Non-Intrusive Appliance Load Monitoring in an Intelligent Device at the Edge layer

José Alonso Aguirre-Núñez*, Juan Pablo Serrano-Rubio*
Information Technologies Laboratory
Tecnológico Nacional de México/ITS de Irapuato
Irapuato, Guanajuato, Mexico
ja.aguirrenunez@gmail.com, juserrano@itesi.edu.mx

Rafael Herrera-Guzman †
Mathematics Department
Center for Research in Mathematics (CIMAT)
Guanajuato, Guanajuato, Mexico
rherrera@cimat.mx

***Abstract*—Nowadays, companies and researchers are developing devices that are connected over the Internet to create new services for users through the collection and analysis of data obtained from sensors. The information obtained from the sensors is collected in the cloud. However, there is a new approach called edge computing whose goal is to process information at the edge of a network instead of processing it in the cloud. Edge computing, combined with machine learning algorithms, has become a powerful tool to optimise tasks in both industrial processes and everyday life. This combination allows decision making in real-time since the data processing is carried out in the place where it is being acquired. In this paper we propose a Non-Intrusive Appliance Load Monitoring (NIALM) which has two functions: a) send detailed energy consumption information to the data server only when it is necessary, and to process the information using an intelligent algorithm based on an Artificial Neural Network to recognise when and how much energy the appliances are consuming. The PCB design of the board includes the ESP12-S microchip. We evaluated the Evolutionary Hyperplane Neural Network against the Evolutionary Spherical Neural Network to decide the best algorithm for our proposed method. The Evolutionary Artificial Neural Networks are trained using the Differential Evolution Algorithm. According to the numerical experiments, the Evolutionary Hyperplane Neural Network showed a better performance of classification up to 82%.**

***Index Terms*—IoT, Edge computing, Differential Evolution Algorithm, Artificial Neural Network**

I. INTRODUCTION

The popularity of the edge computing is due to the combination of Internet of Things (IoT) devices and the machine learning algorithms in real-time applications [1], [2]. This real-time applications in combination with IoT, provides low latency, mobility, and the reduction of the traffic of the networks. For that reason, edge computing has the potential to reduce response times, increase battery life, and reduce bandwidth usage [3]. The edge computing devices are located in the Edge Layer and their goal is to connect devices locally as well as to manage the collection and connection of the devices to the data server. The advantages of the Edge Layer is to select the data which is important to store in the data server and reduce the load of data being transmitted [4].

Thanks to CONACyT, FORDECyT and Tecnológico Nacional de México/ITS de Irapuato for their support.

In this paper, we propose a Non-Intrusive Appliance Load Monitoring which is implemented using a device board installed in the edge layer, and we incorporate an intelligent algorithm based on Evolutionary Artificial Neural Network (EANN) to filter the information of the appliances energy consumption. The information obtained by the EANN stored in the data server. The implementation of the EANN topology is carried out as an embedded system.

Our approach aims to help people know accurately their energy consumption to enable them to reduce it or optimise it and, thus, tackle worldwide problems such as climate change. Nowadays, climate change is the main environmental challenge, and energy consumption is one of the most important causes of this phenomenon [5], [6]. Recently, there has been a significant increase in energy consumption [7] due to people relying mostly on electric devices and appliances in their daily lives. This increase in energy consumption is worrying societies and governments worldwide since the production of such energy releases big amounts of CO_2 into the atmosphere. For instance, in the US alone, 1763 million metric tons (MMt) of CO_2 was released into the atmosphere in 2018 by the electric power production sector [8], [9]. Due to the environmental changes and problems we have witnessed during the last decades, it is wise to look for different ways of saving energy.

There is a technology intended to reduce, or help reduce and manage, the energy consumption at home called Home Energy Management System (HEMS), which displays information about the energy consumption in home appliances. Accordingly, the HEMS model may allow you to manage the home appliances connected to it. The display and energy consumption control have shown to be effective ways to reduce energy consumption at home [10], [11]. However, despite the information presented by these HEMS, which helps the users to reduce or minimise their energy consumption, it is only for a limited period of time since the gadgets become obsolete due to the users becoming unaware of them out of habit [12], [13].

For this reason, it is important to look for other ways to measure energy consumption and present the information obtained in a more user-friendly way. In this case, a Non-Intrusive Appliance Load Monitoring (NIALM) or Non-

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

Intrusive Load Monitoring (NILM), have some advantages against other energy measurement devices, which are of simple installation and low cost for the user [14], [15]. These devices are intended to gather information from several appliances connected at the same time and identify the changes of energy consumption due to different appliance events to find what device is connected to the NILM or NIALM device.

One of the paradigms of NILMs and NIALMs is to identify which appliance is connected and what consumption is having separately when more than one appliance is connected to the same power outlet. The output signal read from the NILM or NIALM is the aggregation of all the energy consumption done by the appliances. This energy aggregation makes the identification of appliances connected to the power outlet very difficult. To identify the appliances connected to the power outlet, researchers have come up with disaggregation algorithms which allow us to separate the aggregated energy signals into the individual appliance energy signals. This allows the researcher to identify which appliance is connected and what consumption is having separately. We propose a Non-Intrusive Appliance Load Monitoring (NIALM) which identifies the appliances connected to the power outlet without disaggregating the signal obtained by our device, allowing us to identify which device is connected and at the same time measure the energy consumption by the appliances connected to it.

