



Article

Identification of Heart Sounds with an Interpretable Evolving Fuzzy Neural Network

Paulo Vitor de Campos Souza ^{*,†}  and Edwin Lughofer [†] Department of Knowledge-Based Mathematical Systems, Johannes Kepler University Linz
Altenberger Strasse 69, 4040 Linz, Austria; edwin.lughofer@jku.at

* Correspondence: paulo.de_campos_souza@jku.at

† These authors contributed equally to this work.

Received: 20 October 2020; Accepted: 10 November 2020; Published: 12 November 2020



Abstract: Heart problems are responsible for the majority of deaths worldwide. The use of intelligent techniques to assist in the identification of existing patterns in these diseases can facilitate treatments and decision making in the field of medicine. This work aims to extract knowledge from a dataset based on heart noise behaviors in order to determine whether heart murmur predilection exists or not in the analyzed patients. A heart murmur can be pathological due to defects in the heart, so the use of an evolving hybrid technique can assist in detecting this comorbidity team, and at the same time, extract knowledge through fuzzy linguistic rules, facilitating the understanding of the nature of the evaluated data. Heart disease detection tests were performed to compare the proposed hybrid model's performance with state of the art for the subject. The results obtained (90.75% accuracy) prove that in addition to great assertiveness in detecting heart murmurs, the evolving hybrid model could be concomitant with the extraction of knowledge from data submitted to an intelligent approach.

Keywords: evolving fuzzy neural network; heart murmur; SOF; pattern classification problem

1. Introduction

Society seeks to work on solutions that reduce the negative impacts of groups of diseases on human beings routinely. Among these diseases, those that attack the heart stand out, and may decrease the standard of living when playing sports or daily activities, and even cause death [1]. Heart diseases are constant targets for applying artificial intelligence in solving problems arising from heart diseases, thereby identifying patterns that can be useful for medical teams to initiate treatments or surgical procedures [2].

A heart murmur is a comorbidity related to blood flow passage through the heart, generating sounds that can identify a circulatory problem regarding the heart's functions. They can be identified in both children and adults, but for the most part, they can be just innocent murmurs with no other changes and with absolutely normal physical development. Another form of this heart murmur is due to complications of heart diseases that generally occurred in childhood, and care must be taken on the part of doctors so that the treatment to be carried out is appropriate for the maintenance of a healthy life on the patient's part [3].

A heart murmur can exist in two ways: physiological or congenital; the second does not have a precise explanation for its cause. A heart murmur (congenital or acquired) needs special attention and adequate tests for its identification in time. Its main reasons for existence are linked to changes in the heart valves; communication between the aorta and the pulmonary artery; and rheumatic fever (caused by the body's immune reaction against antigens or components of the streptococcus). It should be noted that degenerative diseases can also lead to a heart murmur [3].

The diagnosis of this disease is carried out by identifying heart sounds, comparing them with the frequency of familiar heart sounds. There are other forms, such as the electrocardiogram, chest X-rays and echocardiogram. However, this paper focuses on identifying heart murmurs by identifying sound patterns coming from the heart. As it is a disease that can be asymptomatic, the correct and effective diagnosis must be present in the routines of the patient and the medical team because in most severe cases of the disease, surgical interventions are fundamental for the maintenance of a good standard of life for the patient.

Several types of research are conducted jointly by artificial intelligence (AI) and the medical field. This type of partnership generates artifacts that can improve research or find fundamental patterns for improving the identification of diseases. Regarding the heart murmur problem, there is a public dataset with healthy heart sounds and a murmur. This dataset provided by Liu et al. [4] assists in constructing intelligent systems and techniques capable of assisting medical teams with diagnosing a heart murmur. Therefore, this dataset supported several surveys carried out to determine anomalies in heart sounds.

Intelligent approaches in detecting heart murmur are frequent in science, as in Rud et al. [5], which uses support vector machines to detect heart murmurs. Quiceno-Manrique et al. [6] and Mukherjee et al. [7] use a feature selection approach to define the best features to correctly classify heart murmurs using phonocardiographic signals or wavelet transform [8] proposed by Kumar et al. and Avendano-Valencia et al. [9], which uses time-frequency representations to define features about the problem.

Despite the growing amount of research in the area of identification of cardiac sounds, few models can bring assertive answers about the problem, and at the same time, extract knowledge about the evaluated data. With rare exceptions (such as the work of Soares et al. [10]), there are few logical relationships generated through the data submitted to the models. To solve this deficiency, hybrid models based on neural networks and fuzzy systems are used for two main processes in parallel: to obtain a high accuracy for the problem while extracting knowledge through fuzzy rules. This paper proposes the creation of an evolving three-layer model capable of solving problems related to heart murmurs. The first layer of the model builds Gaussian neurons, where a self-organizing technique [11] obtains the centers and sigmas based on empirical operators [12]. The first layer neurons' weights are defined by a leave-one-feature-out (LOFO) technique, which is based on a separability criterion measuring the impact of each feature in terms of its contribution to the discriminatory power of the whole feature set [13]. In the second layer, in the definition of fuzzy rules, logical fuzzy neurons are used to aggregate neurons in the first layer. A neural aggregation network is interconnected with the second layer through output weights estimated by a weighted recursive least squares technique, with weights equal to the fuzzy rules' activation degrees generated in the second layer of the model. The neural aggregation network is responsible for the process of defuzzification and for obtaining answers about the identification of heart murmurs.

This paper expects to use an evolving fuzzy neural network based on logical neurons (EFNN-LN) to identify cardiac problems based on sound evaluations of the heart. Unlike related works (as discussed above), our approach is able to extract linguistically readable fuzzy rules from the data and thus allows insight for the experts (new knowledge, better understanding of the process, etc.)—which we will demonstrate in the results section by showing the extracted rules for the two classes (heart murmur or no heart murmur) in if-then form. Thus, one main contribution of this paper is the extraction of readable rules to understand heart murmur problems (in which situations they occur, i.e., in which combinations of feature value ranges). Another contribution is the increased readability of rules by providing feature weights: features with low weights can be ignored and thus the rule length reduced → more compact representation; furthermore, the impacts of features in the rules may give clues for experts as to which features can be seen as more and which ones as less important to describe the classes. Furthermore, the fuzzy neurons are allowed to be constructed through advanced aggregation operators, apart from conventionally used t-norms in current related

evolving fuzzy neural network approaches. This induces more flexibility in terms of allowing both AND* and OR connections of the rule antecedent parts (where in conventional approaches typically only AND connections are induced).

The paper is composed as follows: Section 2 presents the main theoretical concepts about the essential components in our model. The ideas of the evolving fuzzy neural network architecture, the fuzzy neuron and all aspects related to incremental, evolving training procedures of our neuro-fuzzy system's components and parameters will be demonstrated. In Section 3, information about the dataset to be used in the experiment is presented to the reader. In Section 4, all the results in comparison with related SoA works and terms of achieved rules (interpretation) are presented to solve the heart sound problems. Finally, Section 5 highlights the conclusions about the study.

2. Evolving Fuzzy Neural Networks and Our Approach: EFNN-LN

Fuzzy neural networks [14] are hybrid models that work in solving problems with knowledge extraction. These intelligent models can extract knowledge through fuzzy rules based on the dataset evaluated by the model. The synergy between artificial neural networks that provide forms of training to solve problems, and fuzzy systems, which foster arguments to facilitate problems' interpretability, works to solve complex problems of different natures, mainly in the health area. Hybrid models are present in several intelligent algorithms for solving complex problems. They work in synergy through the use of the most considerable advantages in the techniques used. Fuzzy neural network models and neuro-fuzzy networks are relevant highlights for this category [15].

These models work with several classes of problems related to robotics, time series prediction, linear regression, dynamic systems, etc. Their main difference compared to traditional intelligent models is the ability to extract knowledge from rules. This knowledge base extracted from the data can be verified with professionals experienced in solving these problems so that this set of rules can create expert systems or promote the dissemination of knowledge in companies through training on the aspects raised by the hybrid models [16]. The evolving fuzzy neural networks follow the same architectural logic as the conventional fuzzy neural networks; however, the parameters, fuzzification and defuzzification methods are based on adaptive and evolving algorithms for updating parameters and gaining new knowledge on demand and on the fly whenever new data samples are submitted to the training procedure [17]. This can be usually done in a single-pass manner [18], thereby allowing the models to be incrementally trained from (fast) streaming data [19]. This opens up enormous flexibility when system state or patients behaviors change (which typically result in concept drift issues [20]), as these changes can be quickly integrated without using old data (so no time-intensive re-training is required). These models have already been used to solve problems in the health area—for instance, take Iglesias et al. with the work on recognition of human activities during daily life [21,22], and Malcangi et al. who used an evolving hybrid model to solve problems in seismocardiograms to control and report heartbeats in everyday activities [23]. Another paper that uses evolving hybrid approaches is related to stroke control proposed by Kasabov et al. [24]. Other work related to performance in the health area can be seen in two recent surveys proposed by de Campos Souza [16] and Škrjanc et al. [17].

This paper's evolving hybrid model has three layers, where the first is responsible for the fuzzification process. The second layer is responsible for building the fuzzy rules and training the model. It is also noteworthy that the first two layers of the model are seen as a fuzzy inference system, capable of transforming data into knowledge. Finally, the third layer is responsible for the defuzzification process through the weights of the fuzzy rules combined with the model inputs. This layer is made up of artificial neurons with linear activation functions and can be seen as a neural aggregation network. Figure 1 schematically presents the model architecture proposed in this paper. Next, the subsequent steps for constructing the fuzzy neural network with evolution based on logical neurons (EFNN-LN) will be defined.

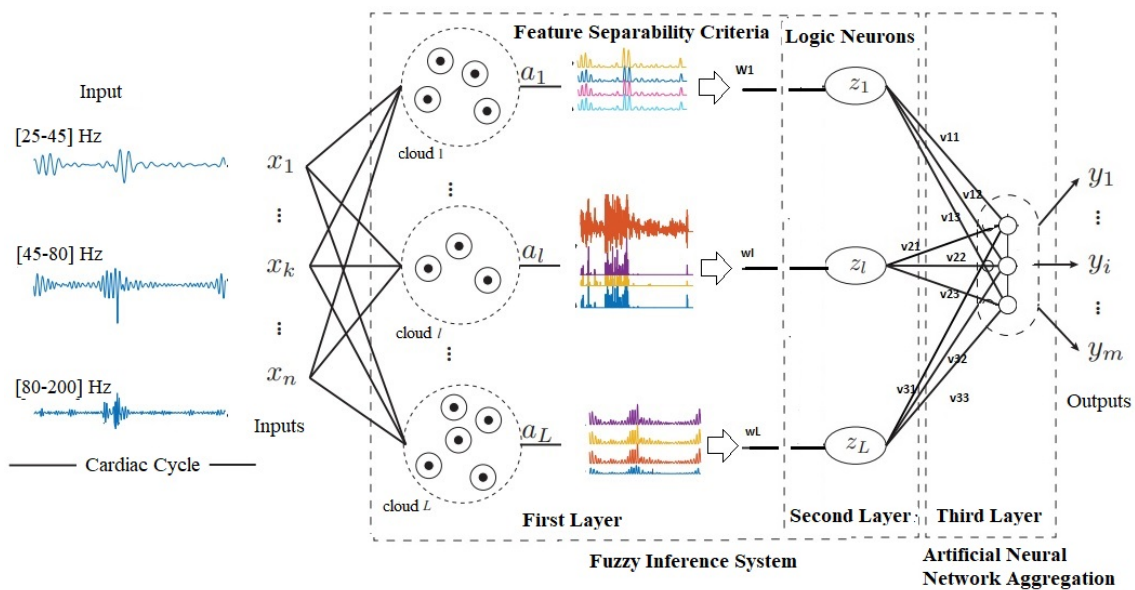


Figure 1. Evolving fuzzy neural network architecture.

2.1. First Layer—Fuzzification Process and Gaussian Neurons

The first layer is composed by L fuzzy neurons built by the fuzzification process described in the hereafter. All fuzzy sets created during the input space granulation are considered Gaussians, with the modal values measured as the cloud centers' projections in the corresponding dimensions. This process of determining neurons in the first layer follows the premises defined in the models reported in [25]. However, a fixed number of neurons is used, which is pre-parameterized according to trial and error tuning phases. The use of our evolving approach opens up the possibility to autonomously seek the adequate number of neurons (according to the nature of the data stream to describe the classes sufficiently, etc.) to minimize the exponential relationship impact of fuzzy neurons created between the number of membership functions chosen for the model and the number of features in the dataset. As a result, it should create more compact models (neurons are evolved when actually needed).

