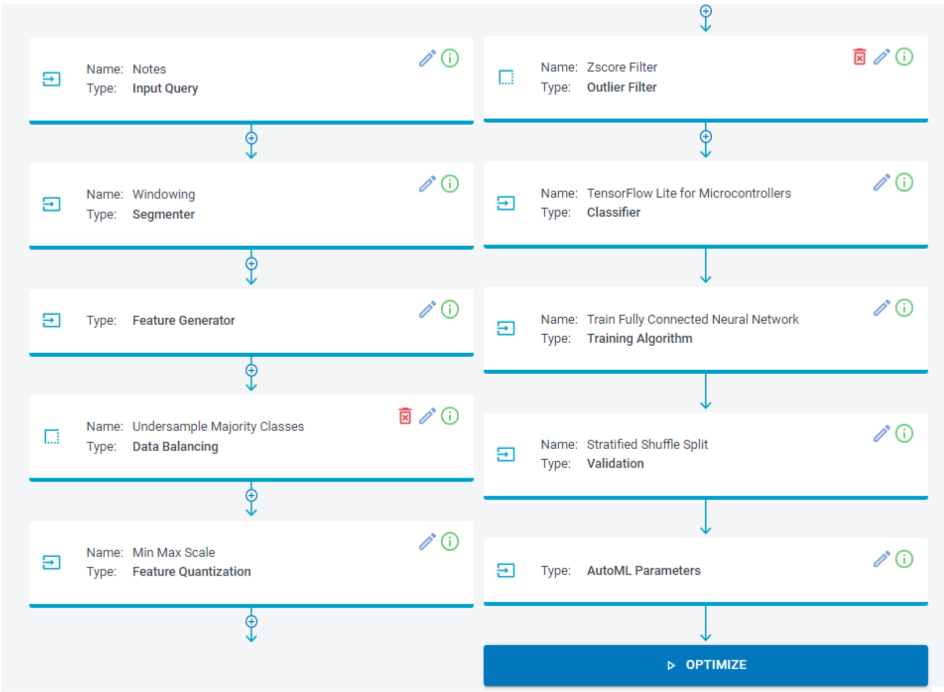


Fig. 32. Pipeline features for the ML algorithm.

After optimization a model is generated whose characteristics are displayed graphically (Fig. 33). As shown in Table 3, the model obtained very good performance indicators, both in terms of accuracy, sensitivity and F1 score, and as a classifier dimension.

AutoML Results

RESTART

MODEL NAME	ACCURACY	CLASSIFER SIZE(B)	NUM. FEATURES	SENSITIVITY	F1-SCORE
Notes_Fold_0	98	20372	23	98	98

Fig. 33. Characteristics of the created model.

Table 3. Model indicators.

Indicator	Value
Accuracy	97.69%
Classifier size	20372 bytes
Number of features	23
Sensitivity	97.6%
F1 Score	98%

In order to better understand and analyze the cases in which the classifier predicted correctly or erroneously, the confusion matrix on the training set, as well as one on the validation set was determined (Fig. 34). From these it is clearly that the model performed very well and similarly on both the training dataset (data that the model had previously seen, having been provided in the training stage) and the validation dataset, which the network had not seen before. The only mispredictions observed were for speech and only in one specific case - when the sound intensity is very low and there is background noise.

Validation

	A	B	D	E	E1	G	Noise	Talking	UNK	Support	Sense %
A	198	0	0	0	0	0	0	0	0	198.00	100.00
B	0	191	0	0	0	0	0	0	3	194.00	98.45
D	0	0	190	0	0	0	2	0	5	197.00	96.45
E	0	0	0	197	0	0	0	0	1	198.00	99.49
E1	0	0	0	0	186	0	0	0	0	186.00	100.00
G	0	3	4	1	1	180	1	0	4	194.00	92.78
Noise	0	0	0	0	0	0	182	0	3	185.00	98.38
Talking	0	0	1	0	0	0	5	180	3	189.00	95.24
Total	198.00	194.00	195.00	198.00	187.00	180.00	190.00	180.00	19	1541.00	
Pos_Pred(%)	100.00	98.45	97.44	99.49	99.47	100.00	95.79	100.00		Acc(%)	97.60

Fig. 34. Confusion matrix for the neural network.

Several variants for the number of epochs were tested, and after graphical analysis of them it was concluded that the best number both in terms of training and prediction time as well as accuracy is 4.

Since the model works very well on test and validation data, it can be run on real-time data to validate its functionality on the PSoC 6 device. The application programmed on the PSoC6 transmits the prediction made in real time via the UART and can be viewed via either Putty/Tera Term or Open Gateway. Via Open Gateway, serial connection can be made to the COM port that has been allocated to the IoT device. A JSON file was also generated when creating the model to map the class number with the name given to them via tags. After the connection is made, the current real-time prediction is displayed in Test *Model* mode, along with the history of previously made predictions (Fig. 35).