The paper has five more sections as follows. Section II describes the board architecture and the schematic diagrams of our board. Section III shows the neural network topology and the training of the algorithm. Section IV shows the results and experiments done with our neural network. Finally, Section V concludes the paper with a brief description of the achievements.

II. Hardware: IoT Board Design and Energy Consumption Sensor

As Hart [15] mentioned, the idea of a NILM, or nowadays NIALM, is to be accessible, low cost, and ease of use for the users. For that reason, we designed and implemented a NIALM device by using open source technology, such as Arduino boards, and other open-source software. Additionally, the sensors used are of really low cost making the device accessible for everyone. These components can also be found all around the world, so anyone can implement the same device without problems.

The board used as a base for the project is based in Arduino, and it is called NodeMCU, this board is mainly used for the Internet of Things (IoT) [16]. The board gives us all the features the chip `ESP8266EX` has, which is WiFi connection at 2.4 GHz, with a protocol 802.11b/g/n. It has 4 MB of flash memory, 80 MHz of the system clock, 80 KB of RAM. This module can be programmed in two different languages, the Arduino language or Python, and it has an operating voltage of 3.3 V [17]. The Arduino ecosystem was designed as an easier way to prototype different electronic and programming projects, the problem is that there is not enough information in order to take such prototyping into a mass production environment.

The endeavour of creating this NIALM took us into the need of changing our Arduino prototype and move it into a mass production environment so we can have a more visual appeal NIALM device with the same features as the board we were using (NodeMCU). We came up with a board designed in KiCAD with the same features we wanted for our board and all the external elements such as capacitor or resistors embedded in Figure 1.

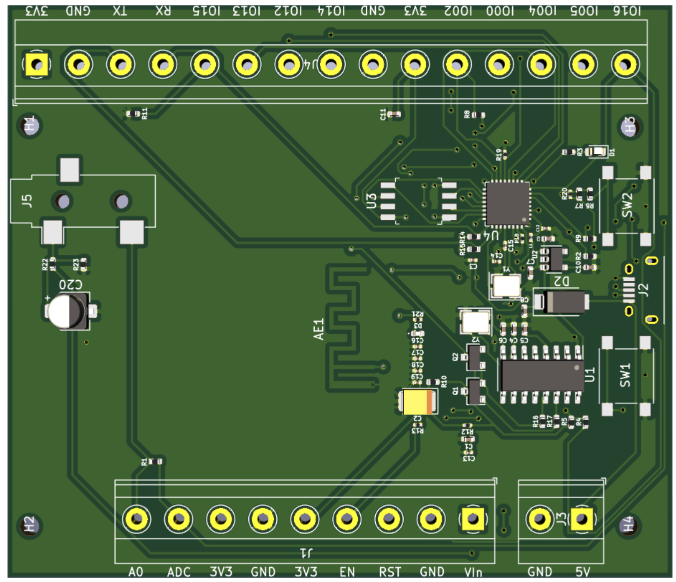


Fig. 1. NIALM device (3D PCB design)

The KiCAD software allows us to create all the schematic diagrams shown in Figs. 2, 3 & 4, and then edit its PCB layout and visualise it in 3D as presented on Figure 1. The schematics we present in this paper shows how we had to design all the board parts so we can create or design our own board. The schematic diagrams presented here can be followed to design your own board with all the features of a NodeMCU and the connection needed for our Current Transformer sensor.

The schematic diagram in Figure 2 shows the chip ESP8266EX and all its connections, this chip is the one that allows us to connect over the internet and use it as an IoT or Edge device. In order to work properly, this diagram also displays the flash memory that our device will use in order to load our programs. The following Figure 3 shows the different parts of our device, it is similar to the NodeMCU. In this diagram, we can observe the power input, the USB to Serial converter, power regulator, connectors, etc.

The last diagram in Figure 4 presents the connection we had to do for our sensor; the sensor is connected to a signal offset to ensure positive input voltages to the device.

The sensor used to measure the power consumption in our device is the SCT-013 30A, it is a Current Transformer (CT) sensor capable of measuring Alternating Current (AC), they are mainly used in the measuring of whole-building energy

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

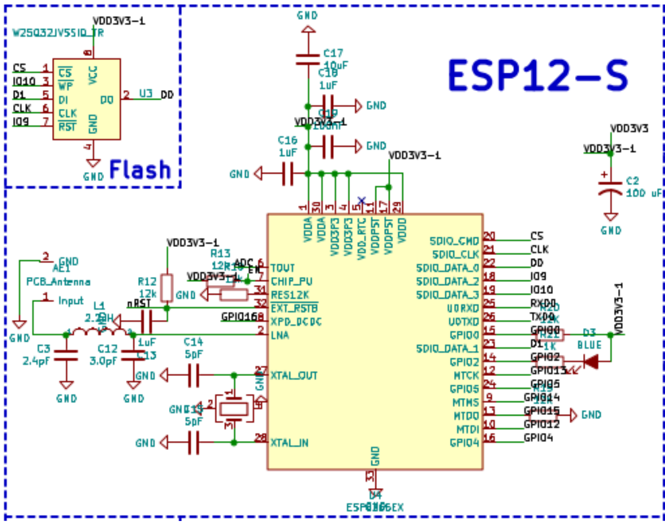


Fig. 2. Schematic diagram of ESP8266EX used as a slave, similar as the ESP-12S board. This allows us WiFi connection.

consumption. CT sensors use to type of cores, the split core, or the solid core, we decided to use a split core sensor to facilitate the connectivity. We have to have in mind that CT sensors, work on single wires, you cannot connect dual-core cables or multi-core cables.