The automatic definition of intelligent model architecture is not a new issue. It has been the subject of study mainly in the neural architecture search (NAS) [26], a technique proposed by Zoph and Le [27] and Baker et al. [28] that works on intelligent models' architectural definitions. They use techniques such as search space (set of operations such as convolutions or fully connected networks) to form valid network architectures or define previous architectures through optimization techniques and finally evaluation strategies to verify the model's efficiency in solving problems. The evolving fuzzy neural network used in this paper uses the same idea of the automatic definition of architecture based on data. However, it uses the fuzzification process defined by data density to build neurons representing interpretability to the problem. Thus, it can be noted that the difference between the NAS and the proposed model is that the architectural definition of the NAS is based on the optimal network to solve the problem, whereas in our evolving fuzzy neural network, the input data are transformed into fuzzy sets. These result in a model architecture which is linguistically interpretable, as can be seen in the results section, where there are examples showing the rules learned from real-world heart sounds classification problems. The neurons in the model proposed in the paper, differently from the NAS approach, also define the best set of neurons to make the problem interpretable. Furthermore, our approach is an adaptive and evolving approach, meaning that it can elegantly act on data streams or additional datasets by updating the model solely with new samples, without requiring (possibly time-intensive) re-training phases from scratch. To the best of our knowledge, this has not been considered in NAS approaches so far.

Another difference from previous related works is related to the weights of Gaussian neurons. Previously published works defined the weights to construct the neurons at random in the interval between 0 and 1. However, this does not facilitate interpretability, because for different training runs different weights will be extracted, leading to non-deterministic results. This may be also the case when the weights are just randomly initialized, but further learned from the data, because different initializations often lead to different (local) solutions, as finding a global unique solution within an (numerical) optimization context is for many (real-world, noisy) datasets an unresolved issue. To solve this situation, a deterministic approach based on an advanced separability criterion will be used in this paper. It calculates feature weights from the data w_{il} (for $i = 1 \dots n$ and $l = 1 \dots L$), which depend on the importance of features according to their contributions to the discriminatory power of the whole feature set with respect to the classes contained in the data. The discriminatory power will be measured by an extended statistical criterion based on Fisher's separability criterion, which is known to be deterministic.

For training the first layer, we evaluate the samples and number of dimensions of the dataset, where for every input variable x_i , L fuzzy sets are represented by $A_i^l, l = 1, \dots, L$, whose membership functions are the activation functions of the corresponding neurons. Thus, the outputs of the initial layer are the membership degrees correlated with the input values, i.e., $a_{il} = v(A_i^l)$ for $i = 1, \dots, n$ and $l = 1, \dots, L$, where n is the number of inputs and L represents the number of fuzzy sets (neurons) for each input. The centers' coordinate-wise values and the dispersions (σ) define these sets, which are obtained from the clouds formed by the evolving approach.

Therefore, it is expected that the Gaussian neurons of the first layer will be built according to the fuzzification process. In this paper, the Gaussian neurons of the first layer are expressed by:

$$a_{il}(x_i, c_{il}, \sigma_{il}) = e^{-\frac{1}{2} \left(\frac{x_i - c_{il}}{\sigma_{il}} \right)^2}, \text{ for } j = 1 \dots n, l = 1 \dots L \quad (1)$$

where c_{il} and σ_{il} are respectively the center and width of the l th Gaussian function for input variable i .

2.1.1. Fuzzification Approach: Self-Organising Fuzzy Logic

Fuzzification approaches are extremely important for the operation of hybrid models, especially fuzzy neural networks. This step is responsible for transforming discrete inputs from datasets into fuzzy outputs, with different degrees of relevance to fuzzy sets representing linguistic terms. From this process, the factors to compose the Gaussian neurons projected in the centers and sigmas are obtained through the fuzzification procedure.

In this paper, a non-parametric approach to fuzzification proposed in Gu and Angelov [11] is used. The self-organizing technique identifies clouds from the observed data through an offline training process, allowing the fuzzy neural network model's architecture to represent the assessed problem's data. Unlike the model's initial approach, in this paper, we will only use the stage of identification of clouds through empirical data analytics operators (EDA), discarding the procedures related to the classifier based on a 0-order AnYa type fuzzy rule-based system. In addition to the offline approach, we will also use the online version of the model, thereby allowing the hybrid model proposed in this paper to evolve on new stream samples to account for possible changes (drifts).

The following basic definitions of EDA will be referenced throughout the paper:

- $\mathbf{x} = \{x_1, x_2, \dots, x_k\} \in \mathbb{R}^d$: input variables. (where the index k indicates the time instance at which the data point arrives.)
- $\mathbf{u} = \{u_1, u_2, \dots, u_L\} \in \mathbb{R}^d$: set of unique data point locations.
- f_1, f_2, \dots, f_L : number of times that different data points occupy the same unique locations.
- K^c : the number of data samples $\{\mathbf{x}\}_{K^c}^c$ belonging to the c -th class.
- U_K^c : the number of unique data samples belonging to the c -th class.
- $\sum_{c=1}^C K^c = K$ and $\sum_{c=1}^C U_K^c = U_K$.

Based on u_1, u_2, \dots, u_L and f_1, f_2, \dots, f_L , it is possible to reconstruct the dataset x_1, x_2, \dots, x_k exactly if necessary, regardless of the order of arrival of the data points [12].

The SOF technique uses relevant concepts such as cumulative proximity, an empirical derivation of the model's data without prior knowledge or previous assumptions being seen as a square form of distance. It can be defined by [29]:

$$\pi_K(x_i) = \sum_{j=1}^K d^2(x_i, x_j); \quad i = 1, 2, \dots, K \quad (2)$$

where $d(x_i, x_j)$ denotes the Euclidean distance between x_i and x_j [29].

Another fundamental factor for the determination of data clouds is the unimodal density, which in turn is defined by [29]:

$$D_K(x_i) = \frac{\sum_{l=1}^K \pi_K(x_l)}{2K\pi_K(x_i)} = \frac{\sum_{l=1}^K \sum_{j=1}^K d^2(x_l, x_j)}{2K \sum_{j=1}^K d^2(x_i, x_j)}; \quad i = 1, 2, \dots, K \quad (3)$$

Finally, the third EDA operator relevant to the fuzzification approach used in this paper is the multimodal density, which can be estimated through a unique data sample, u_i , through the weighted sum of its unimodal density by its repetition times. It is calculated using [29]:

$$D_K^{MM}(u_i) = f_i D_K(u_i) = f_i \frac{\sum_{l=1}^K \pi_K(x_l)}{2K\pi_K(u_i)}; \quad i = 1, 2, \dots, U_K \quad (4)$$

These three EDA operators $\pi_K(x_i)$, $D_K(x_i)$ and $D_K^{MM}(u_i)$ are fundamental to the recursive procedures to be performed by the fuzzification approach. They allow models that use the technique to act dynamically with data in the stream. The values of the EDA elements listed above can be updated using different types of distances/dissimilarity. In this paper, we will use the cosine distance, as suggested in Gu and Angelov [29]. Ways of updating the parameters recursively can be found in Angelov [12].

The (online) update and evolving procedure takes place by category, where SOF can identify each class's prototypes separately and form a fuzzy rule of type AnYa of order 0 based on the prototypes identified per class. These prototypes are identified based on the teaching and mutual distributions of the data samples, through calculating the multimodal density in all the unique data samples using the following equation [29]:

$$D_{K^c}^{MM}(u_i^c) = f_i^c \frac{\sum_{l=1}^{K^c} \sum_{j=1}^{K^c} d^2(x_l^c, x_j^c)}{2K^c \sum_{j=1}^{K^c} d^2(u_i^c, x_j^c)}, \quad i = 1, 2, \dots, U_{K^c}^c \quad (5)$$

where c represents c -th class, C , $\{x\}_{K^c}^c = \{x_1^c, x_2^c, \dots, x_{K^c}^c\}$ $\{x\}_{K^c}^c \subset \{x\}_K$ are the data samples of the c -th class and $\{u\}_{U_{K^c}^c}^c = \{u_1^c, u_2^c, \dots, u_{U_{K^c}^c}^c\}$ and $\{f\}_{U_{K^c}^c}^c = \{f_1^c, f_2^c, \dots, f_{U_{K^c}^c}^c\}$ correspond to a unique data sample set and frequencies of occurrence, respectively. K^c is the number of data samples of the c -th class and $U_{K^c}^c$ is the number of unique data samples of the c -th class [29].

After this definition, a list \mathbf{r} is used to organize previously calculated values, mainly between mutual distances and multimodal density values, defining as r_1 the element with the highest modal density. Therefore, the next elements, $r_2, r_3 \dots r_n$, are calculated from the sample of data with the minimum distance to the previously indexed element. Therefore, r_2 is related to r_1 , just as r_3 is related to the distance to r_2 , and so on, until all samples are indexed [29].

After indexing the terms in the list \mathbf{r} , the list of prototypes \mathbf{p} is defined, which are then identified as the local maximum of the multimodal densities classified through the following condition [29]:

$$IF (D_{K^c}^{MM}(r_i) > D_{K^c}^{MM}(r_{i+1})) AND (D_{K^c}^{MM}(r_i) > D_{K^c}^{MM}(r_{i-1})) THEN (r_i \in \mathbf{p}) \quad (6)$$

After determining all the prototypes, the relevant ones are discarded through a filtering process. This process begins with prototypes to attract samples of nearby data to form clouds of data. This operation can be expressed by [29]:

$$\forall p = \arg \min_{p_n \in p} (d(x_i, p_n)); \quad x_i \in \{x\}_{K^c}^c \quad (7)$$

where $\forall p$ is the winning prototype. After the formation of all data clouds formed around the existing prototypes, the clouds' centers φ_i are formed. The multimodal values of the centers are also calculated and defined as follows [29]:

$$D_{K^c}^{MM}(\varphi_i) = S_i D_{K^c}(\varphi_i) \quad (8)$$

where S_i is the support (number of members) of the i th data cloud. Then, for each data cloud, the collection of the centers of its neighboring data clouds is identified using the following principle [29]:

$$IF \left(d^2(\varphi_i, \varphi_j) \leq G_{K^c}^{c,L} \right) \quad THEN \left(\varphi_j \in \{\varphi\}_i^{neighbouring} \right) \quad (9)$$

where $\varphi_j \neq \varphi_i$ and $G_{K^c}^{c,L}$ the average radius of the influential local area around each data sample, which corresponds to the L th level of granularity and is derived from the data of the c -th class based on the user's choice in an offline way [29]. Finally, the most representative prototypes of the c th class, denoted by p_c , are selected from the centers of the existing data clouds that satisfy the following rule [29]:

$$IF \left(D_{K^c}^{MM}(\varphi_i) > \max_{\varphi \in \{\varphi\}_i^{neighbouring}} (D_{K^c}^{MM}(\varphi)) \right) \quad THEN (\varphi_i \in p_c) \quad (10)$$

Thus, the offline procedure of the data defines the centers that will reference the Gaussian projection that will be part of the first layer's fuzzy neurons.

In each new sample submitted for evaluation in the online phase of the SOF, the SOF parameters are updated, allowing the approach to be consistent with the data streaming evaluation. Recursive updating allows parameters to be modified as new data samples come in and are presented to the training procedure. In this stage, the average radius of the local areas of influence is subsequently updated recursively. In online training, considering the new $K + 1$ th instance, where a new data sample of the c -th class is represented by $x_{K^c+1}^c$, the parameters $\mu_{K^c+1}^c$, $X_{K^c+1}^c$, and $\Sigma_{K^c+1}^c$ of the c -th class will be updated recursively by [29]:

$$\mu_{K^c+1}^c = \frac{K^c}{K^c+1} \mu_{K^c}^c + \frac{1}{K^c+1} x_{K^c+1}^c; \quad \mu_1^c = x_1^c \quad (11)$$

$$X_{K^c+1}^c = \frac{K^c}{K^c+1} X_{K^c}^c + \frac{1}{K^c+1} x_{K^c+1}^{cT} x_{K^c+1}^c; \quad X_1^c = x_1^{cT} x_1^c \quad (12)$$

$$\Sigma_{K^c+1}^c = \frac{K^c + 1}{K^c} \left(X_{K^c+1}^c - \mu_{K^c+1}^{cT} \mu_{K^c+1}^c \right) \quad (13)$$

Please note that the centers and covariance matrices are updated class-wise independently, to achieve class-layered cloud partitions with N^c prototypes for the c -th class.