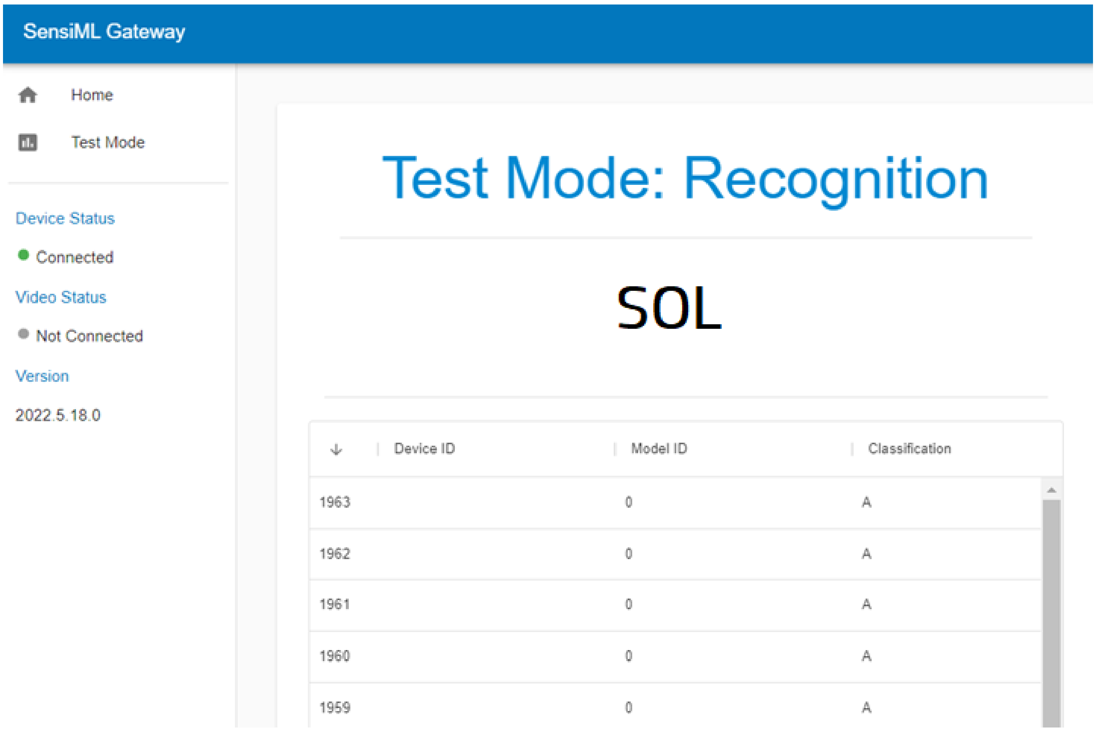


Fig. 35 Testing the model on the IoT device.

In addition, the prediction is also displayed on the CY8KIT-028-TFT screen connected to the PSoC6 (Fig. 36).

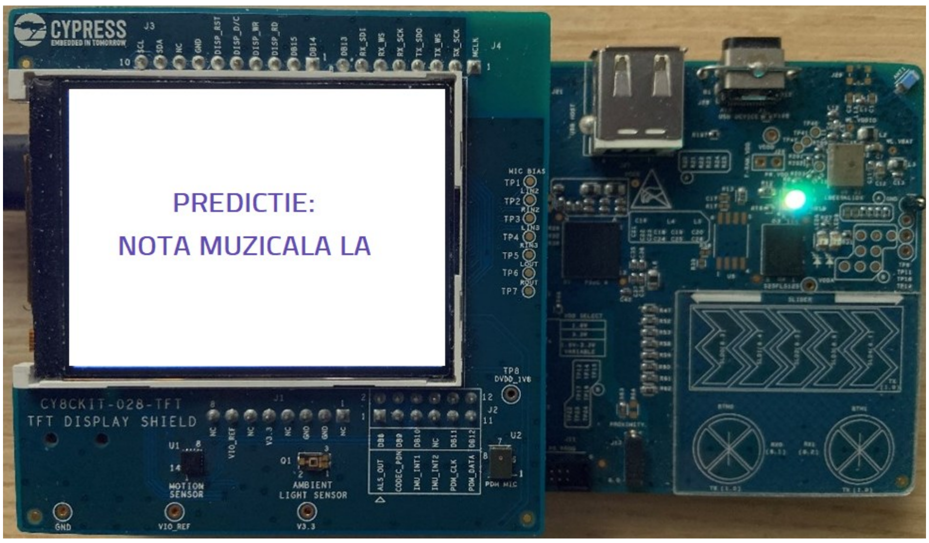


Fig. 36. Display of the prediction on the TFT screen.

3.4. Chapter summary. Dissemination

This chapter illustrates the application of deep learning models in audio signal processing, in particular for the recognition of states conveyed by music, the detection of musical genre, as well as musical notes, speech and noise, using *edge* processing techniques.

Since previous studies have not addressed in detail the relationship between music features and their therapeutic effects through audio signal processing, the first study described in this chapter proposed a machine learning solution for recognizing the therapeutic effect conveyed by music. The developed algorithm is a classifier using a multiclass neural network. It comprises an input layer with ten features, a fully connected hidden layer with several hidden units and an output layer. A web application has also been developed to customize the hyperparameters of the machine learning model, as well as another application to predict whether a song is suitable for a particular person.

In the second experiment, an artificial neural network was built to classify audio files according to musical genre. The audio signals were transformed into spectrograms, from which significant features were extracted . The main frequency spectral features, which are obtained by converting the signal from time domain to frequency domain using Fourier Transform, are fundamental frequency, frequency components, spectral centroid, spectral flux, spectral density, spectral drift. After 128 epochs, the accuracy achieved by the algorithm was more than 99%, thus achieving a higher accuracy than those presented in similar studies.

The latest practical experiment proposes real-time automatic detection of musical notes, speech and background noise using a deep learning model based on a fully connected neural network. In this experiment, the SensiML plugin was used, which helps to collect data from PSoC 6 through attached sensors and also provides methods to label the data that was acquired. After the dataset was collected and properly annotated, a deep learning model that is optimized for PSoC 6 was developed and trained. The main innovation of this solution is that the data processing takes place in real time, at the *edge*, on the PSoC6 chip, and can thus be run on the smart board described in Chapter 2.

The study described in this chapter has been published in the MDPI Sensors journal:

- **H. A. Modran**, T. Chamunorwa, D. Ursuțiu, C. Samoilă, H. Hedesiu: "Using Deep Learning to Recognize Therapeutic Effects of Music Based on Emotions", Sensors. 2023; 23(2):986, ISSN 1424-8220, DOI: https://doi.org/10.3390/s23020986
  IF: 3.847, CiteScore: 6.8
  Web of Science Number: WOS:000927263000001
  Journal Rank: JCR - Q2 / CiteScore - Q1

4. REDUCING SIGNAL NOISE USING ARTIFICIAL INTELLIGENCE

4.1 Noise reduction in audio signals using Deep Learning

Noise reduction in audio signals is a very popular issue. The aim is to filter noise from the input signal while avoiding degradation of signal quality. Thus, the main task of the noise reduction algorithm is to suppress the background noise in order to achieve an improvement of the audio signal [101].

An Autoencoder is a neural network that uses unsupervised learning, which means it does not need a *target* function. There is only one training set, which is also the target set. A convolutional autoencoder uses convolutional neural networks. The idea is simple, involving a neural network that has the same number of output nodes as input nodes. In addition, symmetry is usually present in the hidden layers of the network. An autoencoder transforms the input signal into a lower dimensional representation using a component called an encoder (Fig. 37). In this way, it is similar to an audio encoder, which compresses an audio signal into a representation with fewer bits than the original audio signal. The decoder part of one of these maps the lower dimensional representation back into the higher dimensional representation. The reconstruction after the decoder part should be as close as possible to the original. Therefore, the original (the training set) is also the target. The input must go through two general steps:

1. Encoder - in this step, the input is compressed, and a reduced representation of it is constructed. It is usually referred to as a *code*, and the number of units in this layer is the dimensionality of the reduction.
2. Decodor - here, the input data is reconstructed from code.