A. Energy Consumption Sensor

The idea of the Internet of Things is to have everything connected over the internet. The purpose of this interconnection is to facilitate human activities, and also help us to understand the world around us, the same way we do with our sensory apparatus, the body [18]. The problem with this interconnection is the enormous amounts of data generated by these sensors and then sent towards the cloud to be processed there. The speed in which the data is gathered is starting to outclass the speed of data transportation, thus becoming a bottleneck for the cloud server [3], and creating other problems such as long response times.

On Figure 5 we can observe the paradigm of cloud computing the data generator (sensor), sends the data towards the cloud, this data is processed and stored in the server, then the users have access to this data by sending a request to the cloud. The data generator (sensor) only purpose in IoT is only to send information. Therefore, researchers are starting to propose that the data must be processed somewhere away from the cloud server, for that reason it would be more efficient to process the data at the edge of the network where it is being generated [3]. To reduce the amount of data sent towards the cloud, we decided to process that data generated and perform almost all tasks that the cloud sever normally does in IoT [2]. The data obtained by our NIALM device will be processed in this same device Figure 1. Despite this device does not have enough computational power, it provides the features of major concern which are fast response time, low battery consumption, and reduced bandwidth [19]. In Figure 6 we present our edge paradigm with our NIALM device. We can observe that our NIALM device is connected over the internet but the kind of connection is different from Figure 5, this connection also can receive requests from the final users, so the users can interact with the NIALM device itself without the need of the cloud server.

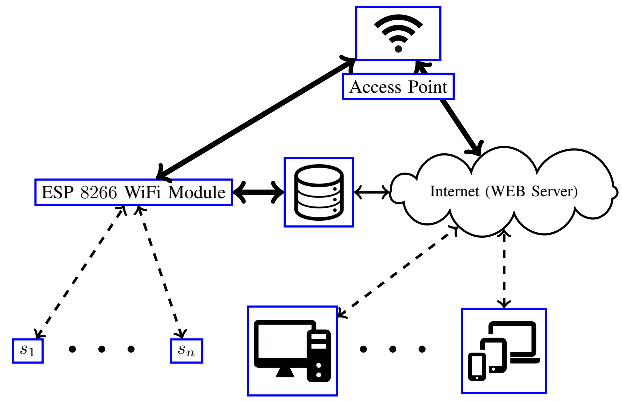


Fig. 6. Diagram of the data transmission, among sensors, users, database and web server.

In Figure 7, we have a better view of our device's diagram and its interaction with all the appliances and users connected to it. If we pay attention to the diagram we can observe that now our NIALM device (ESP 8266) has an Artificial Algorithm embedded in it (Sec. III) to process the data, and also take decisions with the actuators connected to it. This AI algorithm will help us with the energy disaggregation problem and will let the user know what appliance or appliances are connected to it and the energy consumption the appliances have.

III. Software: IoT Edge System based on Evolutionary Spherical Neural Network

The purpose of a NILM or NIALM is to classify the load of several appliances connected at the same time, the problem is when two or more appliances are connected at the same time the load consumption signals are *aggregated* (added). For instance, we can observe on Figure 12(a); The main purpose of a NILM or NIALM is to disaggregate (separate) the energy consumption signals on Figures 11(a) & 11(c) of the devices to let the user know which appliances are connected to the power source. There are different methods to disaggregate the signal such as use some signals features like voltage waveform or signal harmonics [20]. Another way to facilitate load disaggregation is to classify the type of load in four different categories [21], which establishes the type of load we can observe in the appliance behaviour.

The energy signals samples presented in Figures 11 and 12 show four energy signals obtained from our NIALM device. Figures 11(a) & 11(c) present the load signal of two appliances a blender and a fridge working in a short period of time. Energy consumption signal in Figure 12(a) show the appliances working at the same time.

As mentioned before we did not program an energy disaggregation algorithm in our device, we processed the signal