In the recursive update process, the average radius of local areas of influence ($G_{K^c}^{c,L}$) (as used in Equation (9)) is updated a posteriori based on the ratio between $d_{K^c}^c$ and $G_{K^c}^{c,L}$ using:

$$G_{K^c+1}^{c,L} = \frac{d_{K^c+1}^c}{d_{K^c}^c} G_{K^c}^{c,L} = \frac{\frac{1}{(K^c+1)^2} \sum_{l=1}^{K^c+1} \pi_{K^c+1}(x_l^c)}{\frac{1}{(K^c)^2} \sum_{l=1}^{K^c} \pi_{K^c}(x_l^c)} G_{K^c}^{c,L} \quad (14)$$

The new sample evaluated by the SOF allows the approach to verify whether it should be come a new prototype. To do so, it must meet the following condition [29]:

$$IF \left(D_{K^{c+1}}(x_{K^{c+1}}^c) > \max_{p \in \{p\}^c} (D_{K^{c+1}}(p)) \right) OR \left(D_{K^{c+1}}(x_{K^{c+1}}^c) < \min_{p \in \{p\}^c} (D_{K^{c+1}}(p)) \right) \quad (15) \\ THEN (x_{K^{c+1}}^c \text{ becomes a new prototype})$$

If Equation (15) is not fulfilled, another condition is evaluated [29]:

$$IF \left(\min_{p \in \{p\}^c} (d^2(x_{K^{c+1}}^c, p)) > G_{K^{c+1}}^{c,L} \right) \quad THEN (x_{K^{c+1}}^c \text{ becomes a new prototype}) \quad (16)$$

If the result is positive for this condition, a new prototype is added to the fuzzy rule of the c -th class, and the corresponding new data cloud with meta-parameters initialized by Equation (17) is added to the SOF classifier [29].

$$N^c \leftarrow N^c + 1; \quad p_{N^c}^c \leftarrow x_{K^{c+1}}^c; \quad S_{N^c}^c \leftarrow 1; \quad \{p\}^c \leftarrow \{p\}^c + p_{N^c}^c \quad (17)$$

where N^c = number of clusters/clouds. Otherwise, it will be assigned to the nearest prototype, and the corresponding data cloud meta-parameters are updated as follows [29]:

$$p_{n^*}^c \leftarrow \frac{S_{n^*}^c}{S_{n^*}^c + 1} p_{n^*}^c + \frac{1}{S_{n^*}^c + 1} x_{K^{c+1}}^c; \quad S_{n^*}^c \leftarrow S_{n^*}^c + 1 \quad (18)$$

with $p_{n^*}^c$ being the nearest prototype to $x_{K^{c+1}}^c$ in the c -th class layer and $S_{n^*}^c$ its support (the number of past samples belonging to it).

For each cloud built by SOF, a Gaussian neuron is created. In this case, the choice of a nonparametric approach determined by the density of the problem data, allows the evolving fuzzy neural network architecture to be optimized to resolve the proposed problem. Figure 2 shows the global density of two features of the heart sounds problem.

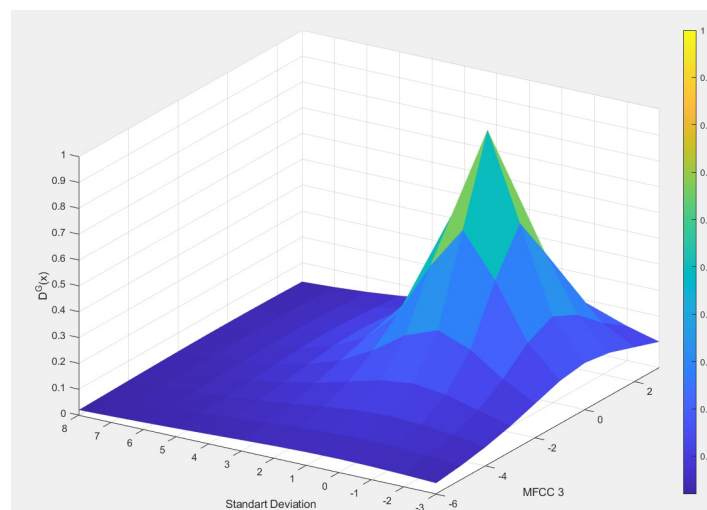


Figure 2. Global density visualization between standard deviation and MFCC 3; in this case one Gaussian neuron would be sufficient to describe the density distribution adequately.

2.1.2. Weight Definition: Leave-One-Feature-Out Criterion

The first layer Gaussian neurons are formed with the centers and the SOF algorithm's sigma value. These values allow the construction of the representation of the data in a fuzzified way. For the neuron's composition, random values in the range between 0 and 1 are commonly used in fuzzy neural network approaches. However, this can impair the model's interpretability. Therefore, we propose an alternative way, that is, to extract feature weights, which indicate their discriminatory power among

the classes and thus their importance for the learning problem and interpretation of the relationships between the input space and the classes.

The central approach is to extract the feature weights based on the use of a class separability criterion, which in Lughofer [13] is established based on a robust expansion of Fisher's separability criterion [30]. This criterion can be defined by:

$$J = \text{trace}(S_w^{-1}S_b) \quad (19)$$

where $\text{trace}(S_w^{-1}S_b)$ defines the sum of the diagonal elements of the matrix $S_w^{-1}S_b$ [13]. S_b expresses the dispersion matrix between classes that measures the class's dispersion averages to the total average, and S_w denotes the within scattering matrix that measures the samples' dispersion in their class averages. The latter can be determined through the average of the class-wise covariance matrices for exact formulas; see [13]. The higher J becomes, the better the separability of the classes.

Our approach is then to use the criterion in Equation (19) for determining the weight of each feature for discriminating between the classes. Thereby, we employ the leave-one-feature-out (LOFO) approach, which discards each feature from the complete set and then calculates Equation (19) for each obtained subset \rightarrow obtaining J_1, \dots, J_N for N features in the dataset. This means that a higher drop in Equation (19) (and thus a lower value of J_i) indicates that feature i is more important as it was left out. Thus, and aiming for a relative importance among all features (relative to the most important feature, which should receive a weight of 1), the feature weights are assigned by:

$$w_j = 1 - \frac{J_j - \min_{i=1, \dots, N} J_i}{\max_{i=1, \dots, N} J_i} \quad (20)$$

That guarantees that the feature causing the minimal value of J_j (hence most important one) gets a weight of 1, whereas all others are relatively compared to the minimum: when they are close, the weight will also be almost 1; in the case when they are significantly higher, the weight will be forced towards the ratio between minimum and maximum.

The separability criterion's incremental update acts on the matrices represented in S_w and S_b . These matrices can be updated during the incremental mode uniquely. In particular, S_b can be updated by updating the class-wise and the overall mean through an incremental mean formula. Recursive covariance update formulas update the covariance matrices per class using rank-1 modification for more robust and faster convergence to the batch calculation [31]; see [13] for further details.

This approach allows the neuron's weight that represents the problem's feature as consistent with the nature of the data evaluated. If this dimension is hugely relevant to solving classifying patterns, that neuron will have a very expressive value (close to 1). The dimension that is less relevant to the differentiation of the classes of the problem, on the other hand, will have its value closer to zero. This factor presented facilitates the construction of better interpretable rules, as features with low weights can be discarded from the antecedents when showing the rules to experts (reduction of rule length, better readability). Figure 3 demonstrates the technique presented in the dataset used in this paper. It is possible to identify that MFCC 4 and MFCC 3 features are the most relevant features of this problem, according to the LOOF technique.

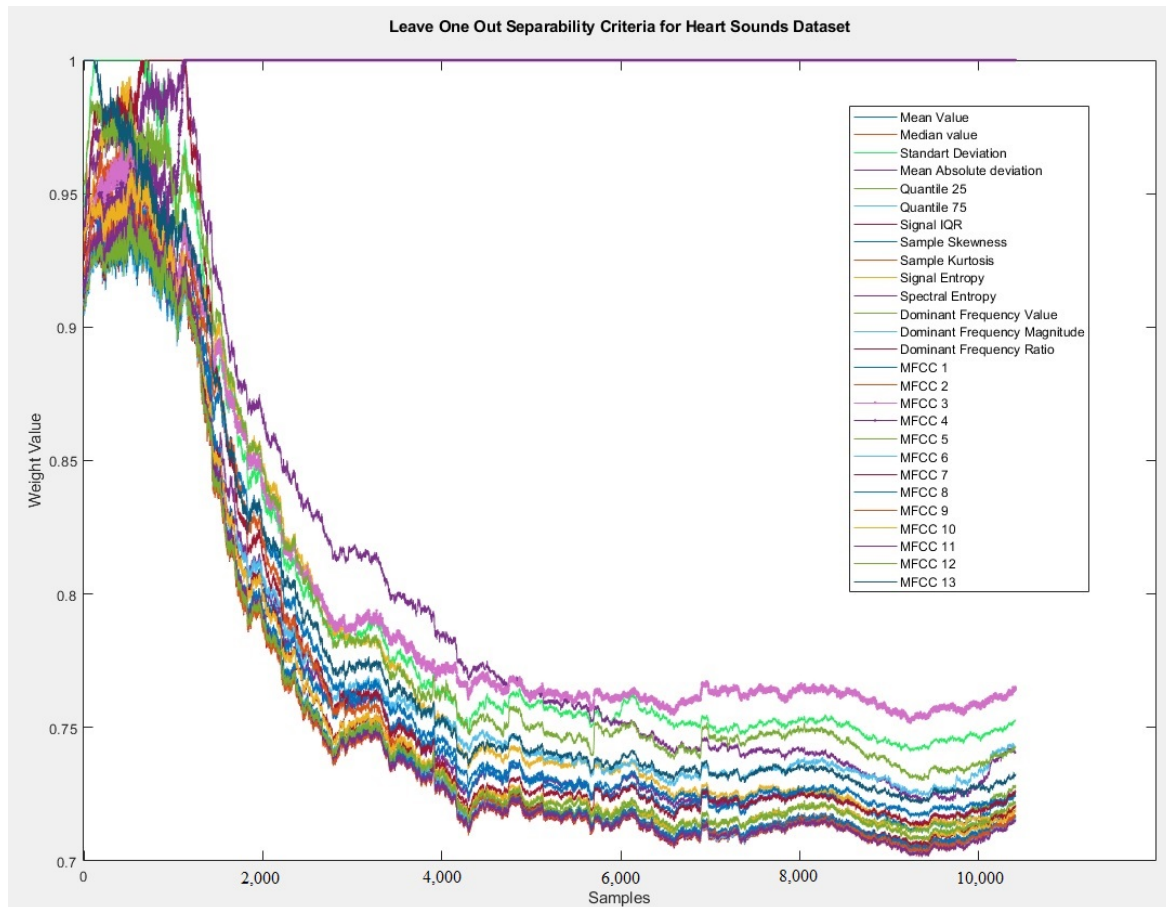


Figure 3. LOFO approach applied to the heart sounds dataset—please mind the change in the weight trends after around 1000 samples have been presented to the algorithm, where it crystallizes out that MFCC 3 and 4 are dominant features compared to all others (at the beginning all are nearly equally important).

2.2. Second Layer—Logical Neuron

The second layer of the model consists of type III fuzzy neurons (that is, those representing the data through fuzzy rules) responsible for aggregating the neurons of the first layer and their respective weights. It is composed of L fuzzy logical neurons that perform a weighted aggregation of all of the first layer neurons using fuzzy operators called t-norm and t-conorm. This aggregation is performed using the feature weights w_{il} (for $i = 1 \dots N$ and $l = 1 \dots L$) defined in the range of 0 to 1. For each input variable j , only one first layer output a_{jl} is defined as input of the l th neuron, so that \mathbf{w} is sparse, and each neuron of the second layer is associated with an input variable.

The logical neurons can be of the AND-neuron, OR-neuron, UNI-neuron, NULL-neuron or UNINULL-neuron type. These neurons are built using fuzzy operators that can use the t-norm or t-conorm to complete their calculations. One neuron differs from the other because of its operator and the ability to express fuzzy rules differently. The following table assists in understanding the terms.

S and s are t-conorm; T and t are t-norm; (g) = identity element; (u) = absorption element; U_1 is a uninorm with an identity element $= \frac{\alpha}{\beta}$; and U_2 is a uninorm with $(\frac{u-\beta}{1-\beta})$ as an identity element.

The great advantage of fuzzy logic operators is that they can sometimes use t-norms, sometimes t-conorms, allowing the neuron to perform operations like AND-neuron or OR-neuron, depending on the context. The function p (responsible for transforming the inputs and corresponding weights into individual transformed values), when applicable to neurons, performs the following steps:

- 1 Each pair (a_i, w_i, \dots) is converted into one value $b_i = p(a_i, w_i)$;
- 2 Calculate the aggregation of all values using the fuzzy operators (U, NU or N^{un}) $(b_1, b_2 \dots b_n)$, where n is the number of inputs.

Therefore, this paper's fuzzy rules to solve the problem of cardiac sounds may have connective AND or OR, depending on the type of neuron in the model structure.

2.3. Third Layer

The third layer of the model is represented by a neural aggregation network responsible for the defuzzification procedure. Therefore, it receives the weights of the second layer's rules and performs the proper calculations to obtain the desired outputs.

The output layer is constituted of one neuron (which can be recognized as a singleton) whose activation function is established by:

$$y = \text{sign} \sum_{j=0}^L f_{\omega}(\vec{Z}_{rul_j}, \vec{v}_j) \quad (21)$$

where $\vec{Z}_{rul_0} = 1$; \vec{v}_0 is the bias; \vec{Z}_{rul_j} and $\vec{v}_j, j = 1, \dots, L$ are the outputs of the fuzzy neurons of the second layer and their corresponding weights, respectively; f_{ω} is the activation function (linear); and sign represents the signal function.

The signal function can be represented by:

$$\text{sign} = \begin{cases} -1 & , \text{if } \sum_{j=0}^L f_{\omega}(\vec{Z}_{rul_j}, \vec{v}_j) < 0 \\ 1 & , \text{if } \sum_{j=0}^L f_{\omega}(\vec{Z}_{rul_j}, \vec{v}_j) > 0 \end{cases} \quad (22)$$

The signal function is necessary for the treatment of the model outputs in binary classification, as is the problem to be addressed in this paper.