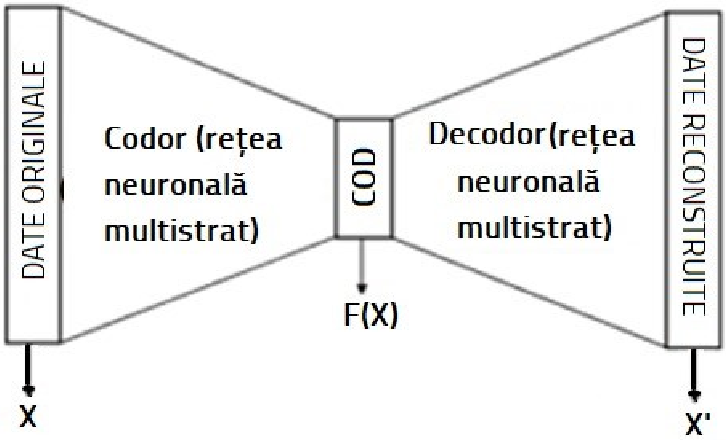


Fig. 37. General architecture of an autoencoder.

In this experiment, a dataset of several thousand *.wav* files was used. It contains the clean versions of the audio recordings as well as the corresponding versions of the noisy audio recordings.

Since the files are stereo, the left channel was used for training the neural network and the right channel for model testing. In the training loop noise is added with a factor of 0.05.

The program was developed in Python and a 1D convolutional layer was used for the encoder. In addition, a *tanh* activation function was used, which proved to work better than the sigmoid function for the specifics of this application. A s*tride* (sampling factor) was also used, for which various values were tried and tested, as well as filtering of size 2N. This *stride* factor was reduced to 64, resulting in superior audio quality. Thus, a larger amount of data is processed, but the disadvantage of this process is that the training and optimisation of the network takes a longer time. A different noise is added to each epoch of the training process, so that the model can generalize to any noise it encounters.

Therefore, in the application developed in Python, the convolutional encoder layer has the following properties:

- 1 input channel (*in_ channels*)
- 32 output channels (out_channels)
- kernel size of 2048
- stride of 32
- padding of 1023

Thus, a single input channel is used, representing the audio signal, and 32 output channels (representing the sub-bands). A transposed convolutional layer was used on the decoder side. For this layer 32 input channels are required, representing the sub-bands, and a single output channel (equivalent to the reconstructed audio signal).

In the training phase, the input signal was also used as a target. Its length was limited to the length of the signal produced by the model output, which can be obtained by simply letting the model run once before training it. Y is the target signal, having the same length as the model output.

As a loss function, the best choice is the Mean *Squared Error* (MSE) and its radical (Root-Mean-Squared-Error - RMSE).

Training the network is then done in a for loop, using 4000 epochs. To improve the accuracy of the model, an Adam optimizer was also used and the initial learning rate was set to 10^{-4} . During the training process, the model started with a very high dropout function in the first iterations and decreased over time:

```
0 0.019962582737207413
10 0.01638539507985115
.....
3980 0.0008305259980261326
3990 0.0008273686398752034
```

After the training process has been completed, the weights generated at this stage can be read and used to graphically display the encoder analysis filter and decoder filters for subband 0 (Fig. 38).

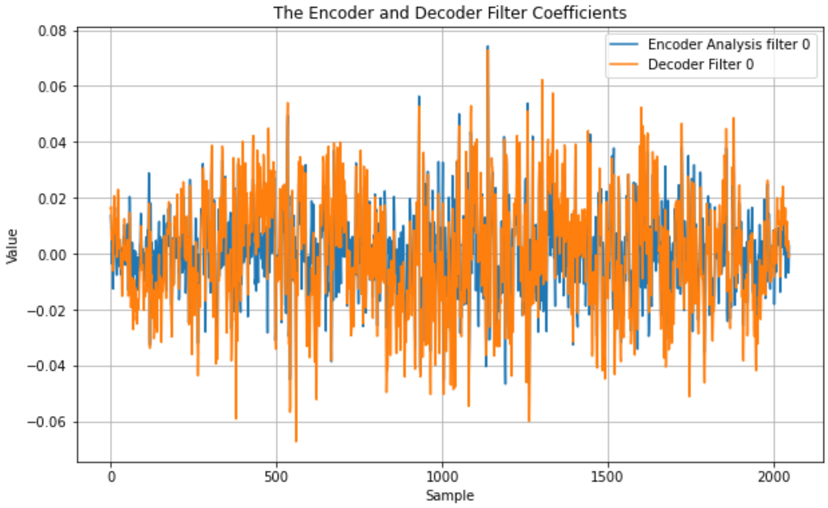


Fig. 38. Encoder and decoder filter coefficients .

The resulting model was tested on the training set, being able to make predictions based on previously determined weights. The auditory version of the predicted audio signal was also tested, and the results were found to be of high accuracy.

The model was also tested on the validation set, which the model had never seen before (Fig. 39). To validate the results, the signal-to-noise ratio was calculated, resulting in a value of approximately 67.1 dB.