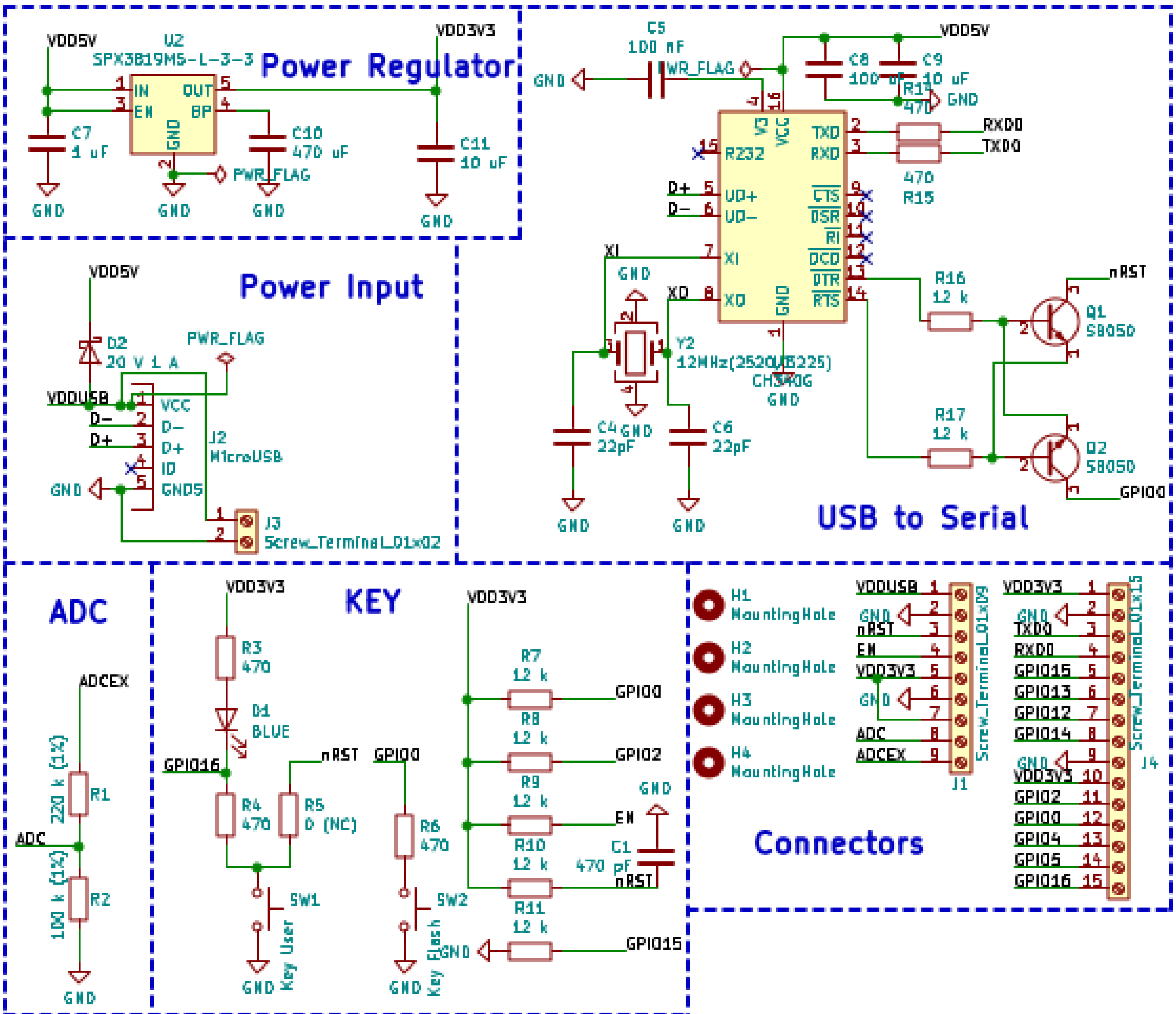


Fig. 3. Schematic diagram of the following connection of Figure 2. The schematic diagram is similar to the NodeMCU. This give us inputs and outputs for more devices.

using the Fast Fourier Transform (FFT) [22] to obtain the frequency of the signal samples to make a supervised learning algorithm [21] to classify the devices connected to our power source.

The learning algorithm that we decided to use is an EANN Model due to that it is well known that this model presents a good performance to classifying patterns [23]. For that reason, we decided to implement two Evolutionary Neural Networks with two different neurons to compare which present the best performance for our classification problem. The first topology is the Evolutionary Hyperplane Neural Network, which is the linear classifier optimised with evolutionary algorithms [24] presented in Figure 8(a), also the Evolutionary Spherical Neural Network [25] which classifies creating spheres by using evolutionary algorithms [24] presented on Figure 8(b). The FFT including the EANN were programmed using the Arduino platform. The training of our EANN model was done off-line.

Additionally, we are trying to find an algorithm that help us reduce the computational load and complexity on our device, for that reason we want to compare the performance of both algorithms.

The topology of our neural network is represented on Figure 10, this topology shows 30 input neurons the number of bins obtained from our histogram of our FFT. Two outputs, which will give us the binary outputs on Tab. I representing the states of our appliances.

The function for our evolutionary spherical neural network is as follows:

$$\mathbf{G}(o) = \frac{1}{1 + exp^{(-\lambda \cdot o)}} \tag{1}$$

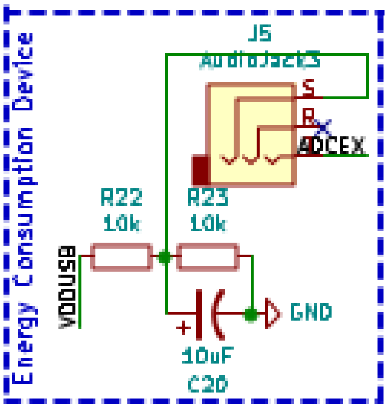


Fig. 4. Schematic diagram of the input connection of our SCT-013 sensor. This allows us to connect the sensor and get the voltage used by a certain device.

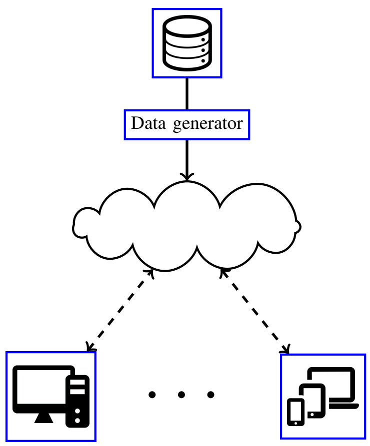


Fig. 5. Cloud computing model.