2.4. Recursive Estimation of Output Weights

The defuzzification process involves turning fuzzy rules into numerical solutions to the problem. In the case of the hybrid model proposed in this paper, the definition of the weights helps in the process of interpretability of the fuzzy rules, and at the same time, helps the neural aggregation network to classify the samples of the cardiac sounds problem. Therefore, the initial weights for the artificial neurons are estimated from the initial dataset through the Moore–Penrose pseudo-inversion technique, as also used in de Campos Souza [25], for binary classification problems. This leads to the estimation of a vector \vec{v}_k per class separately through:

$$\vec{v}_k = Z_{rul}^+ \vec{y}_k \quad \forall k = 1, \dots, c \quad (23)$$

where $Z_{rul}^+ = Z_{rul}^T Z_{rul}$ is the pseudo-inverse of the Moore–Penrose matrix [32] of Z_{rul} . Z_{rul} contains the activation levels (as rows) for all selected neurons (as columns). y_k denotes the column indicator vector containing 1 s at the row positions for samples belonging to class k and 0 for all samples belonging not to class k . This leads to an indicator-based least squares estimation (regression problem on $\{0, 1\}$), where a different output weight vector \vec{v}_k (over all selected neurons) per class is achieved. That is, following the well-known one-versus-rest classification realized through regression by indicator matrix [33], which we extend here to a nonlinear version according to the integration of the activation levels in the regression matrix Z . According to the analysis in [34], the nonlinear version does not suffer from any class masking effects (as is the case for the classical linear version; see [33]).

For new incoming online stream samples, the weights are recursively updated to meet the system dynamics. Therefore, we employ the recursive weighted least squares (RWLS) approach [35], which is a

fast converging well-known estimator, as an incremental Gauss–Newton step is embedded, which can converge to the real optimum within single iterations (thus, single sample updates). We extend RWLS to an indicator-based version for each class, termed as I-RWLS, with formulas for updating the \vec{v}_k column vector for the k th class given by (with an update from sample $t - 1$ to t):

$$\eta = \vec{z}^t Q^{t-1} \left(\lambda + (\vec{z}^t)^T Q^{t-1} \vec{z}^t \right)^{-1} \quad (24)$$

$$Q^t = (I_{L^t} - \eta^T \vec{z}^t) \lambda^{-1} Q^{t-1} \quad (25)$$

$$\vec{v}_k^t = \vec{v}_k^{t-1} + \eta^T (y_k^t - \vec{z}^t \vec{v}_k^{t-1}) \quad (26)$$

where the index k again denotes the class index $k = 1, \dots, c$. \vec{z}^t denotes the regressor (row) vector of the current sample (i.e., the activation levels of all neurons in the current stream sample), J is the current kalman gain (row) vector and I_{L^t} is an identity matrix based on the number of neurons in the second layer, $L_s^t \times L_s^t$; $\lambda \in]0, 1]$ denotes a possible forgetting factor, but is to 1 per default (no forgetting). Q denotes the inverse Hessian matrix $Q = (\vec{Z}_{rul}^T \vec{Z}_{rul})^{-1}$ and is set initially as $\omega I_{L_s^t}$, where ω is a big number (e.g., 1000), such as in Rosa et al. [36]; see also Chapter 2 in [37] for a detailed convergence analysis. This matrix is directly and incrementally updated by the second equation above without requiring (time-consuming and possibly unpredictable) re-inversion of matrices. ψ is a forgetting for increased flexibility during learning (thus, being able to address concept drifts), but is set to the default value of 1 (no forgetting) in all our test experiments. The training algorithm proposed in this paper can be viewed below (Algorithm 1):

Algorithm 1 EFNN-SOF training and update algorithm.

Initial Batch Learning Phase (Input: data matrix X with K samples):

- 1: Extract L clouds in the first layer using the SOF approach (L is automatically estimated therein).
- 2: Estimate center values \vec{c} and widths $\vec{\sigma}$ for the L clouds derived from SOF.
- 3: Calculate the combination (feature) weights \vec{w} for neuron construction using Equation (20).
- 4: Construct L logic neurons on the second layer of the network by welding the L fuzzy neurons of the first layer, using logical neurons concept and weights \vec{w} .
- 5:
- 6: **for** $i = 1, \dots, K$ **do**
- 7: Calculate the regression vector $z(\vec{x}_i)$ by activation levels of all neurons in \vec{x}_i .
- 8: Store it as one row entry into the activation level matrix Z .
- 9: **end for**
- 10: Extract reduced activation level matrix Z_{rul} according to the $L_s < L$ selected neurons.
- 11: Estimate the weights \vec{v}_k of the output layer for all classes $k = 1, \dots, c$ by Equation (23) using Z_{rul} and indicator vectors \vec{y}_k .

Update Phase (Input: single data sample \vec{x}_t):

- 1: Update L_s clouds and evolving new ones on demand in the first layer using evolving SOF approach ($\rightarrow L_{s,upd}$ clouds).
 - 2: Update the feature weights \vec{w} by updating the within- and between-class scatter matrix and recalculating Equation (20).
 - 3: Perform Steps 2 and 4 with $L_{s,upd}$ clouds.
 - 4: Calculate the degree of change of all neurons (rules) by Equation (29).
 - 5: Calculate the regression vector $z(\vec{x}_t)$ by activation levels of all neurons in \vec{x}_t .
 - 6: Update the weights \vec{v}_k of the output layer by Equation (26).
-

2.5. Interpretability Through Fuzzy Rules

The interpretability of hybrid models becomes essential for artificial intelligence to stop being something incomprehensible and for its results to help companies and people to understand the reason for their results. Fuzzy neural networks work with building fuzzy rules. In this paper, they can be expressed by:

$$\begin{aligned}
& \text{Rule}_1 : \text{If } x_1 \text{ is } A_1^1 \text{ with impact } w_{11} \dots \\
& \quad \text{and/or } x_2 \text{ is } A_1^2 \text{ with impact } w_{21} \dots \\
& \quad \text{and/or } x_n \text{ is } A_1^n \text{ with impact } w_{n1} \dots \\
& \quad \text{Then } y_1 \text{ is } v_1 \\
& \text{Rule}_L : \text{If } x_1 \text{ is } A_L^1 \text{ with impact } w_{1L} \dots \\
& \quad \text{and/or } x_2 \text{ is } A_L^2 \text{ with impact } w_{2L} \dots \\
& \quad \text{and/or } x_n \text{ is } A_L^n \text{ with impact } w_{nL} \dots \\
& \quad \text{Then } y_L \text{ is } v_L
\end{aligned} \tag{27}$$

with A_i^1, \dots, A_i^n fuzzy sets represented as linguistic terms for the n inputs appearing in the i -th rules, and y_1, \dots, y_L are consequent terms, in our case classification responses (either 0 = normal heart sounds or 1 = heart murmur); thus, each rule stands for one concrete readable relationship between the input features and the presence of heart murmur; whenever a weight w_{ij} is close to 0, the corresponding antecedent part can be neglected when showing the rules to experts; this reduces the rule length and increases the transparency and readability of the rules.

In the case of using UNINULL-neuron, the fuzzy rules are represented as follows:

$$\begin{aligned}
& \text{Rule}_1 : \text{If } x_{i1} \text{ is } A_1^1 \text{ with impact } w_{11} \dots \\
& \quad \text{and/or}_{(g,u,\beta)} x_{i2} \text{ is } A_1^2 \text{ with impact } w_{21} \dots \\
& \quad \text{and/or}_{(g,u,\beta)} x_{iN} \text{ is } A_1^N \text{ with impact } w_{N1} \dots \\
& \quad \text{Then } y_1 \text{ is } v_1 \\
& \text{Rule}_2 : \text{If } x_{i1} \text{ is } A_M^1 \text{ with impact } w_{1L} \dots \\
& \quad \text{and/or}_{(g,u,\beta)} x_{i2} \text{ is } A_M^2 \text{ with impact } w_{2L} \dots \\
& \quad \text{and/or}_{(g,u,\beta)} x_{i2} \text{ is } A_M^N \text{ with impact } w_{NL} \dots \\
& \quad \text{Then } y_L \text{ is } v_L
\end{aligned} \tag{28}$$

The triple values involved in these rules allow the fuzzy neuron to act with different operators. Table 1 explains the relationship between g , u and β and the respective fuzzy operators.

Table 1. Relationship between g , u and β and the respective fuzzy operators.

g	u	β	Fuzzy Operator	Fuzzy Neuron
0	0	0	t-conorm (OR)	UNI-neuron
0	0	1	t-conorm (OR)	NULL-neuron
0	1	0	t-norm (AND)	AND-neuron
0	1	1	t-conorm (OR)	OR-neuron
1	0	0	t-conorm (OR)	OR-neuron
1	0	1	t-norm (AND)	AND-neuron
1	1	0	t-norm (AND)	NULL-neuron
1	1	1	t-norm (AND)	UNI-neuron

The complexity of these rules is linked to the number of dimensions of the problem. Therefore, any technique that can identify the relevance of a feature to a problem becomes fundamental for reducing the complexity of the rules, because when the impact (defined by the weight of the Gaussian neuron) is small, that dimension can be disregarded from the rule generated. Factors like this and updating rules through evolving training, facilitate the construction of knowledge about the problem. The antecedent connectors can be of the AND or OR type, as they can vary according to the type of neurons used in the second layer.

In the case of binary problems, the weights of the rules indicate that the rule is closer to indicating normal behavior the lower its weight for the problem, in the same way that high values of v indicate that the rule contributes enormously to the identification of heart murmur through the analysis of cardiac sounds.

Degree of Rule Variations over Time

Aside from the linguistically readable if-then structure, which can be directly extracted from the produced clouds, as described in Section 2, a different form of interpretability in data stream modeling (evolving sense) would be to track the degree of modification to the rules over time. That may be a crucial indicator within an interactive background for users/experts to inform them whether and when to re-inspect changed rules: only in the occurrence of notable variations is it worth doing so. With this insight, the degree of change of rules could be used as a structural active learning based trigger for demanding expert feedback, indicating whether the knowledge of rules has significantly improved during the adaptation of new data samples.

To meet online and especially single-pass demands, we suggest using a fast geometric-based criterion to measure the degree of change of a rule, which does not expect any past data samples. Consequently, we lean on the idea in Lughofer [38], where the intersection degree of Gaussian membership functions is used as an indicator of whether two distinctive rules have become overlapping, which can occur over data stream samples due to cluster fusion impacts. In particular, consider investigating the overlap degree of a rule i before and after its update (termed as $R_i(bef)$ and $R_i(after)$) dimension-wise and calculating an amalgamated value that can be applied as a similarity degree $S(R_i(bef), R_i(after))$. The degree of change is then given by

$$\Lambda(R_i) = 1 - S(R_i(bef), R_i(after)) \quad (29)$$

$bef = N - n$ and $after = N$ assuming that n new samples have passed the data stream-based adaptation phase with N samples used so far for model training and evolution.

The dimension-wise calculation of S is justified because the antecedents of fuzzy rules are regularly represented by a conjunction of single dimension-wise membership functions. Therefore, the approach is per the viewpoint of a fuzzy rule base: two rules are only similar if all their antecedent parts are similar. The x-coordinates of the intersection points of two Gaussians used as fuzzy sets in the same antecedent part of rule i (here for the j -th) before and after its update can be calculated by [38]:

$$\begin{aligned} inter_x(1) &= -\frac{c_{bef,j}\sigma_{after,j}^2 - c_{after,j}\sigma_{bef,j}^2}{\sigma_{bef,j}^2 - \sigma_{after,j}^2} \\ &+ \sqrt{\left(\frac{c_{bef,j}\sigma_{bef,j}^2 - c_{bef,j}\sigma_{after,j}^2}{\sigma_{after,j}^2 - \sigma_{bef,j}^2}\right)^2 - \frac{c_{bef,j}^2\sigma_{after,j}^2 - c_{after,j}^2\sigma_{bef,j}^2}{\sigma_{after,j}^2 - \sigma_{bef,j}^2}} \\ inter_x(2) &= -\frac{c_{bef,j}\sigma_{after,j}^2 - c_{after,j}\sigma_{bef,j}^2}{\sigma_{bef,j}^2 - \sigma_{after,j}^2} \\ &- \sqrt{\left(\frac{c_{bef,j}\sigma_{bef,j}^2 - c_{bef,j}\sigma_{after,j}^2}{\sigma_{after,j}^2 - \sigma_{bef,j}^2}\right)^2 - \frac{c_{bef,j}^2\sigma_{after,j}^2 - c_{after,j}^2\sigma_{bef,j}^2}{\sigma_{after,j}^2 - \sigma_{bef,j}^2}} \end{aligned} \quad (30)$$

with being $c_{bef,j}$ the j -th center coordinate of the rule before its update, and $c_{after,j}$ the j -th center coordinate of the rule after its update.