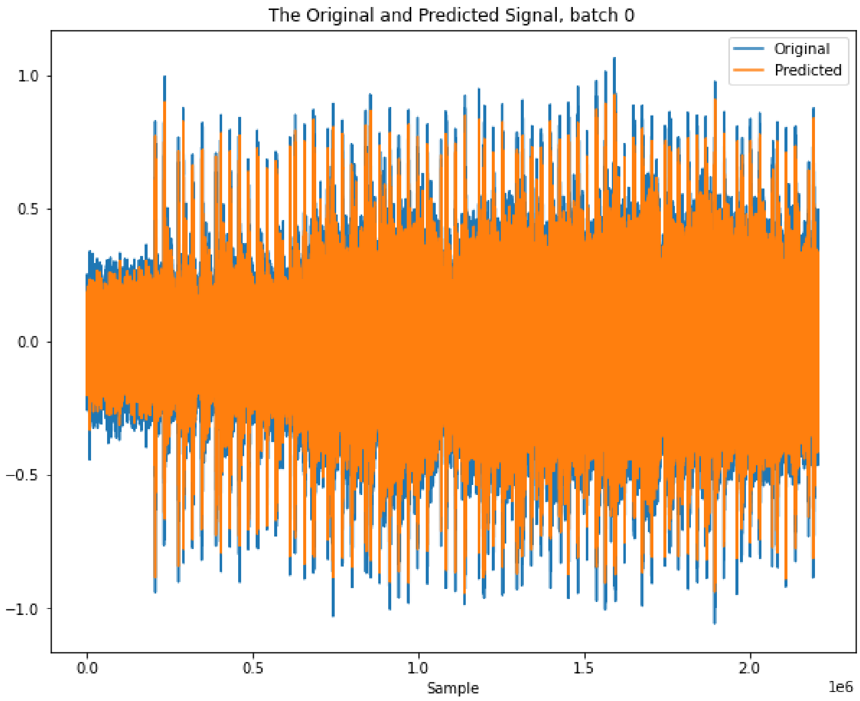


Fig. 39. Model predicted values on the validation set .

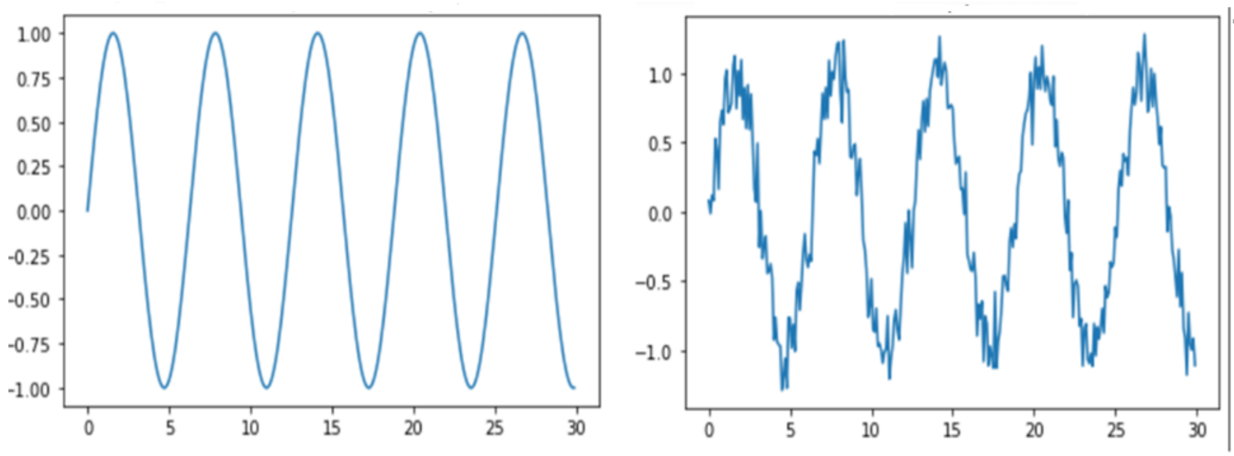
4.2. Signal noise reduction autoencoder

Feeding autoencoders noisy data as inputs and clean data as outputs makes it possible for them to recognize noise in the training data. In this way, autoencoders can serve as denoisers.

In this experiment an autoencoder was implemented to demonstrate this through a three-step process:

- making a large dataset of useful signal samples
- creating a large data set of samples to which Gaussian (random) noise was added
- developing an autoencoder that learns to convert noisy inputs into the original signal, which reduces noise, and is able to generalize to new data outside the data set being used

As useful signals (and therefore targets for automatic coding), pure samples from a small domain. Fig. 40 (a) shows the shape of the signal. 100,000 samples were used for the model. To each of them, Gaussian noise was added. While the overall shape remains present, it is clear that the signal becomes noisy - Fig. 40 (b).



(a) useful signal (b) signal with additive noise

Fig. 40. Representation of generated signals.

The architecture of the deep learning model developed is illustrated in Fig. 40 and comprises eight layers:

- the input layer, which takes the input data
- three Conv1D convolutional layers with 128, 64 and 32 filters respectively, serving as encoder
- three Conv1D transposition layers with 32, 64 and 128 filters, serving as decoders
- a Conv1D layer with a single output, a Sigmoid activation function and padding, which serves as an output layer

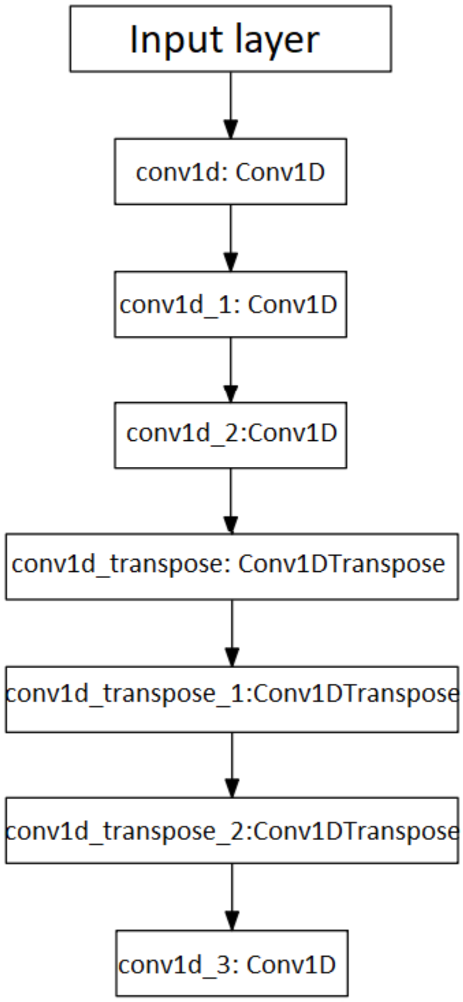


Fig. 41. Structure of the Deep Learning model.

After generating the data, the autoencoder is created, which involves the following operations:

- setting configuration parameters for the model
- data pre-processing
- definition of the model architecture
- compile the model and start training
- view the waveforms in the test set for which noise has been removed to visually validate that the algorithm is working properly

The neural network model was configured with parameters defined in Table 4.

Table 4. Autoencoder neural network parameters.