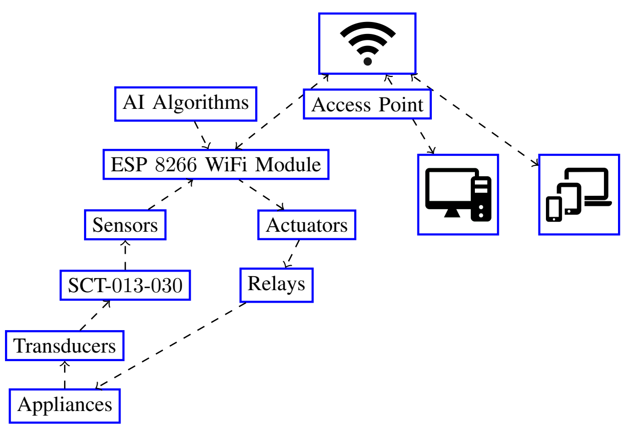
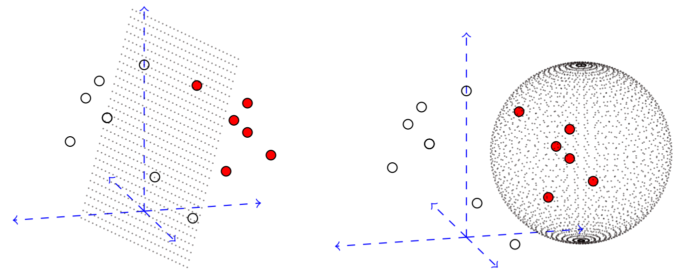


Fig. 7. Diagram of our NILM or NIALM device with all of its components.



(a) Decision region generated for the Evolutionary Hyperplane Neural Network in a 3-dimensional space.

(b) Decision region generated for the Evolutionary Spherical Neural Network in a 3-dimensional space.

Fig. 8. Decision regions generated for a Evolutionary Spherical Neural Network and a Evolutionary Hyperplane Neural Network.

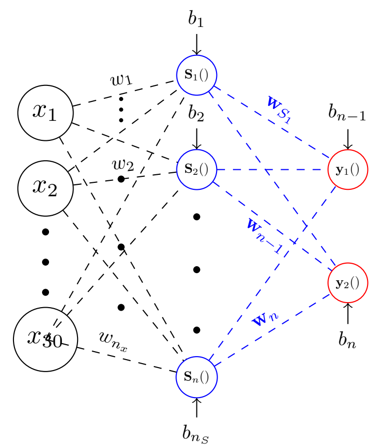


Fig. 9. Evolutionary Hyperplane Neural Network topology.

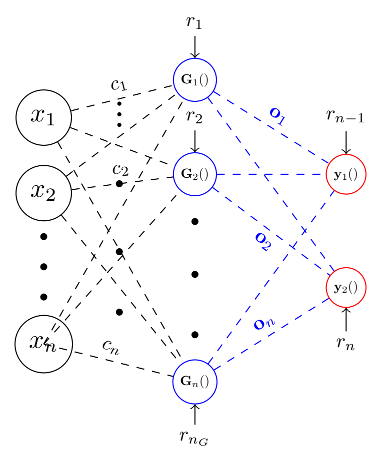


Fig. 10. Evolutionary Hyperspherical Neural Network topology.

where $o = ||\bar{c} - \bar{x}||^2 - r$. λ is the slope of our sigmoid function and it is randomly defined at the beginning the same as the centres and radius of our neural network, and they are estimated with our differential evolution algorithm during training.

Similarly, the function of our Evolutionary Hyperplane Neural Network has the same structure than our function 1:

$$\mathbf{S}(n) = \frac{1}{1 + exp^{(-\lambda \cdot n)}} \tag{2}$$

where $n = \sum xw^t - \theta$. And λ and θ are randomly initialised and estimated with the evolutionary algorithm.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

TABLE I
LABELS FOR THE DIFFERENT CASES IN THE MONITORING OF ENERGY CONSUMPTION. THESE LABELS ARE USED TO INDICATE THE TARGET OUTPUTS OF THE EANN MODELS.

State	Off	Blender	Fridge	On
N_1	0	1	0	1
N_2	0	0	1	1

IV. EXPERIMENTS AND RESULTS

To test our device, we decided to use two different appliances to evaluate and to identify the signature of the signal. These appliances were the fridge and a blender. Figure 11 presents the energy consumption signals and its FFT histogram for both appliances which are obtained in an independently form. These energy consumption signals are obtained in a period of time of 4 minutes and 16 seconds. We get the data every 4 seconds, thus 64 data points are obtained in that period of time. On the fridge plot, we cannot see when the fridge stopped working but we can observe that it has a peak at the beginning, but during the rest of its function we have a regular load of approximately $220\ W$.

An interesting case for our experiments is to analyse the energy consumption signal when both appliances are working. Figure 12(a) and 12(b) present the consumption energy signal when both appliances are working and its FFT histogram respectively.