The maximal membership degree of the two Gaussian membership functions in the intersection coordinates is then used as the overlap and thus similarity degree of the corresponding rules' antecedent parts in the j -th dimension:

$$S_{bef,after}(j) = \max(\mu_i(inter_x(1)), \mu_i(inter_x(2))) \quad (31)$$

with being $\mu_i(inter_x(1))$ the membership degree of the j -th fuzzy set in rule i in the intersection point $inter_x(1)$ (which is the same before and after the update due to intersection). The amalgamation overall rule's antecedent parts leads to the final similarity degree between the rule before and after its update:

$$S(R_i(bef), R_i(after)) = \text{Agg}_{j=1}^p S_{bef,after}(j) \quad (32)$$

where Agg denotes an aggregation operator and p is the number of inputs. A feasible choice is a t -norm [39], as a strong non-overlap along one single dimension is sufficient that the clusters do not overlap at all (hence are dissimilar, thus S should become a low value); this is because one dimension can already be sufficient for tearing the rules away in a geometric sense; see also [38].

3. Materials and Methods

The dataset used in this study came from a set of public data provided by Liu et al. [4]. It is the combination of three datasets. The University of Michigan Health System provided the first dataset with 23 recordings of heart sounds with a total duration of 1496.8 s. The second dataset that makes up the data used in this research is also published, comprising 176 recordings for segmentation of cardiac sounds and 656 recordings for cardiac sound classification lasting from 1 to 30 s and having a frequency range of 195 Hz. Finally, 64 final recordings were from the non-open database of cardiac auscultation of heart murmurs, provided by eGeneral Medical Inc. All three bases were combined and were the reference of studies by Liu et al. [4]. At the PhysioNet/CinC Challenge 2016, an extensive collection of heart sound recordings was added to the dataset. These data came from different clinical and non-clinical environments in the real world. The data include not only clear heart sounds but also very loud recordings, providing authenticity to the challenge. The data were collected from normal individuals and people with some cardiac problems—mostly children and adults—from different locations on the patients' bodies, with a predominance of the aortic area, pulmonary area, tricuspid area and mitral area [4]. These data were collected to construct the dataset through the union of large databases that used different equipment to collect them. For data collected by the Massachusetts Institute of Technology (MIT), an electrocardiogram (ECG) with an electronic stethoscope Welch Allyn Meditron (Skaneateles Falls, New York, NY, USA) was used, being capable of a frequency response between 20 and 20 kHz. The Aalborg University database used a Littmann E4000 electronic stethoscope (3M, Maplewood, Minnesota) with the stethoscope's frequency response capacity ranging between 20 and 1000 Hz. The Aristotle University of Thessaloniki used the audio scope (electronic stethoscope) in order to record amplified and unfiltered signals. In the KN Toosi University of Technology database, an electronic stethoscope (3M Littmann's 3200) in conjunction with echocardiography was used to collect the data. In the work carried out by the University of Haute Alsace, the stethoscopes provided by Infrac Corporation (Strasbourg, France) were used. Finally, Dalian University of Technology and Shiraz University used electronic stethoscopes from different companies.

Table 2 summarizes the main dimensions, average values and their respective standard deviations, in addition to maximum and minimum values for each feature.

Figure 4 shows the organization of the dataset features evaluated in this paper across all samples. In this figure, it is possible to identify the complexity of the data.

The procedures to be performed with this type of dataset involve tasks such as recording sounds in selected patients, pre-processing the collected signals and segmenting and classifying the collected data so that medical procedures can be performed in cases of any cardiac problem detected. In the pre-processing stage of the signal, it is interesting to highlight the selection of the best characteristics in the collected signals and the application of filters to detect baseline changes and high-frequency noise. Thus, the data will be ready to be delineated as to the beginning and end of a heartbeat in different elements (S1, S2, systolic and diastolic). This identification occurs in the segmentation phase so that in the end, in the classification phase, the patterns of cardiac signals are determined and

compared with signals from cardiac pathologies. In this case, intelligent systems based on artificial intelligence can continuously act to classify these heart-related disease patterns. After defining an abnormal pattern or that corresponds to known heart disease, medical procedures and treatments can be performed [4]. The use of intelligent models facilitates and automates the decision-making process. Therefore, it is essential to expand the studies related to heart murmurs' identification by approaches based on artificial intelligence.

Table 2. Input features of the cardiac sounds dataset [4].

Dataset Dimensions	Mean	Standard Deviation	Max	Min
Mean Value	−0.0131	0.0396	0.1696	−0.4726
Median Value	−0.0134	0.0399	0.2364	−0.4744
Standard Deviation	0.0824	0.0659	0.6126	0.0021
Mean Absolute Deviation	0.0484	0.0510	0.5283	0.0009
Quantile 25	−0.0413	0.0657	0.1212	−0.5731
Quantile 75	0.0144	0.0519	0.5727	−0.3869
Signal IQR	0.0557	0.0875	1.0888	0
Sample Skewness	−0.1052	0.8711	10.3526	−6.2164
Sample Kurtosis	17.3596	15.8936	754.0798	1.5330
Signal Entropy	−1.8492	0.7922	0.6655	−6.8675
spectral entropy	0.2589	0.1994	0.7812	−0.3333
dominant frequency value	26.8757	26.0346	254.0303	0
Dominant Frequency Magnitude	0.1472	0.2091	1.0000	0.0102
Dominant Frequency Ratio	0.3399	0.2023	1.0000	0.0184
MFCC 1	96.8203	5.4674	118.0333	77.2747
MFCC 2	6.9214	4.3758	17.7092	−15.3445
MFCC 3	1.3577	3.6801	13.9213	−15.4105
MFCC 4	−2.0712	3.8150	14.8043	−14.6614
MFCC 5	−2.0186	3.4275	15.6168	−17.0746
MFCC 6	−2.1773	3.1318	13.6594	−14.6325
MFCC 7	−1.9079	2.7580	14.5581	−16.6508
MFCC 8	−1.7505	2.5032	17.4436	−14.6091
MFCC 9	−1.4809	2.2581	9.3733	−11.8959
MFCC 10	−1.3294	2.2276	10.3163	−14.1284
MFCC 11	−1.0183	1.9913	8.2289	−10.8879
MFCC 12	−1.0138	1.8840	12.0782	−16.5322
MFCC 13	−1.0221	1.6583	6.6063	−10.5979
Class	0.7574	0.4287	1.0000	0

The results obtained by related works depend on techniques for feature selections or complementary intelligent approaches to obtain these results. In this paper, all dimensions of the problem will be used to obtain interpretable assessments on the heart murmur problem. Another relevant factor in the results obtained by RNN and HMM can be explained due to benefiting from its remarkable ability to model sequential patterns because the data in this article are essentially sequences. It is worth mentioning that the approaches also worked offline or with a lower percentage of training or testing, diverging from the evolving approaches proposed in the experiments.

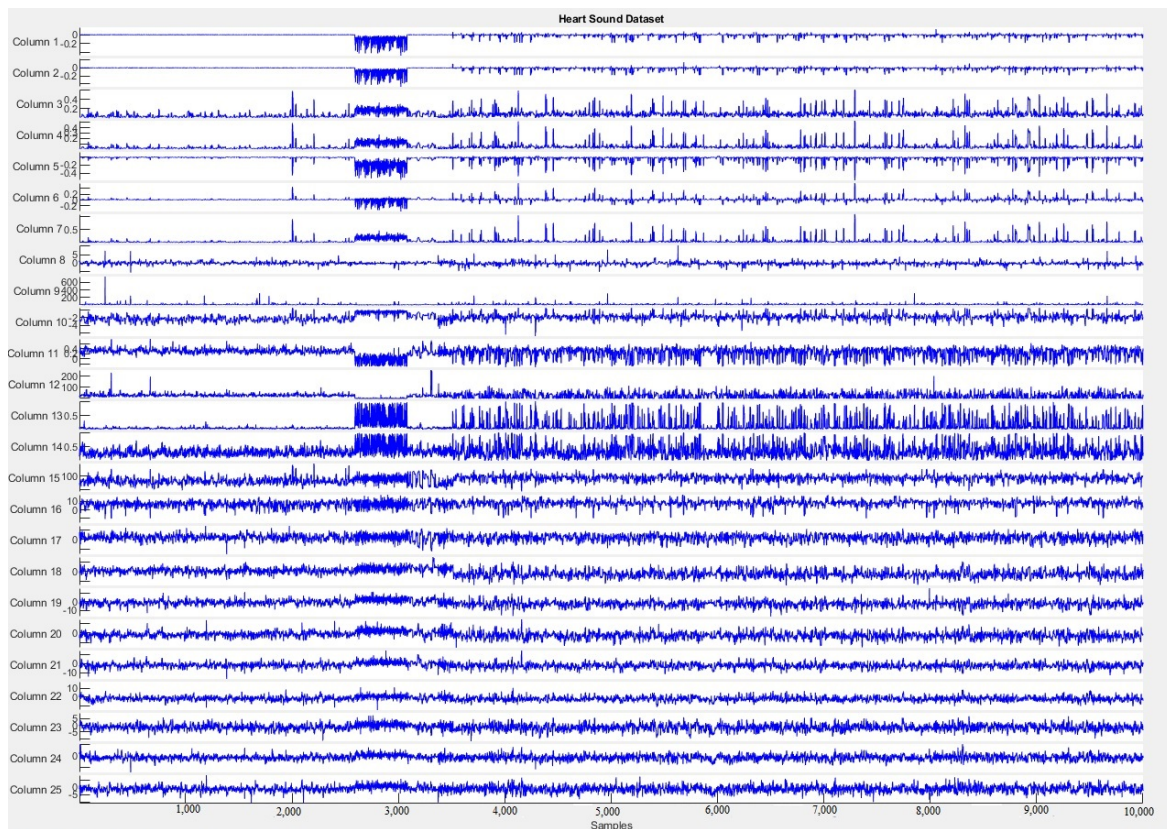


Figure 4. Heart sounds dataset features over time.

4. Heart Sounds Classification Test

This section presents the tests to be performed on the dataset—the models' configurations, the hyperparameters, forms of evaluation and the tests' results. The tests were configured based on a specific proportion of samples for offline training (10%), and the rest were used for online adaptive training to prove the efficiency of the model's evolving approach. The tests were compared with the models proposed in this paper, where each model constructed a different type of neuron according to the variants shown in Table 3. Therefore, it is identified as “evolving-SOF-type” in the legend of the figures and in the text. For example, a model that uses AND-neurons is represented in the graphics as evolving-SOF-AND. These models were compared with a renowned state-of-the-art technique, termed ALMMo-0* [10], for the online detection of cardiac sounds. The parameters used by the ALMMo-0* model were the same as those suggested and successfully used in Soares et al. [10]. To help compare the results of the online models; the model in its offline version of the fuzzification process was also used in the tests.

Simulations were performed on a Core (TM) 2 Duo CPU, 2.27 GHz, with 3 GB RAM. Time is represented by the sum of training time and test (seconds) in each of the models. Neurons represent the most representative neurons after the pruning or regularization of the models.

Table 3. Variants of logical neurons and their construction.

Neuron	Neuron Representation	Fuzzy Logic Operator	Relevancy Transformation	Reference
AND	$\vec{Z}_{rul} = AND(\mathbf{w};\mathbf{a}) = T_{i=1}^n(w_i \text{ s } a_i)$	T (product)	-	[14]
OR	$\vec{Z}_{rul} = OR(\mathbf{w};\mathbf{a}) = S_{i=1}^n(w_i \text{ t } a_i)$	S (probabilistic sum)	-	[14]
UNI	$\vec{Z}_{rul} = UNI(\mathbf{w};\mathbf{a}) = U_{i=1}^n p(w_i, a_i)$	$U(x, y, g) = \begin{cases} g T(\frac{x}{g}, \frac{y}{g}), \text{ if } y \in [0, g] \\ g + (1 - g) S(\frac{x-g}{1-g}, \frac{y-g}{1-g}), \text{ if } y \in (g, 1] \end{cases}$	$p(w, a; g) = wa + \bar{w}g,$	[40]
NULL	$\vec{Z}_{rul} = NULL(\mathbf{w};\mathbf{a}) = N_{i=1}^n p(w_i, a_i)$	$NU(x, y, u) = \begin{cases} uS(\frac{x}{u}, \frac{y}{u}), \text{ if } y \in [0, u] \\ u + (1 - u) T(\frac{x-u}{y-u}, \frac{1-y}{1-u}), \text{ if } y \in [u, 1] \end{cases}$	$p(w, a, u) = wa + \bar{w}u$	[41]
UNINULL	$\vec{Z}_{rul} = UNINULL(\mathbf{w}, \mathbf{a}, \beta, g, u) = N_{i=1}^{un} p(w_i, a_i)$	$N^{un}(x, y, \beta, g, u) = \begin{cases} \beta U_1(\frac{x}{\beta}, \frac{y}{\beta}), \text{ if } x, y \in [0, \beta] \\ \beta + (1 - \beta) U_2(\frac{x-\beta}{1-\beta}, \frac{y-\beta}{1-\beta}), \text{ if } x, y \in [\beta, 1] \end{cases}$	$p(w, a, \beta, g, u) = \begin{cases} wa + \bar{w}\frac{g}{\beta}, \text{ if } U_1 \\ wa + \bar{w}\frac{u-\beta}{1-\beta}, \text{ if } U_2 \end{cases}$	[42]

4.1. Evaluation Criteria

The factors evaluated in this paper are based on the accuracy, calculated in batch offline mode as:

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \quad (33)$$

where TP = true positive, TN = true negative, FN = false negative and FP = false positive. For the online mode we exploit the usage of accuracy trend lines over time in order to see the evolution of the accuracy, as a result of the evolution of the models. Such an accuracy trend line can be achieved by the interleaved predict and then train evaluation scenario [43]. In this scenario, (i) a prediction is made on a new incoming stream sample; (ii) the prediction \hat{y} is compared with the real class label y (always included in our evaluation dataset); (iii) the accuracy is updated according to the correctness of the prediction by:

$$Accuracy(K + 1) = \frac{Accuracy(K) * K + I_{\hat{y}=y}}{K + 1}, \quad (34)$$

where I denotes the indicator function and is 1 when the prediction is correct, i.e., $\hat{y} = y$; 0 otherwise ($Accuracy(0) = 0$); and (iv) afterwards the model is updated. This means that the update of the accuracy in Equation (34) results in an accumulated one-step-ahead prediction accuracy, which can be stored in a vector for each K leading to an accuracy trend line—it is a commonly used measure for timely-tracked model accuracy in the stream mining community [44].