Parameter	Value
Input_shape	(300, 1)
Batch	125
Ephochs	10
Train_test_split	0.3
Validation_Split	0.2
Max_norm_value	2.0

The distribution of model layers is shown in Fig. 42, resulting in a total of 65,729 parameters in all 7 layers of the neural network.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 298, 128)          512

 conv1d_1 (Conv1D)           (None, 296, 64)           24640

 conv1d_2 (Conv1D)           (None, 294, 32)           6176

 conv1d_transpose (Conv1DTra  (None, 296, 32)          3104
 nspose)

 conv1d_transpose_1 (Conv1DT  (None, 298, 64)          6208
 ranspose)

 conv1d_transpose_2 (Conv1DT  (None, 300, 128)         24704
 ranspose)

 conv1d_3 (Conv1D)           (None, 300, 1)            385

=================================================================
Total params: 65,729
Trainable params: 65,729
Non-trainable params: 0
_________________________________________________________________
```

Fig. 42. Distribution of neural network layers.

After the training process is complete, it is necessary to test whether the model is working properly. To achieve this, several reconstructions were generated: a noisy sample from the test set (which are data that the model has never seen before) and tested whether it generates the signal without noise. Fig. 43 illustrates the result of the test performed on a sample.

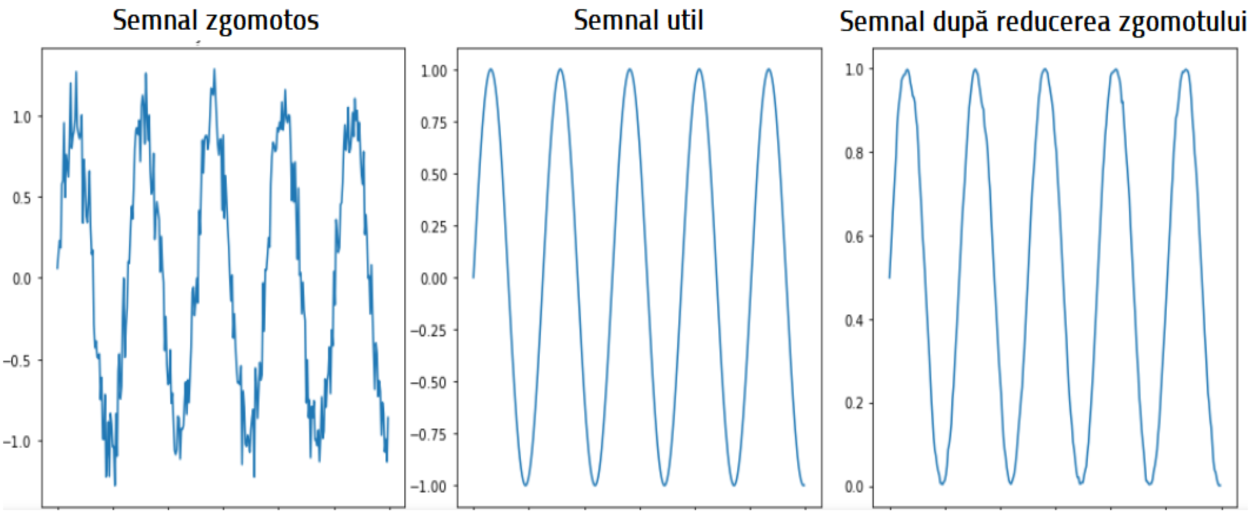


Fig. 43. Model Testing.

As can be seen in Fig. 43, the encoder has learned to remove much of the noise. In order to validate the results concretely, the value of the signal-to-noise ratio was determined and the result obtained was 24.12 dB.

4.3. Chapter summary. Dissemination

The current chapter presents the experiments carried out to suppress noise in audio signals and to develop a model for reducing noise in signals.

Noise reduction in audio signals is a very popular issue. The aim is to filter noise from the input signal while avoiding degradation of signal quality. The first practical experiment proposes a "Denoising Autoencoder" as a possible solution to this problem, using a dataset of several thousand stereo audio files in *.wav* format. The smart model developed was tested on the training set and then on the validation set, and the results proved to be very good. The auditory version of the predicted audio signal was also tested and found to be highly accurate.

Providing autoencoders with noisy data as inputs and clean data as outputs makes it possible for them to recognize ideosyncratic noise for training data. In this way, autoencoders can serve as denoisers. Thus, the second experiment is to develop an autoencoder to prove this theory. The developed deep learning model uses three convolutional layers serving as encoder and three transcoding layers as decoder. To test the model developed, several reconstructions were generated: a noisy sample from the test set (data that the model has never seen before) and tested whether it preserves the original signal without noise. The results show that this model has learned to remove much of the noise, and is thus very good.

The experiment described in this chapter was presented at the International Conference on Remote Engineering and Virtual Instrumentation (REV) 2022, held in Cairo (Egypt) from 28 February to 2 March 2022, and published in the conference volume - SpringerLink, which is also undergoing ISI indexing:

- **H. A. Modran**, D. Ursutiu, C. Samoila, T. Chamunorwa: "Noise Reduction Through Artificial Intelligence Techniques: An Introductory Study", In Artificial Intelligence and Online Engineering - Proceedings of the 19th International Conference on Remote Engineering and Virtual Instrumentation (REV) 2022, http://dx.doi.org/10.1007/978-3-031-17091-1_3

5. FINAL CONCLUSIONS. ORIGINAL CONTRIBUTIONS. DISSEMINATION OF RESULTS AND FUTURE RESEARCH DIRECTIONS

5.1. Original contributions

This section presents the original contributions of this PhD thesis, which are directly connected with the main scientific objectives and the specific objectives detailed in the Introduction:

1. Development of an intelligent data acquisition and processing system based on PSoC 6

An innovative intelligent data acquisition and processing system has been developed based on the PSoC6 microcontroller, which is software reconfigurable. Its final version is battery powered and contains a dual-core CYBLE-416045-02 chip, USB type C connection, BLE communication, as well as microBNC and Pmod connectors. The developed board is very small in size, has several advantages and innovations that allow developers to realize a wide range of applications for acquiring real-world signals, which can then be processed by Artificial Intelligence techniques at the edge. Several PSoC 6 applications have been developed for this board, but they do not present an exhaustive list of all the capabilities of the board or the applications it can run. Given the reconfigurability of the system, the number of AI applications that can be programmed on the microcontroller is not finite, depending only on the capability and inventiveness of the developers.

2. Biomedical and physiological signal processing

a. Predicting cardiovascular disease using Artificial Intelligence techniques

The approach used in this study is based on the development of a blood pressure data collection and processing system using machine learning algorithms. The acquired data is obtained using a powerful IoT device based on the PSoC 6 microcontroller integrated circuit. The novel contribution of this experiment is that the data is processed inside a virtual tool created in the LabVIEW environment, which uses a machine learning model developed in Python. Thus, the developed model consists of a multilayer perceptron neural network to predict whether a person is at high risk of developing cardiovascular disease. Compared to similar studies, the machine learning model developed achieved higher performance and accuracy.

b. Fatigue detection via a smart watch using machine learning