The features to train the Evolutionary Artificial Neural Networks consist of the bins of FFT histograms. The FFT histograms are calculated using 30 bins. Therefore, the number of inputs to the Evolutionary Artificial Neural Networks is set to 30. For practical purposes, we denote the topologies of Evolutionary Artificial Neural Networks as the following sequence: number of input neurons – a number of hidden neurons – a number of output neurons. The training of Evolutionary Artificial Neural Networks is based on Differential Evolution Algorithm. The experimental setup is given as follows:

- The number of neurons in the hidden layer was varied to obtain the best topology for our Evolutionary Artificial Neural Network. The number of neurons is as follows: 2, 5, 10, 15, 20, 25 and 30.
- We use a 4 fold cross-validation to evaluate the performance of each topology [26].
- We use a database of 68 input patterns which includes four classes. Table I presents the four kinds of target outputs.
- The population for the evolutionary algorithm was set to 100 individuals.
- The initial population was initialised in the domain of $[-100, +100]$
- The stop criteria of the training occurs when the evolutionary algorithm reaches an error of $1e^{-3}$ or 2500 fitness function evaluations.
- The fitness function is the least square function.

Table II presents the results of the classification rate of the Evolutionary Neural Networks under 4 fold cross-validation. The results are presented according to the number of neurons employed in the hidden layer. According to the numerical experiments, the Evolutionary Hyperplane Neural Network obtains the best classification value for the testing data set using only 5 neurons in the hidden layer. On the other hand, the Evolutionary Hypersphere Neural Network presents the best minimum value for the classification when it uses only 2 neurons in the hidden layer. In addition, we present the percentage of classification rate in Table III using the best topology obtained for both models.

We implement the Evolutionary Hyperplane Neural Network in the embedded system using 5 neurons. Note that the mean classification value on both Evolutionary Artificial Neural Networks is much better in the Evolutionary Hyperplane Neural Network than Evolutionary Spherical Neural Network. Thus, we conclude that the best boundaries to form the decision regions for this classification problem are the hyperplanes instead of non-linear decision regions.

TABLE II
RESULTS OF THE CLASSIFICATION RATE OF THE EVOLUTIONARY HYPERPLANE NEURAL NETWORK AND EVOLUTIONARY HYPERSPHERE NEURAL NETWORK UNDER FOUR FOLD CROSS-VALIDATION METHOD. THE RESULTS ARE PRESENTED ACCORDING TO THE NUMBER OF NEURONS IN THE HIDDEN LAYER.

	Evolutionary Hyperplane Neural Network		Evolutionary spherical neural network	
Neurons	Train	Test	Train	Test
2	0.4411	0.0955	0.1024	0.1293
5	0.0882	0.0955	0.1249	0.1621
10	0.0931	0.1323	0.1470	0.2058
15	0.1102	0.1764	0.1519	0.1911
20	0.0931	0.1162	0.1495	0.2132
25	0.1299	0.1470	0.1568	0.1911
30	0.1348	0.1764	0.1593	0.1691
Mean	0.0990	0.1334	0.1417	0.1803
Standard Deviation	0.0304	0.0348	0.0206	0.0289

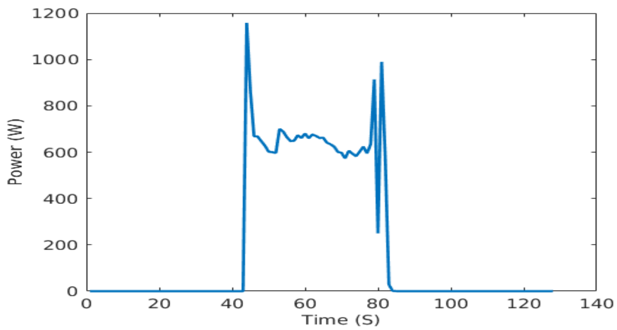
Our first ideas were that the Evolutionary Hyperspherical Neural Network will perform better, because the non-linear decision regions can help to reduce the number of the hidden neurons as presented in [25], [27] with a good classification rate. The Evolutionary Hyperplane Neural Network performed

TABLE III
CLASSIFICATION RATE OF EVOLUTIONARY ARTIFICIAL NEURAL NETWORKS.

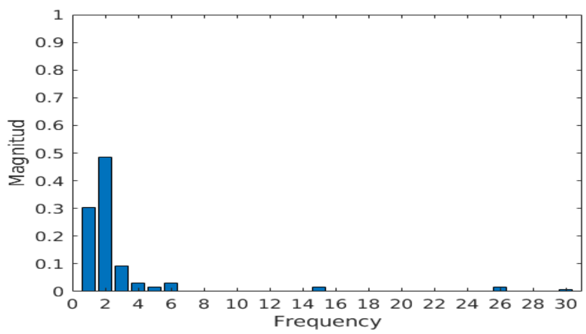
	Evolutionary Hyperplane Neural Network	Evolutionary Hypersphere Neural Network
k-fold	5 neurons	2 neurons
1	82.3529%	47.0588%
2	76.4706%	70.5882%
3	94.1176%	76.4706%
4	76.4706%	88.2353%
Mean	82.3529%	70.5882%

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

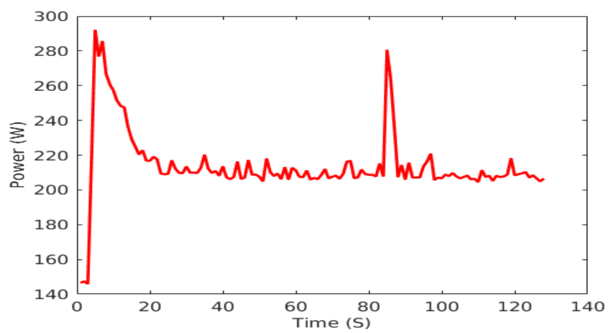
better with 5 neurons in the hidden layer with a success rate of 82% and the Evolutionary Hyperspherical Neural Network performed better with 2 neurons in the hidden layer with a success rate of 70%.