4.2. Results

In Figure 5, the accumulated accuracy over time is plotted in one graph for all methods (construction variants and SoA ALMMo-0*)—the closer the trend to 100% on the y-axis, the better the model performance. It is possible to notice that the models had different performances. It should be noted that the models with neurons of the UNI-neuron, NULL-neuron and UNINULL-neuron type obtained the best results of the experiment with similar behavior to each other. All three models have surpassed the state-of-the-art for resolving the identification of cardiac sounds in online training. It is important to note that models that used only AND-neurons and OR-neurons (the classical construction in fuzzy neural network approaches) were the worst performers, nor could they outperform the related approach ALMMo-0*. This factor can be explained due to the complex nature of the problem, which requires variations in the neuron constructions (allowing a combination of ANDs and ORs in rules) to efficiently solve the problem. The three models with such an advanced construction turned out to be the best performers, even being able to outperform ALMMo-0*, and this more or less over the whole timeline of all stream samples (thus, in each point of time showing a superior overall performance).

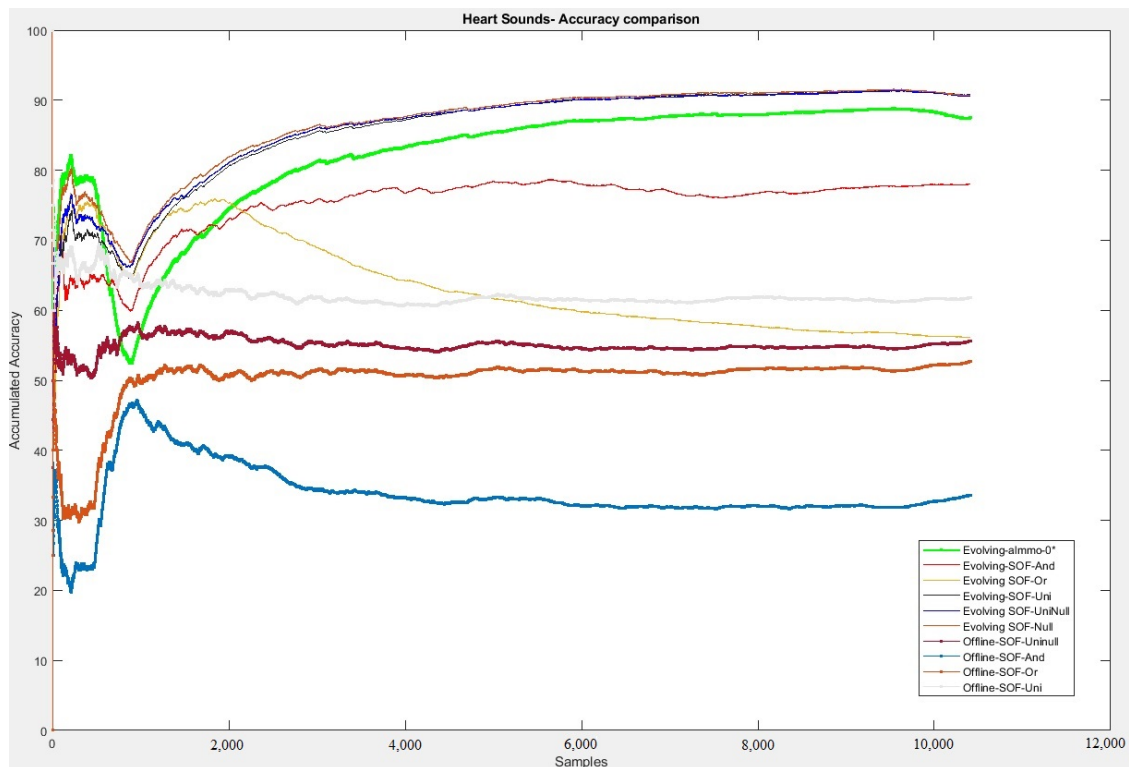


Figure 5. Heart sounds dataset—accumulated accuracy over the whole data stream.

4.3. Further Discussions

When comparing the results with those shown in Table 4, the best neuron construction variant of the model in this paper showed a better (final) accuracy than those reported in Potes et al., Zahibi et al., Whitaker et al., Li et al. and Noman et al., even though these were trained on the full dataset (comprising around 10.3 K samples) in a batch, offline training mode, thereby having the ability to “see” all the data and iterate over the whole dataset multiple times, which is convenient in neural network-based learning schemes (such as stochastic gradient descent approaches and others). This is remarkable performance on the part of our method, which always uses only one sample for model updating, thereby having a much narrower view on the data. Another factor to be highlighted is the difference between the online and offline versions in solving the problem (the offline versions marked by the last four lines in the legend of Figure 5). Offline models use the same fuzzy rules with the same parameters as initially learned to classify new patterns. Thus, they are hardly able to improve the accuracy over time as the evolving methods significantly do (up to more than 90% accuracy) due to the integration of new data in their parameter and structure updates.

It should be noted that the superior accuracy results obtained by those obtained in Dominguez-Morales et al. were also obtained with an offline model. In Latif et al., Zeng et al. and Aziz et al., the tests were conducted by explicit feature selection techniques before the final results. Therefore, not all the features of the problem were used and a kind of curse of dimensionality reduction could be achieved, but this was done in connection with external expert knowledge (which requires manual effort and may be not necessarily available). In the tests by Shukla et al., the accuracy results were obtained only with a subset of 3669 samples selected by different criteria. Chen et al.’s work also obtained better numerical results through the use of a modified frequency wavelet transform as a pre-processing step on the feature space. It should be noted that all these results from related works which were numerically superior to our model were not produced in an evolving way, but fully in an offline, batch training phase seeing the whole set at once. Such models, however, cannot be applied for a larger (possibly infinite) online training process, where stream samples come in one by one (and are not collected and stored as a full dataset in advance). Another critical factor is the lack of knowledge

generation by state-of-the-art models on the theme of cardiac sounds. The authors emphasized the techniques and results obtained, but did not demonstrate the final relationship between the problem's features and the output classes (heart murmur or no hear murmur), as their models appeared to be black boxes without any show of interpretable components (such as rules).

Table 4. Related approaches applied to the cardiac sound dataset.

Author	Model	Accuracy/Macc	Reference
Potes et al.	CNN	0.8602	[45]
Zahibi et al.	SVM	0.8509	[46]
Dominguez-Morales et al.	DNN.	0.9416	[47]
Latif et al.	RNN.	0.9861	[48]
Whitaker et al.	SVM	0.8926	[49]
Xiao et al.	CNN	0.9300	[50]
Deng et al.	MFCC-CRNN	0.9701	[51]
Shukla et al.	HMM	0.9807	[52]
Soares et al.	ALMMo-0 *	0.9304	[10]
Zeng et al.	TQWT-VMD-NN	0.9789	[53]
Chen et al.	CNN	0.9391	[54]
Aziz et al.	SVM	0.9524	[55]
Chowdhury et al.	DNN	0.9710	[56]
Li et al.	TWSVM	0.9040	[57]
Noman et al.	MSAR	0.8610	[58]

The fuzzy rules of the problem had as a starting parameter the total of four rules. After the evolving, adaptive training phase was entirely performed on all the stream samples, 49 fuzzy rules turned out to be generated to address the problem. That means that the model autonomously learned from the new samples received and created new logical interactions to solve the problem, which in turn led to a rising accumulated accuracy trend line over time, as shown in Figure 5. These characteristics are very similar to human behavior when learning new things and are then able to make better estimations/predictions when new situations arise. Regarding the initial fuzzy rules, a set of four partitions for each feature was used to represent knowledge (Table 5). Regarding each feature's impact, it is possible to highlight that all features initially had high impact values in the fuzzy rules, where the most significant emphasis goes to MFCC 13. After the evolution training, the LOO technique adequately identified the weight of features for the problem, where the most relevant for identifying cardiac sounds were MFCC 3 and 4—as shown in Figure 3.

Table 5. Interpretability with respect to the (degree of) changes in fuzzy neurons during the evolution phase with new incoming data samples.

Rule 1 did change in 27 membership functions and by a degree of 0.7896 with consequent change from Normal Heart Sound to Heart Murmur.
Rule 2 did change in 27 membership functions and by a degree of 1.0000 with consequent change from Heart Murmur to Normal Heart Sound.
Rule 3 did change in 27 membership functions and by a degree of 1.0000 with no consequent change.
Rule 4 did change in 27 membership functions and by a degree of 1.0000 with no consequent change.

During the offline training step, four rules were created. For the interpretability criteria, the following conversions were considered: the first membership function is very small; the second membership function represents small; the third membership function represents medium; and finally, the fourth membership function is presented as high. Thus, the offline rules can be described and interpreted as follows:

Rule 1—If (mean value is small) with impact 0.9179 or (median value is small) with impact 0.9332 or (standard deviation is very small) with impact 0.9466 or (mean absolute deviation is small) with

impact 0.9401 or (quantile 25 is small) with impact 0.9066 or (quantile 75 is small) with impact 0.9066 or (signal iqr is small) with impact 0.9066 or (sample skewness is very small) with impact 0.9064 or (sample Kurtosis is small) with impact 0.9290 or (signal entropy is very small) with impact 0.9078 or (spectral entropy is small) with impact 0.9372 or (dominant frequency value is small) with impact 0.9076 or (dominant frequency magnitude is small) with impact 0.9076 or (dominant frequency ratio is small) with impact 0.9261 or (MFCC 1 is small) with impact 0.9164 or (MFCC 2 is small) with impact 0.9071 or (MFCC 3 is small) with impact 0.9088 or (MFCC 4 is small) with impact 0.9159 or (MFCC 5 is small) with impact 0.9534 or (MFCC 6 is small) with impact 0.9068 or (MFCC 7 is small) with impact 0.9083 or (MFCC 8 is small) with impact 0.9101 or (MFCC 9 is small) with impact 0.9226 or (MFCC 10 is small) with impact 0.9361 or (MFCC 11 is small) with impact 0.9119 or (MFCC 12 is small) with impact 0.9085 or (MFCC 13 is small) with impact 1.0000 then (sound identification is normal heart sound) with certainty 0.2350.

Rule 2—If (mean value is medium) with impact 0.9179 or (median value is high) with impact 0.9332 or (standard deviation is high) with impact 0.9466 or (mean absolute deviation is high) with impact 0.9401 or (quantile 25 is high) with impact 0.9066 or (quantile 75 is high) with impact 0.9066 or (signal IQR is high) with impact 0.9066 or (sample skewness is high) with impact 0.9064 or (sample Kurtosis is high) with impact 0.9290 or (signal entropy is high) with impact 0.9078 or (spectral entropy is high) with impact 0.9372 or (dominant frequency value is high) with impact 0.9076 or (dominant frequency magnitude is high) with impact 0.9076 or (dominant frequency ratio is high) with impact 0.9261 or (MFCC 1 is high) with impact 0.9164 or (MFCC 2 is high) with impact 0.9071 or (MFCC 3 is very small) with impact 0.9088 or (MFCC 4 is medium) with impact 0.9159 or (MFCC 5 is high) with impact 0.9534 or (MFCC 6 is high) with impact 0.9068 or (MFCC 7 is high) with impact 0.9083 or (MFCC 8 is high) with impact 0.9101 or (MFCC 9 is high) with impact 0.9226 or (MFCC 10 is high) with impact 0.9361 or (MFCC 11 is high) with impact 0.9119 or (MFCC 12 is high) with impact 0.9085 or (MFCC 13 is high) with impact 1.0000 then (sound identification is heart murmur) with certainty -0.2782 .