This study aims to develop a non-invasive proactive model for real-time fatigue estimation, which has not been done in previous studies. This will use the relationship between tremor, heart rate and SpO2 on the one hand and fatigue onset on the other hand and will take into account age, gender, comorbidities that may affect tremor and exercise performed to create a personalized prediction of fatigue level for the user. Biomedical data from several people has been acquired and processed and Artificial Intelligence and machine learning algorithms are to be applied to this collected data. The expected results are to define a relevant warning framework for the presence of early indications of fatigue.

c. Gesture classification using *edge* processing techniques

The original contribution of this experiment is to bring physiological signal processing by Artificial Intelligence techniques directly onto the very small PSoC6 chip, without the need for expensive hardware or reliable internet connections, thus being able to run directly on the intelligent system presented in Chapter 2 of the thesis. The developed application performs classification of human activities based on data coming from the motion sensor (accelerometer and gyroscope). The model programmed on PSoC6 was pre-trained on the computer using the Keras library and classifies several common activities: stationary, walking and running, with a very high accuracy (over 99%).

3. Audio Signal Processing - connected with the subjects taught in the Master of Music Therapy

a. Recognizing emotions conveyed by music using Deep Learning techniques

As the relationship between music features and their therapeutic effects through audio signal processing has not been addressed in detail so far, the contribution of this experiment was to propose a machine learning solution for the recognition of the therapeutic effect conveyed by music. The developed algorithm is a classifier using a multiclass neural network. It comprises an input layer with ten features, a fully connected hidden layer with several hidden units and an output layer. A web application has also been developed to customize the hyperparameters of the machine learning model, as well as another application to predict whether a song is suitable for a particular person. The dominant emotion conveyed by a given musical sequence was estimated using an Artificial Intelligence model. The basic emotion wheel describes the types of emotions into which songs are classified. A categorical approach was used, music was divided into groups and each group was described with an adjective - e.g. sad, happy, boring, etc.

b. Analysis of audio signals using deep learning (music genre classification)

In this study, an artificial neural network was developed to classify audio songs according to music genre. Audio signals were transformed into spectrograms, from which significant features were extracted. Compared to the solutions presented by other similar studies, only the main frequency spectral features were used, which are obtained by converting the signal from the time domain to the frequency domain using the Fourier Transform - i.e. fundamental frequency, frequency components, spectral centroid, spectral flux, spectral density, spectral drift. After 128 epochs, the accuracy achieved by the algorithm was more than 99%, thus achieving higher accuracy than illustrated in other similar existing studies.

c. Detection of musical notes, speech and noise by edge processing

The study proposes real-time automatic detection of musical notes, speech and background noise using a deep learning model based on a fully connected neural network. The SensiML plugin was used in this experiment, which helps to collect data from PSoC6 through attached sensors, and also provides methods to label the data that was captured. After the dataset was collected and properly annotated, a deep learning model was developed and trained that is optimized for PSoC 6. The main innovation of this solution is that the data processing takes place in real time on the PSoC 6 chip, and can thus be run on the smart board described in Chapter 2.

4. Reducing signal noise

a. Reducing audio noise from signals using Deep Learning

Noise reduction in audio signals is a very topical issue. The aim is to filter noise from the input signal while avoiding degradation of signal quality. This experiment proposes an innovative "Denoising Autoencoder" as a possible solution to this problem, using a dataset of several thousand stereo audio files. The smart model developed was tested on the training set and then on the validation set, and the results proved to be very good. The auditory version of the predicted audio signal was also tested and found to be highly accurate.

b. Signal noise reduction autoencoder

Providing autoencoders with noisy data as inputs and clean data as outputs makes it possible for them to recognize noise in the training data. In this way, autoencoders can serve as denoisers. The original contribution made by this experiment is to develop an autoencoder to practically prove this theory. The deep learning model developed uses three convolutional layers serving as encoder and three transcoding layers as decoder. Testing of the model was carried out on a noisy sample from the test set, which the model has never seen before, and was verified to preserve the characteristics of the original signal without noise. The results show that this model has learned to remove a large part of the noise, and, therefore, proved to be useful.

5.2. Dissemination of results and international awards

The following articles and manuscripts have been published in the field of the PhD thesis:

A. Papers published in ISI WoS indexed journals:

- **H. A. Modran**, T. Chamunorwa, D. Ursuțiu, C. Samoilă, H. Hedesiu: "Using Deep Learning to Recognize Therapeutic Effects of Music Based on Emotions", Sensors. 2023; 23(2):986, ISSN 1424-8220, https://doi.org/10.3390/s23020986.
  Web of Science Number: WOS:000927263000001
  IF: 3.847, CiteScore: 6.8 (Journal Rank: JCR - Q2 / CiteScore - Q1)
- T. Chamunorwa, **H. A. Modran**, D. Ursuțiu, C. Samoilă, H. Hedeșiu: "Reconfigurable Wireless Sensor Node Remote Laboratory Platform with Cloud Connectivity", Sensors 21(19):6405, ISSN 1424-8220, http://dx.doi.org/10.3390/s21196405.
  Web of Science Number: WOS:000927263000001
  IF: 3.847, CiteScore: 6.8 (Journal Rank: JCR - Q2 / CiteScore - Q1)