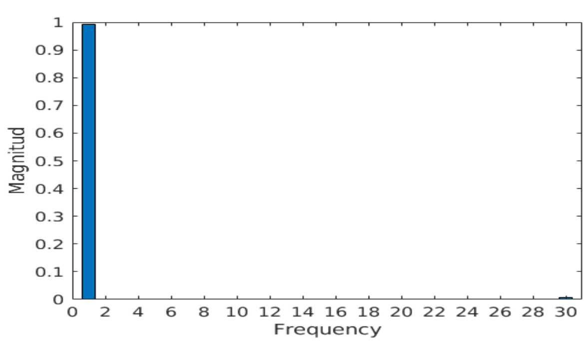
(a) Blender energy consumption signal



(b) FFT histogram of Figure 11(a)

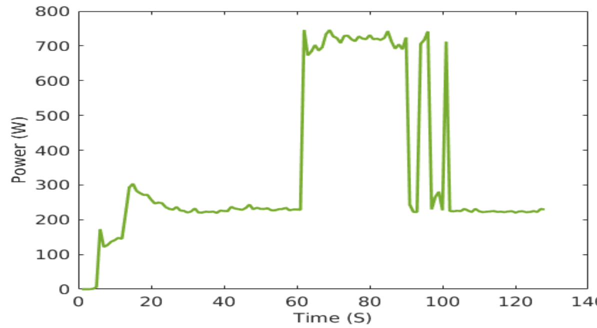


(c) Fridge energy consumption signal

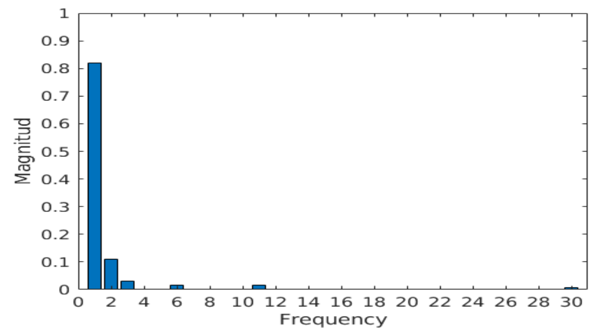


(d) FFT histogram of Figure 11(c)

Fig. 11. Consumption energy signals for the fridge and blender.



(a) Fridge and blender working at the same time consumption signal.



(b) FFT histogram of Figure 12(a)

Fig. 12. Energy signals of the fridge and blender working at the same time.

A. Edge device performance

One of the main topics while implementing edge computing on edge devices is the specifications they have. The edge devices present limitations in comparison with a computer or a server, thus it is necessary to consider the implementation and programming of algorithms in an efficient way. The edge devices present limitations in comparison with a computer or a server, thus it is necessary to consider the implementation and programming of algorithms in an efficient way. We can observe the description of the hardware in section II.

The device we use has 80 KB of RAM memory and we have to perform all operations and keep all the variables there. With the whole implementation of the Fast Fourier Transform, the Histogram, and the Feed-forward operation for our Neural Network with a topology of 30 input neurons, 5 hidden neurons and 2 output neurons, we are consuming 84% of the RAM edge device. Therefore, the rest memory is used for the procedures which operate inside of our program.

If we do not use the RAM memory in an efficient way, some memory problems can appear like overwriting data, incorrect data values. Some of the variables stored on the device are the Twiddle factor matrices, one for real numbers and other for imaginary numbers, and the weights and biases for our neural network topology. These matrices are obtained from the Fast Fourier Transform. The twiddle factor matrices have a length of 64×64 float numbers consuming 32 KB of RAM memory. Other data which is processed by the device corresponds to the values of the energy consumption obtained by the sensor.

This data is collected during a period of 256 seconds, and at the same time, another array is taking the same sample every 4 seconds. The first array contains the whole information of data sampling, and it will be used to inform the user about their energy consumption. The other array of 64 data is just one-fourth of the entire data collected; this array will be used for our neural network to identify the state of the devices connected to our device. This array is processed in various steps, first, we obtain the Fast Fourier transform of it, then we perform the absolute value of the Fast Fourier Transform to normalise the data and also remove the imaginary part from our result. Next, we obtain the histogram of the normalised Fast Fourier Transformation with 30 bins and finally, the data obtained is then sent to the Neural Network for classification.

V. Conclusions

In this paper, a Non-Intrusive Appliance Load Monitoring is presented. Our proposal includes the PCB design of the device and the embedded software which is installed on the device board. The software incorporates an Evolutionary Artificial Neural Network. We implemented two Evolutionary Artificial Neural Network to compare which model obtains a better performance for recognising of four classes of scenarios of energy consumption of appliances than the Evolutionary Hypersphere Neural Network. We use two kinds of appliances: a fridge and a blender.