Rule 3—If (mean value is high) with impact 0.9179 and (median value is medium) with impact 0.9332 and (standard deviation is medium) with impact 0.9466 and (mean absolute deviation is medium) with impact 0.9401 and (quantile 25 is medium) with impact 0.9066 and (quantile 75 is medium) with impact 0.9066 and (signal IQR is medium) with impact 0.9066 and (sample skewness is small) with impact 0.9064 and (sample Kurtosis is medium) with impact 0.9290 and (signal entropy is medium) with impact 0.9078 and (spectral entropy is medium) with impact 0.9372 and (dominant frequency value is medium) with impact 0.9076 and (dominant frequency magnitude is medium) with impact 0.9076 and (dominant frequency ratio is medium) with impact 0.9261 and (MFCC 1 is small) with impact 0.9164 and (MFCC 2 is medium) with impact 0.9071 and (MFCC 3 is small) with impact 0.9088 and (MFCC 4 is high) with impact 0.9159 and (MFCC 5 is medium) with impact 0.9534 and (MFCC 6 is medium) with impact 0.9068 and (MFCC 7 is medium) with impact 0.9083 and (MFCC 8 is medium) with impact 0.9101 and (MFCC 9 is medium) with impact 0.9226 and (MFCC 10 is medium) with impact 0.9361 and (MFCC 11 is medium) with impact 0.9119 and (MFCC 12 is medium) with impact 0.9085 and (MFCC 13 is medium) with impact 1.0000 then (sound identification is normal heart sound) with certainty 0.3899.

Rule 4—If (mean value is very small) with impact 0.9179 or (median value is very small) with impact 0.9332 or (standard deviation is small) with impact 0.9466 or (mean absolute deviation is very small) with impact 0.9401 or (quantile 25 is very small) with impact 0.9066 or (quantile 75 is very small) with impact 0.9066 or (signal IQR is very small) with impact 0.9066 or (sample skewness is medium) with impact 0.9064 or (sample Kurtosis is very small) with impact 0.9290 or (signal entropy is small) with impact 0.9078 or (spectral entropy is very small) with impact 0.9372 or (dominant frequency value is very small) with impact 0.9076 or (dominant frequency magnitude is very small) with impact 0.9076 or (dominant frequency ratio is very small) with impact 0.9261 or (MFCC 1 is medium) with impact 0.9164 or (MFCC 2 is very small) with impact 0.9071 or (MFCC 3 is small) with impact 0.9088 or

(MFCC 4 is very small) with impact 0.9159 or (MFCC 5 is very small) with impact 0.9534 or (MFCC 6 is very small) with impact 0.9068 or (MFCC 7 is very small) with impact 0.9083 or (MFCC 8 is very small) with impact 0.9101 or (MFCC 9 is very small) with impact 0.9226 or (MFCC 10 is very small) with impact 0.9361 or (MFCC 11 is very small) with impact 0.9119 or (MFCC 12 is very small) with impact 0.9085 or (MFCC 13 is very small) with impact 1.0000 then (sound identification is normal heart sound) with certainty 0.6577.

After evolution, 49 fuzzy rules were created, and their pertinence functions showed similar behaviors for the interpretation criteria. The following four rules represent the initial rules (as shown above) after their update with all stream samples. The rules' consequent values are defined in the range between -1 and 1 , where the first value indicates a strong probability of a heart murmur, and the value 1 represents a strong tendency to healthy heart sounds:

Rule 1—If (mean value is high) with impact 0.7192 or (median value is medium) with impact 0.7152 or (standard deviation is very small) with impact 0.7527 or (mean absolute deviation is very small) with impact 0.7257 or (quantile 25 is medium) with impact 0.7153 or (quantile 75 is small) with impact 0.7153 or (signal IQR is very small) with impact 0.7153 or (sample skewness is medium) with impact 0.7248 or (sample Kurtosis is small) with impact 0.7167 or (signal entropy is very small) with impact 0.7210 or (spectral entropy is small) with impact 0.7410 or (dominant frequency value is small) with impact 0.7283 or (dominant frequency magnitude is medium) with impact 0.7430 or (dominant frequency ratio is small) with impact 0.7210 or (MFCC 1 is small) with impact 0.7255 or (MFCC 2 is small) with impact 0.7170 or (MFCC 3 is small) with impact 0.7650 or (MFCC 4 is very small) with impact 1.0000 or (MFCC 5 is small) with impact 0.7429 or (MFCC 6 is very small) with impact 0.7157 or (MFCC 7 is small) with impact 0.7257 or (MFCC 8 is small) with impact 0.7170 or (MFCC 9 is small) with impact 0.7164 or (MFCC 10 is small) with impact 0.7196 or (MFCC 11 is small) with impact 0.7155 or (MFCC 12 is small) with impact 0.7218 or (MFCC 13 is very small) with impact 0.7326 then (sound identification is heart murmur) with certainty -1 .

Rule 2—If (mean value is high) with impact 0.7192 or (median value is very high) with impact 0.7152 or (standard deviation is medium) with impact 0.7527 or (mean absolute deviation is medium) with impact 0.7257 or (quantile 25 is medium) with impact 0.7153 or (quantile 75 is high) with impact 0.7153 or (signal IQR is small) with impact 0.7153 or (sample skewness is medium) with impact 0.7248 or (sample Kurtosis is medium) with impact 0.7167 or (signal entropy is high) with impact 0.7210 or (spectral entropy is high) with impact 0.7410 or (dominant frequency value is very high) with impact 0.7283 or (dominant frequency magnitude is high) with impact 0.7430 or (dominant frequency ratio is very high) with impact 0.7210 or (MFCC 1 is high) with impact 0.7255 or (MFCC 2 is medium) with impact 0.7170 or (MFCC 3 is medium) with impact 0.7650 or (MFCC 4 is medium) with impact 1.0000 or (MFCC 5 is high) with impact 0.7429 or (MFCC 6 is high) with impact 0.7157 or (MFCC 7 is high) with impact 0.7257 or (MFCC 8 is high) with impact 0.7170 or (MFCC 9 is high) with impact 0.7164 or (MFCC 10 is very high) with impact 0.7196 or (MFCC 11 is very high) with impact 0.7155 or (MFCC 12 is high) with impact 0.7218 or (MFCC 13 is small) with impact 0.7326 then (sound identification is normal heart sound) with certainty 0.7191.

Rule 3—If (mean value is small) with impact 0.7192 and (median value is small) with impact 0.7152 and (standard deviation is high) with impact 0.7527 and (mean absolute deviation is high) with impact 0.7257 and (quantile 25 is very small) with impact 0.7153 and (quantile 75 is high) with impact 0.7153 and (signal IQR is high) with impact 0.7153 and (sample skewness is medium) with impact 0.7248 and (sample Kurtosis is medium) with impact 0.7167 and (signal entropy is high) with impact 0.7210 and (spectral entropy is medium) with impact 0.7410 and (dominant frequency value is medium) with impact 0.7283 and (dominant frequency magnitude is medium) with impact 0.7430 and (dominant frequency ratio is medium) with impact 0.7210 and (MFCC 1 is small) with impact 0.7255 and (MFCC 2 is medium) with impact 0.7170 and (MFCC 3 is small) with impact 0.7650 and (MFCC 4 is very small) with impact 1.0000 and (MFCC 5 is small) with impact 0.7429 and (MFCC 6 is small) with impact 0.7157 and (MFCC 7 is small) with impact 0.7257 and (MFCC 8 is small) with impact 0.7170

and (MFCC 9 is small) with impact 0.7164 and (MFCC 10 is small) with impact 0.7196 and (MFCC 11 is small) with impact 0.7155 and (MFCC 12 is medium) with impact 0.7218 with impact 0.7326 and (MFCC 13 is medium) then (sound identification is normal heart sound) with certainty 0.1631.

Rule 4—If (mean value is very small) with impact 0.7192 or (median value is very small) with impact 0.7152 or (standard deviation is small) with impact 0.7527 or (mean absolute deviation is medium) with impact 0.7257 or (quantile 25 is very small) with impact 0.7153 or (quantile 75 is small) with impact 0.7153 or (signal IQR is high) with impact 0.7153 or (sample skewness is small) with impact 0.7248 or (sample Kurtosis is medium) with impact 0.7167 or (signal entropy is medium) with impact 0.7210 or (spectral entropy is very small) with impact 0.7410 or (dominant frequency value is small) with impact 0.7283 or (dominant frequency magnitude is small) with impact 0.7430 or (dominant frequency ratio is very small) with impact 0.7210 or (MFCC 1 is small) with impact 0.7255 or (MFCC 2 is small) with impact 0.7170 or (MFCC 3 is small) with impact 0.7650 or (MFCC 4 is very small) with impact 1.0000 or (MFCC 5 is very small) with impact 0.7429 or (MFCC 6 is very small) with impact 0.7157 or (MFCC 7 is very small) with impact 0.7257 or (MFCC 8 is very small) with impact 0.7170 or (MFCC 9 is small) with impact 0.7164 or (MFCC 10 is small) with impact 0.7196 or (MFCC 11 is small) with impact 0.7155 or (MFCC 12 is small) with impact 0.7218 or (MFCC 13 is medium) then with impact 0.7326 (sound identification is normal heart sound) with certainty 2.7535.

5. Conclusions

The evolving fuzzy neural network model proposed in this paper obtained better results than state-of-the-art for online training. The generated fuzzy rules represent the model's transparency in solving complex problems, and at the same time, determining logical and linguistic relationships to facilitate feedback regarding how the model arrived at the final results. This approach proved to be promising because it managed to deal with a complex problem in an organized and coherent way, extracting knowledge about the data submitted to the model. A positive point to be observed was the powerful performance of models based on online training, surpassing by three percent the accuracy of the state-of-the-art model under the same established test conditions. As a weakness of this model, a long time was observed in defining the weights of neurons in the first layer, mainly in training in evolution. A non-parametric model's advantage brings the complexity of this simple model to solve the problem, requiring no additional procedures to solve the problem. Only the choice of fuzzy neuron defines the element to be used in solving the problem. It should be noted that some models, especially those with greater flexibility in the use of t-norm and t-conorm operators, performed better concerning accuracy. The hybrid model proposed in this paper has good prospects for working with new cases of the same nature, allowing new cases to be analyzed and with a 90% possibility of identifying problems related to heart murmurs. The extracted fuzzy rules can help professionals quickly identify patterns and adopt specific treatments to attack existing patterns in the patient's diagnosis.

Future work related to this paper can be related to working with different datasets on heart murmurs, and tests with other models. It is also advisable to create other evolving fuzzy neural network architectures with different fuzzification approaches, defuzzification and training. Another future approach to the model would be to work on the prediction of future treatments such as those addressed in the work of Jeong et al. [59], which addresses blood pressure problems.

Forthcoming efforts may also be carried out as an expansion of this work acting dynamically in evaluating other datasets available in the literature, appraising other features, types of heart problems caused by a heart murmur or other types of diseases that can be collected through the same procedures that originated the dataset covered in this paper. An advantage of the proposed model is that it can act dynamically with numerical data of any problem, such as in the medical area, generating fuzzy rules adapted as new information is collected. The (linguistically readable) fuzzy rules generated from this use of the model can support and facilitate routine treatments, early identification of problems and other comorbidities commonly treated in hospitals and addressed in cutting-edge research.

Author Contributions: Conceptualization, P.V.d.C.S. and E.L.; methodology, P.V.d.C.S.; software, P.V.d.C.S.; validation, E.L.; formal analysis, E.L.; investigation, P.V.d.C.S.; resources, P.V.d.C.S.; data curation, E.L.; writing—original draft preparation, P.V.d.C.S.; writing—review and editing, E.L.; visualization, P.V.d.C.S.; supervision, E.L.; project administration, E.L.; funding acquisition, E.L. All authors have read and agreed to the published version of the manuscript.

Funding: Open Access Funding by the Austrian Science Fund (FWF).