- T. Chamunorwa, **H. A. Modran**, D. Ursuțiu, C. Samoilă, H. Hedeșiu: "Cloud-Based, Expandable-Reconfigurable Remote Laboratory for Electronic Engineering Experiments", Electronics 11(20):3292, ISSN 2079-9292, http://dx.doi.org/10.3390/electronics11203292.
  Web of Science Number: WOS:000927263000001
  IF: 2.69, CiteScore: 4.7

B. Papers published in SpringerLink conference volumes indexed by ISI WoS (or still to be indexed):

- **H. A. Modran**, D. Ursuțiu, C. Samoilă, T. Chamunorwa: "Learning Methods Based On Artificial Intelligence in Educating Engineers for the New Jobs of the 5th Industrial Revolution", In Educating Engineers for Future Industrial Revolutions - Proceedings of the 23rd International Conference on Interactive Collaborative Learning (ICL) 2020, Volume 1, http://dx.doi.org/10.1007/978-3-030-68201-9_55.
  Web of Science Number: WOS:000772405700055
- **H. A. Modran**, D. Ursuțiu, C. Samoilă, T. Chamunorwa: "Artificial Intelligence System for predicting cardiovascular diseases using IoT devices and Virtual Instrumentation", In Online Engineering and Society 4.0 - Proceedings of the 18th International Conference on Remote Engineering and Virtual Instrumentation (REV) 2021, http://dx.doi.org/10.1007/978-3-030-82529-4_28. Web of Science Number: WOS:000772185600028
- **H. A. Modran**, D. Ursuțiu, C. Samoilă, T. Chamunorwa: "Intelligent IoT Biomedical Bluetooth Data Acquisition System", In New Realities, Mobile Systems, and Applications - Proceedings of the International Conference on Interactive Mobile Communication, Technologies, and Learning (IMCL) 2021, http://dx.doi.org/10.1007/978-3-030-96296-8_88.
- **H. A. Modran**, D. Ursuțiu, C. Samoilă, T. Chamunorwa: "Noise Reduction Through Artificial Intelligence Techniques: An Introductory Study", In Artificial Intelligence and Online Engineering - Proceedings of the 19th International Conference on Remote Engineering and Virtual Instrumentation (REV) 2022, http://dx.doi.org/10.1007/978-3-031-17091-1_3.
- **H. A. Modran**, T. Chamunorwa, D. Ursuțiu, C. Samoilă: "Fatigue Estimation using Wearable Devices and Virtual Instrumentation", In Open Science in Engineering - Proceedings of the 20th International Conference on Remote Engineering and Virtual Instrumentation (REV) 2023 (accepted and submitted, in publication).
- **H. A. Modran**, T. Chamunorwa, D. Ursuțiu, C. Samoilă: "Integrating Artificial Intelligence and ChatGPT into Higher Engineering Education", accepted for presentation and publication at the 26th International Conference on Interactive Collaborative Learning (ICL) 2023, to be held in Madrid (Spain), 26 - 29 September 2023.

C. Books, Laboratory guidance

- **H. A. Modran**, D. Ursuțiu: "Instrumentație Virtuală: Îndrumar de laboratorio", Transilvania University of Brasov, ISBN 9786061915460 (2022)

D. International Awards Received

- International Conference Remote Engineering and Virtual Instrumentation (REV) 2021 - **Best Paper Award 2021: Student Paper Award Winner**
- International Conference Remote Engineering and Virtual Instrumentation (REV) 2021 - **Best Paper Award 2021: Special Session Award Winner**

- International Conference New Trends on Sensing-Monitoring-Telediagnosis for Life Sciences (NT SMT-LS) 2022 - **Oral Presentation Award**

Also, together with the research group of CVTC Center from Transilvania University, in connection with activities related to the PhD thesis, the following articles were published (8 articles in ISI indexed conference volumes):

- T. Chamunorwa, D. Ursuțiu, C. Samoilă, **H. A. Modran**: "Embedded System Learning Platform for Developing Economies", In Educating Engineers for Future Industrial Revolutions - Proceedings of the 23rd International Conference on Interactive Collaborative Learning (ICL) 2020, Volume 1, http://dx.doi.org/10.1007/978-3-030-68201-9_60.
- T. Chamunorwa**,** D. Ursuțiu, C. Samoilă, H. Hedeșiu, **H. A. Modran**: "Electronic Educational Laboratory Platform for Students", In Online Engineering and Society 4.0 - Proceedings of the 18th International Conference on Remote Engineering and Virtual Instrumentation (REV) 2021, http://dx.doi.org/10.1007/978-3-030-82529-4_30.
- C. Samoila, D. Ursuțiu, **H. A. Modran**: "The Potential For Transformation Into The Virtual Organization Of Remote Experiment Networks", In Mobility for Smart Cities and Regional Development - Challenges for Higher Education - Proceedings of the 24th International Conference on Interactive Collaborative Learning (ICL) 2021, http://dx.doi.org/10.1007/978-3-030-93907-6_88.
- T. Chamunorwa**,** D. Ursuțiu, C. Samoilă, **H. A. Modran**: "Electronic Laboratory Educational Board", In New Realities, Mobile Systems, and Applications - Proceedings of the International Conference on Interactive Mobile Communication, Technologies, and Learning (IMCL) 2021, http://dx.doi.org/10.1007/978-3-030-96296-8_89.
- T. Chamunorwa, D. Ursuțiu, C. Samoiăa, **H. A. Modran**: "Software Configurable Hardware-based Remote Laboratory System", In Artificial Intelligence and Online Engineering - Proceedings of the 19th International Conference on Remote Engineering and Virtual Instrumentation (REV) 2022, http://dx.doi.org/10.1007/978-3-031-17091-1_2.
- D. Ursuțiu, **H. A. Modran**, T. Chamunorwa, C. Samoilă, P. Kane: "Digital Tools and Energy Harvesting in IoT Education", In Learning in the Age of Digital and Green Transition - Proceedings of the 25thth International Conference on Interactive Collaborative Learning (ICL) 2022, http://dx.doi.org/10.1007/978-3-031-26190-9_99.
- T. Chamunorwa, **H. A. Modran**, D. Ursuțiu, C. Samoilă: "Embedded Student Board for Digitalization of Engineering Education", In Learning in the Age of Digital and Green Transition - Proceedings of the 25thth International Conference on Interactive Collaborative Learning (ICL) 2022, http://dx.doi.org/10.1007/978-3-031-26190-9_100.
- C. Samoila, D. Ursuțiu, **H. A. Modran**, T. Chamunorwa: "New Challenges for Remote Experiment Design in the Digitalization Era (Industry 4.0)", In Learning in the Age of Digital and Green Transition - Proceedings of the 25thth International Conference on Interactive Collaborative Learning (ICL) 2022, http://dx.doi.org/10.1007/978-3-031-26190-9_98.

BIBLIOGRAPHY (extract)