The importance of the location of this device is that it will reduce significantly the response times, and the bandwidth used to communicate with the cloud server. Our edge device acts as a filter by first processing the data it receives from the sensor, and only sending the information of the energy consumption of the appliance to the cloud server. The information of the sensor, will not interact with our cloud server. Dealing with edge devices, we face several computational problems, which are memory use and processing power. However, as future work of this proposal is to evaluate the capability of the system to carried out the learning on-line.

On the other hand, we could test the training of the Evolutionary Neural Network models complementing with other features to increase the classification rate of the problem.

Acknowledgment

The authors would like to thank the Science and Technology Council of Mexico, CONACyT, for its financial support during this research through the project CB-$2015/256126$, the project FORDECyT 296737 “Consorcio en Inteligencia Artificial” and the SNI Research Assitant Grant 17923. The first two authors would like to thank the Tecnológico Nacional de México/ITS de Irapuato for its support.

References

[1] F. Al-Turjman, *Edge Computing*, 2019.

[2] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, “Edge computing: A survey,” *Future Generation Computer Systems*, vol. 97, pp. 219 – 235, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X18319903

[3] J. Cao, Q. Zhang, and W. Shi, *Introduction. In: Edge Computing: A Primer.* Springer, 2018.

[4] H. Ning, Y. Li, F. Shi, and L. T. Yang, “Heterogeneous edge computing open platforms and tools for internet of things,” *Future Generation Computer Systems*, vol. 106, pp. 67 – 76, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X19312841

[5] R. S. Tol, “The economic effects of climate change,” *Journal of economic perspectives*, vol. 23, no. 2, pp. 29–51, 2009.

[6] Y. Li, W. A. Pizer, and L. Wu, “Climate change and residential electricity consumption in the yangtze river delta, china,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 2, pp. 472–477, 2019.

[7] W. B. Group, “Electric power consumption (kwh per capita).” [Online]. Available: https://data.worldbank.org/indicator/EG.USE.ELEC.KH.PC

[8] “U.s. energy information administration - eia - independent statistics and analysis,” Oct 2019. [Online]. Available: https://www.eia.gov/tools/faqs/faq.php?id=77t=11

[9] “December 2019 monthly energy review,” *Monthly Energy Review*, Dec 2019. [Online]. Available: http://www.eia.gov/totalenergy/data/monthly

[10] S. Darby, “The effectiveness of feedback on energy consumption,” *A Review for DEFRA of the Literature on Metering, Billing and direct Displays*, vol. 486, 01 2006.

[11] T. Ueno, R. Inada, O. Saeki, and K. Tsuji, “Effectiveness of displaying energy consumption data in residential houses. analysis on how the residents respond,” 01 2005.

[12] K. Armel, A. Gupta, G. Shrimali, and A. Albert, “Is disaggregation the holy grail of energy efficiency? the case of electricity,” *Energy Policy*.

[13] A. Kumar and P. Bhattacharjee, “Non-intrusive appliance identification for energy disaggregation of indian households–an use case for energy informatics,” in *2018 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, Dec 2018, pp. 239–242.

[14] C. Zhou, S. Liu, and P. Liu, “Neural network pattern recognition based non-intrusive load monitoring for a residential energy management system,” in *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*, July 2016, pp. 483–487.

[15] G. W. Hart, “Nonintrusive appliance load monitoring,” *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec 1992.

[16] N. Team, “Nodemcu-an opensource firmware based on esp8266 wifi-soc,” *URL https://www.nodemcu.com/index*$_{e}n.html$, 2014.

[17] “Nodemcu v3.” [Online]. Available: https://docs.zerynth.com/latest/official/board.zerynth.nodemcu3/docs/index.html

[18] C. Bell, *MicroPython for the Internet of Things: A Beginner’s Guide to Programming with Python on Microcontrollers*, 01 2017.

[19] J. Cao, Q. Zhang, and W. Shi, *Introduction*. Cham: Springer International Publishing, 2018, pp. 1–9.

[20] A. F. Arguimbau, “Algorithms for energy disaggregation,” 2016.

[21] M. Aiad and P. H. Lee, “Unsupervised approach for load disaggregation with devices interactions,” *Energy and Buildings*, vol. 116, pp. 96 – 103, 2016.

[22] D. Takahashi, *Fast Fourier Transform*. Singapore: Springer Singapore, 2019, pp. 5–13.

[23] J. M. Bishop and R. J. Mitchell, “Neural networks-an introduction,” in *IEEE Colloquium on Neural Networks for Systems: Principles and Applications*, Jan 1991.

[24] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: a Practical Approach to Global Optimization*. Springer-Verlag Berlin and Heidelberg Co. KG, 2006.

[25] J. P. Serrano-Rubio, A. Hernandez-Aguirre, and R. Herrera-Guzman, “Hyperconic multilayer perceptron,” *Neural Processing Letters*, vol. 45, no. 1, pp. 29–58, 2017.

[26] P. Refaeilzadeh, L. Tang, and H. Liu, *Cross-Validation*. New York, NY: Springer New York, 2018, pp. 677–684.

[27] J. P. Serrano-Rubio, R. Herrera-Guzmán, and A. Hernández-Aguirre, “Hyperconic multilayer perceptron for function approximation,” in *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, Nov 2015, pp. 004 702–004 707.