Acknowledgments: The authors acknowledge the Austrian Science Fund (FWF): contract number P32272-N38, acronym IL-EFS.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional neural network
SVM	Support Vector Machine
RNN	Recurrent Neural Networks
DRNN	Deep Recurrent Neural Network
MFCC	Mel-Frequency Cepstrum Coefficient
CRNN	Deep Convolutional and Recurrent Deural Network
DNN	Deep Neural Network
HMM	Hidden Markov model
ALMMo-0*	Zero-Order Autonomous Learning Multiple-Model
TQWT	Tunable Q-factor Wavelet Transform
VMD	Variational Mode Decomposition
NN	Neural Network
TWSVM	Twin Support Vector Machine
MSAR	Markov-Switching Autoregressive
LOFO	Leave-One-Feature-Out

References

1. Ulbricht, T.; Southgate, D. Coronary heart disease: Seven dietary factors. *Lancet* **1991**, *338*, 985–992. [[CrossRef](#)]
2. Hsieh, N.C.; Hung, L.P.; Shih, C.C.; Keh, H.C.; Chan, C.H. Intelligent postoperative morbidity prediction of heart disease using artificial intelligence techniques. *J. Med. Syst.* **2012**, *36*, 1809–1820. [[CrossRef](#)] [[PubMed](#)]
3. Sun, R.; Liu, M.; Lu, L.; Zheng, Y.; Zhang, P. Congenital heart disease: Causes, diagnosis, symptoms, and treatments. *Cell Biochem. Biophys.* **2015**, *72*, 857–860. [[CrossRef](#)] [[PubMed](#)]
4. Liu, C.; Springer, D.; Li, Q.; Moody, B.; Juan, R.A.; Chorro, F.J.; Castells, F.; Roig, J.M.; Silva, I.; Johnson, A.E.; et al. An open access database for the evaluation of heart sound algorithms. *Physiol. Meas.* **2016**, *37*, 2181. [[CrossRef](#)]
5. Rud, S.; Yang, J.S. A support vector machine (SVM) classification approach to heart murmur detection. In *International Symposium on Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 52–59.
6. Quiceno-Manrique, A.; Godino-Llorente, J.; Blanco-Velasco, M.; Castellanos-Dominguez, G. Selection of dynamic features based on time–frequency representations for heart murmur detection from phonocardiographic signals. *Ann. Biomed. Eng.* **2010**, *38*, 118–137. [[CrossRef](#)]
7. Mukherjee, A.; Pathak, N.; Roy, A. Heart murmur detection using fractal analysis of phonocardiograph signals. *Int. J. Comput. Appl.* **2014**, *88*, 30–35. [[CrossRef](#)]
8. Kumar, D.; Carvalho, P.D.; Antunes, M.; Henriques, J.; Maldonado, M.; Schmidt, R.; Habetha, J. Wavelet transform and simplicity based heart murmur segmentation. In *Proceedings of the 2006 Computers in Cardiology, Valencia, Spain, 17–20 September 2006*; pp. 173–176.
9. Avendano-Valencia, L.; Godino-Llorente, J.; Blanco-Velasco, M.; Castellanos-Dominguez, G. Feature extraction from parametric time–frequency representations for heart murmur detection. *Ann. Biomed. Eng.* **2010**, *38*, 2716–2732. [[CrossRef](#)]

10. Soares, E.; Angelov, P.; Gu, X. Autonomous Learning Multiple-Model zero-order classifier for heart sound classification. *Appl. Soft Comput.* **2020**, *94*, 106449. [[CrossRef](#)]
11. Gu, X.; Angelov, P.P. Self-organising fuzzy logic classifier. *Inf. Sci.* **2018**, *447*, 36–51. [[CrossRef](#)]
12. Angelov, P.; Gu, X.; Kangin, D. Empirical data analytics. *Int. J. Intell. Syst.* **2017**, *32*, 1261–1284. [[CrossRef](#)]
13. Lughofer, E. On-line incremental feature weighting in evolving fuzzy classifiers. *Fuzzy Sets Syst.* **2011**, *163*, 1–23. [[CrossRef](#)]
14. Pedrycz, W. Neurocomputations in relational systems. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 289–297. [[CrossRef](#)]
15. Pedrycz, W. Fuzzy neural networks and neurocomputations. *Fuzzy Sets Syst.* **1993**, *56*, 1–28. [[CrossRef](#)]
16. De Campos Souza, P.V. Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature. *Appl. Soft Comput.* **2020**, *92*, 106275. [[CrossRef](#)]
17. Škrjanc, I.; Iglesias, J.A.; Sanchis, A.; Leite, D.; Lughofer, E.; Gomide, F. Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A Survey. *Inf. Sci.* **2019**, *490*, 344–368. [[CrossRef](#)]
18. Lughofer, E. Single-Pass Active Learning with Conflict and Ignorance. *Evol. Syst.* **2012**, *3*, 251–271. [[CrossRef](#)]
19. Sayed-Mouchaweh, M.; Lughofer, E. *Learning in Non-Stationary Environments: Methods and Applications*; Springer: New York, NY, USA, 2012.
20. Khamassi, I.; Sayed-Mouchaweh, M.; Hammami, M.; Ghedira, K. Discussion and review on evolving data streams and concept drift adapting. *Evol. Syst.* **2017**, *9*, 1–23. [[CrossRef](#)]
21. Iglesias, J.A.; Angelov, P.; Ledezma, A.; Sanchis, A. Human activity recognition based on evolving fuzzy systems. *Int. J. Neural Syst.* **2010**, *20*, 355–364. [[CrossRef](#)]
22. Ordonez, J.; Iglesias, J.; de Toledo, P.; Ledezma, A.; Sanchis, A. Online activity recognition using evolving classifiers. *Expert Syst. Appl.* **2013**, *40*, 1248–1255. [[CrossRef](#)]
23. Malcangi, M.; Quan, H.; Vaini, E.; Lombardi, P.; Di Rienzo, M. Evolving fuzzy-neural paradigm applied to the recognition and removal of artefactual beats in continuous seismocardiogram recordings. *Evol. Syst.* **2018**, *11*, 443–452. [[CrossRef](#)]
24. Kasabov, N.; Feigin, V.; Hou, Z.G.; Chen, Y.; Liang, L.; Krishnamurthi, R.; Othman, M.; Parmar, P. Evolving spiking neural networks for personalised modelling, classification and prediction of spatio-temporal patterns with a case study on stroke. *Neurocomputing* **2014**, *134*, 269–279. [[CrossRef](#)]
25. Souza, P.V.C. Regularized fuzzy neural networks for pattern classification problems. *Int. J. Appl. Eng. Res.* **2018**, *13*, 2985–2991.
26. Elsken, T.; Metzen, J.; Hutter, F. Neural Architecture Search: A Survey. *J. Mach. Learn. Res.* **2019**, *20*, 1–21.
27. Zoph, B.; Le, Q.V. Neural architecture search with reinforcement learning. *arXiv* **2016**, arXiv:1611.01578.
28. Baker, B.; Gupta, O.; Naik, N.; Raskar, R. Designing neural network architectures using reinforcement learning. *arXiv* **2016**, arXiv:1611.02167.
29. Du, W.; Guo, X.; Wang, Z.; Wang, J.; Yu, M.; Li, C.; Wang, G.; Wang, L.; Guo, H.; Zhou, J.; et al. A New Fuzzy Logic Classifier Based on Multiscale Permutation Entropy and Its Application in Bearing Fault Diagnosis. *Entropy* **2020**, *22*, 27. [[CrossRef](#)]
30. Dy, J.G.; Brodley, C.E. Feature subset selection and order identification for unsupervised learning. In Proceedings of the Seventeenth International Conference on Machine Learning (ICML '00), Stanford, CA, USA, 2–29 July 2000; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2000; pp. 247–254.
31. Qin, S.; Li, W.; Yue, H. Recursive PCA for Adaptive Process Monitoring. *J. Process Control* **2000**, *10*, 471–486.
32. Albert, A. *Regression and the Moore-Penrose Pseudoinverse*; Elsevier: Amsterdam, The Netherlands, 1972.
33. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference and Prediction—Second Edition*; Springer: New York, NY, USA; Berlin/Heidelberg, Germany, 2009.
34. Lughofer, E. Flexible evolving fuzzy inference systems from data streams (FLEXFIS++). In *Learning in Non-Stationary Environments: Methods and Applications*; Sayed-Mouchaweh, M., Lughofer, E., Eds.; Springer: New York, NY, USA, 2012; pp. 205–246.
35. Huang, G.B.; Chen, L.; Siew, C.K. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Netw.* **2006**, *17*, 879–892. [[CrossRef](#)]

36. Rosa, R.; Gomide, F.; Dovzan, D.; Skrjanc, I. Evolving neural network with extreme learning for system modeling. In Proceedings of the 2014 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), Linz, Austria, 2–4 June 2014; pp. 1–7.
37. Lughofer, E. *Evolving Fuzzy Systems—Methodologies, Advanced Concepts and Applications*; Springer: Berlin/Heidelberg, Germany, 2011.
38. Lughofer, E.; Bouchot, J.L.; Shaker, A. On-line elimination of local redundancies in evolving fuzzy systems. *Evol. Syst.* **2011**, *2*, 165–187. [[CrossRef](#)]
39. Klement, E.P.; Mesiar, R.; Pap, E. Triangular norms. Position paper III: Continuous t-norms. *Fuzzy Sets Syst.* **2004**, *145*, 439–454. [[CrossRef](#)]
40. Lemos, A.; Caminhas, W.; Gomide, F. New uninorm-based neuron model and fuzzy neural networks. In Proceedings of the Fuzzy Information Processing Society (NAFIPS), 2010 Annual Meeting of the North American, Toronto, ON, Canada, 12–14 July 2010; pp. 1–6.
41. Calvo, T.; De Baets, B.; Fodor, J. The functional equations of Frank and Alsina for uninorms and nullnorms. *Fuzzy Sets Syst.* **2001**, *120*, 385–394. [[CrossRef](#)]
42. De Campos Souza, P.V.; Lughofer, E. An Advanced Interpretable Fuzzy Neural Network Model based on Uni-Nullneuron constructed from N-uninorms. *Fuzzy Sets Syst.* **2020**.
43. Bifet, A.; Holmes, G.; Kirkby, R.; Pfahringer, B. MOA: Massive Online Analysis. *J. Mach. Learn. Res.* **2010**, *11*, 1601–1604.
44. Bifet, A.; Holmes, G.; Kirkby, R.; Pfahringer, B. *Data Stream Mining—A practical Approach*; Technical Report; Department of Computer Sciences, University of Waikato: Hamilton, New Zealand, 2011.
45. Potes, C.; Parvaneh, S.; Rahman, A.; Conroy, B. Ensemble of feature-based and deep learning-based classifiers for detection of abnormal heart sounds. In Proceedings of the 2016 Computing in Cardiology Conference (CinC), Vancouver, BC, Canada, 11–14 September 2016; pp. 621–624.
46. Zabihi, M.; Rad, A.B.; Kiranyaz, S.; Gabbouj, M.; Katsaggelos, A.K. Heart sound anomaly and quality detection using ensemble of neural networks without segmentation. In Proceedings of the 2016 Computing in Cardiology Conference (CinC), Vancouver, BC, Canada, 11–14 September 2016; pp. 613–616.
47. Dominguez-Morales, J.P.; Jimenez-Fernandez, A.F.; Dominguez-Morales, M.J.; Jimenez-Moreno, G. Deep neural networks for the recognition and classification of heart murmurs using neuromorphic auditory sensors. *IEEE Trans. Biomed. Circuits Syst.* **2017**, *12*, 24–34. [[CrossRef](#)] [[PubMed](#)]
48. Latif, S.; Usman, M.; Rana, R.; Qadir, J. Phonocardiographic sensing using deep learning for abnormal heartbeat detection. *IEEE Sens. J.* **2018**, *18*, 9393–9400. [[CrossRef](#)]
49. Whitaker, B.M.; Suresha, P.B.; Liu, C.; Clifford, G.D.; Anderson, D.V. Combining sparse coding and time-domain features for heart sound classification. *Physiol. Meas.* **2017**, *38*, 1701. [[CrossRef](#)]
50. Xiao, B.; Xu, Y.; Bi, X.; Zhang, J.; Ma, X. Heart sounds classification using a novel 1-D convolutional neural network with extremely low parameter consumption. *Neurocomputing* **2020**, *392*, 153–159. [[CrossRef](#)]
51. Deng, M.; Meng, T.; Cao, J.; Wang, S.; Zhang, J.; Fan, H. Heart sound classification based on improved MFCC features and convolutional recurrent neural networks. *Neural Netw.* **2020**, *130*, 22–32. [[CrossRef](#)]
52. Shukla, S.; Singh, S.K.; Mitra, D. An efficient heart sound segmentation approach using kurtosis and zero frequency filter features. *Biomed. Signal Process. Control* **2020**, *57*, 101762. [[CrossRef](#)]
53. Zeng, W.; Yuan, J.; Yuan, C.; Wang, Q.; Liu, F.; Wang, Y. A new approach for the detection of abnormal heart sound signals using TQWT, VMD and neural networks. *Artif. Intell. Rev.* **2020**, 1–35.
54. Chen, Y.; Wei, S.; Zhang, Y. Classification of heart sounds based on the combination of the modified frequency wavelet transform and convolutional neural network. *Med. Biol. Eng. Comput.* **2020**, *58*, 2039–2047. [[CrossRef](#)] [[PubMed](#)]
55. Aziz, S.; Khan, M.U.; Alhaisoni, M.; Akram, T.; Altaf, M. Phonocardiogram Signal Processing for Automatic Diagnosis of Congenital Heart Disorders through Fusion of Temporal and Cepstral Features. *Sensors* **2020**, *20*, 3790. [[CrossRef](#)] [[PubMed](#)]
56. Chowdhury, T.H.; Poudel, K.N.; Hu, Y. Time-Frequency Analysis, Denoising, Compression, Segmentation, and Classification of PCG Signals. *IEEE Access* **2020**, *8*, 160882–160890. [[CrossRef](#)]
57. Li, J.; Ke, L.; Du, Q. Classification of Heart Sounds Based on the Wavelet Fractal and Twin Support Vector Machine. *Entropy* **2019**, *21*, 472. [[CrossRef](#)]

58. Noman, F.; Salleh, S.; Ting, C.; Samdin, S.B.; Ombao, H.; Hussain, H. A Markov-Switching Model Approach to Heart Sound Segmentation and Classification. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 705–716. [[CrossRef](#)]
59. Jeong, Y.S.; Kang, A.R.; Jung, W.; Lee, S.J.; Lee, S.; Lee, M.; Chung, Y.H.; Koo, B.S.; Kim, S.H. Prediction of Blood Pressure after Induction of Anesthesia Using Deep Learning: A Feasibility Study. *Appl. Sci.* **2019**, *9*, 5135. [[CrossRef](#)]

Publisher’s Note : MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).