[74] H. A. Modran, D. Ursutiu, C. Samoila, T. Chamunorwa: "*Intelligent IoT Biomedical Bluetooth Data Acquisition System*", In New Realities, Mobile Systems, and Applications - Proceedings of the International Conference on Interactive Mobile Communication, Technologies, and Learning (IMCL) 2021, http://dx.doi.org/10.1007/978-3-030-96296-8_88 (2022)

[75] H. A. Modran, D. Ursuțiu, C. Samoilă, T. Chamunorwa: "*Artificial Intelligence System for predicting cardiovascular diseases using IoT devices and Virtual Instrumentation*", In Online Engineering and Society 4.0 - Proceedings of the 18th International Conference on Remote Engineering and Virtual Instrumentation (REV) 2021, http://dx.doi.org/10.1007/978-3-030-82529-4_28.

[76] D. Sharma, M. Ferguson, T. Kamp, F. Zhao: "*Constructing biomimetic cardiac tissues: a review of scaffold materials for engineering cardiac patches*", Emergent Materials 181-191, DOI: https://doi.org/10.1007/s42247-019-00046-4 (2019).

[77] C. Krittanawong, H. Virk, S. Bangalore et al: "*Machine learning prediction in cardiovascular diseases: a meta-analysis*", Scientific Reports 10, 16057, DOI: https://doi.org/10.1038/s41598-020-72685-1 (2020).

[79] K. Tae, Y. Roh, Y. Hun Oh, H. Kim, S. Whang: *"Data Cleaning for Accurate, Fair, and Robust Models: A Big Data - AI Integration Approach"*, arXiv:1904.10761v1 (2019).

[80] P. Langley: *"The changing science of machine learning,"* Machine Learn Journal, vol. 82, Springer Link, DOI: https://doi.org/10.1007/s10994-011-5242-y (2011).

[83] Z. Omary, F. Mtenzi: "*Machine Learning Approach to Identifying the Dataset Threshold for the Performance Estimators in Supervised Learning"*, International Journal for Infonomics (IJI), Volume 3, Issue 3, DOI: http://dx.doi.org/10.20533/iji.1742.4712.2010.0034 (2010).

[84] J. Wan, Z. Qin, P. Wang, Y. Sun, X. Liu: "*Muscle fatigue: general understanding and treatment*", Experimental & molecular medicine, Vol. 49 (10), e384-e384, DOI: https://doi.org/10.1038/emm.2017.194 (2017).

[88] H. A. Modran, T. Chamunorwa, D. Ursuțiu, C. Samoilă: "*Fatigue Estimation using Wearable Devices and Virtual Instrumentation*", In Open Science in Engineering - Proceedings of the 20th International Conference on Remote Engineering and Virtual Instrumentation (REV) (2023)

[89] H. A. Modran, T. Chamunorwa, D. Ursuțiu, C. Samoilă, H. Hedesiu: "*Using Deep Learning to Recognize Therapeutic Effects of Music Based on Emotions*", Sensors. 2023; 23(2):986, ISSN 1424-8220, https://doi.org/10.3390/s23020986 (2023).

[91] Emotion Wheel, [Online] at https://positivepsychology.com/emotion-wheel/, accessed 01 December 2022.

THESIS ABSTRACT (EN/RO)

Abstract

The doctoral thesis presents a complex approach for developing a prediction system based on hybrid Artificial Intelligence algorithms, with application both in biomedical and audio signal processing, as well as in signal noise reduction. Chapter 1 presents the current state of the art in this field. In Chapter 2, "Development of an intelligent data acquisition and processing system" the development of a reconfigurable intelligent data acquisition and processing system using Artificial Intelligence methods, based on PSoC 6, is presented. Artificial Intelligence models presented in the following chapters, being mostly developed using the Tensorflow library, can be exported, programmed and run on this system. Chapter 3, "Applications of Artificial Intelligence in biomedical and physiological signal processing" describes the application of Artificial Intelligence algorithms in biomedical and physiological signal processing, focused on the detection of cardiovascular diseases, fatigue using a smart watch and human activities, through techniques of processing and prediction at the edge. Chapter 4, entitled "Application of Artificial Intelligence in Audio Signal Processing" illustrates the application of deep learning models in audio signal processing, in particular for the detection of music mood, music genre classification, as well as musical notes, speech and noise, using edge processing techniques. In Chapter 5, "Reduction of noise from signals using Artificial Intelligence, the experiments carried out for the reduction of noise from audio signals and the development of a model for filtering noise from signals are presented. The last chapter of the thesis presents the general conclusions and the original contributions of the author.

Rezumatul Tezei

Teza de doctorat prezintă abordare complexă privind realizarea unui sistem de predicție bazat pe algoritmi hibrizi de Inteligență Artificială, cu aplicare atât în procesarea semnalelor biomedicale și audio, cât și în reducerea zgomotului din semnalele. Capitolul 1 prezintă stadiul actual. În Capitolul 2, “Dezvoltarea unui sistem inteligent de achiziție și procesare a datelor” este prezentată dezvoltarea unui sistem inteligent reconfigurabil de achiziție și prelucrare a datelor prin metode de Inteligență Artificială, bazat pe PSoC 6. Modele de Inteligență Artificială prezentate în capitolele următoare, fiind în mare parte dezvoltate folosind librăria Tensorflow, pot fi exportate, programate și rulate pe acest sistem. Capitolul 3, “Aplicații ale Inteligenței Artificiale în procesarea semnale biomedicale și fiziologice” descrie aplicarea algoritmilor de Inteligență Artificiale în prelucrarea semnalelor biomedicale si fiziologice, axată pe detectarea bolilor cardiovasculare, a oboselii folosind un ceas inteligent și a activităților, prin tehnici de procesare și predicție la margine. Capitolul 4, intitulat “Aplicarea Inteligenței Artificiale în procesarea semnalele audio” ilustrează aplicarea modelelor de învățare profundă în procesarea semnalelor audio, în special pentru detecția stărilor transmise de muzica, recunoașterea genului muzical, precum și a notelor muzicale, a vorbirii și a zgomotului, utilizând tehnici de procesare de tip edge. În Capitolul 5, “Reducerea zgomotului din semnale folosind Inteligența Artificială sunt prezentate experimentele realizate pentru reducerea zgomotului din semnalele audio și dezvoltarea unui model de filtrare a zgomotului din semnale. Ultimul capitol al tezei prezintă concluziile generale ale lucrării și contribuțiile originale ale autorului.