Exploration on Bubble Entropy

George Manis, Dimitrios Platakis and Roberto Sassi

***Abstract*—Bubble entropy is a recently proposed entropy metric. Having certain advantages over popular definitions, bubble entropy finds its place in the research community map. It belongs to the family of entropy estimators which embed the signal into an m-dimensional space. Two are the main drawbacks for which those methods are criticized: the high computational cost and the dependence on parameters. Bubble entropy can be an answer to both, since computation can be performed in linear time and the dependence on parameters can be considered minimal in many practical situations.**

Popular entropy definitions, which are built over an embedding of the signal, mainly rely on two parameters: the size of the embedding space m and a tolerance r, which set a threshold over the distance between two points in the m-dimensional space to be considered similar. Bubble entropy totally eliminates the necessity to define a threshold distance, while it largely decouples the entropy estimation from the selection of the actual size of the embedding space in stationary conditions.

Bubble entropy is compared to popular entropy definitions on theoretical and experimental basis. Theoretical analyses reveal significant advantages. Experimental analyses, comparing congestive heart failure patients and controls subjects, show that bubble entropy outperforms other popular, well established, entropy estimators in discriminating those two groups. Furthermore, machine learning-based feature ranking and experiments show that bubble entropy serves as a valuable source of features for AI decision-support algorithms.

***Index Terms*—bubble entropy; entropy estimator; embedding space; elimination of parameters; fast computation; sample entropy; permutation entropy; HRV analysis; Biomedical Engineering**

I. Introduction

ENTROPY, as a pillar in information theory, was introduced by Claude Shannon in 1948. Since then, it has been a powerful tool in various areas of signal interpretation. In biomedical engineering, entropy analysis has been used to measure the irregularity of biomedical signals. Significant applications of entropy included cardiovascular and neurosensorial signal analysis, cancer research, sleep disorders, and gait analysis.

Bubble entropy is a new entropy estimator with limited dependency on parameters. In [1], it was introduced as almost free of parameters. The word *"almost"* had been used to express that bubble entropy had successfully eliminated the most influential and, nevertheless, difficult to estimate parameter: the threshold distance r. Still, bubble entropy did not lack discriminating capability, compared to other standard, widely accepted and used definitions of entropy.

George Manis and Dimitrios Platakis are with the Department of Computer Science and Engineering, School of Engineering, University of Ioannina, Ioannina, 45500 Greece.

Roberto Sassi is with the Dipartimento di Informatica, Università degli Studi di Milano, Milano, 20133 Italy.

During these years, the research community has shown interest in this new metric. Many papers appeared in the literature employing bubble entropy in their methodological pipeline [2]–[11]. Some of them used bubble entropy as feature to train and test machine learning classifiers [2]–[4]. Others used the method as the main means of analysis. In [5], in order to examine the association between pressure injuries and abdominal temperature, bubble entropy was used to assess the irregularity of the temperature time series. Furthermore, bubble entropy was used for an automated categorization of sleep stages using electroencephalogram (EEG) signals [6], [12], [13] or to discriminate between term and preterm births [8]. In some papers bubble entropy or extensions of bubble entropy appeared in the title [7]–[9] or in the keywords [11], [14]. It was included in reviews [15], [16] and software libraries [17], [18]. Extensions and modifications have also been proposed in other works [10], [11].

We duly investigated the properties of bubble entropy, employing theoretical [19]–[21] and experimental analyses [22], [23]. These papers revealed interesting aspects, which gave motivation for further analysis and exploration.

In this paper, we summarize the up-to-date research, we investigate new challenges, present new ideas, and suggest lines of research. The aim of the paper is multiple:

- It introduces an integrated depiction of bubble entropy, summarizing in a single section the advantages of the method and theoretical and experimental analyses which supported them in earlier literature.
- It investigates why and how it is valuable to consider for bubble entropy a spectrum of values of m at once, instead of selecting a specific embedding space dimension, when the series cannot be considered stationary.
- It illuminates the ability of the estimator to work and retrieve information from high-dimensional spaces, extracting different information than that extracted in low-dimensional spaces, still when the series cannot be considered stationary.
- It compares bubble entropy with popular definitions of entropy, spotting the method on the research map. It discusses where and how bubble entropy outperforms sample entropy [24], [25], the most popular definition of entropy during the last two decades. It also compares bubble entropy with permutation entropy [26], a definition of entropy which superficially looks similar to bubble entropy and often confused with. This paper clarifies the relation between bubble entropy and permutation entropy and presents a number of advantages of bubble entropy over permutation entropy.
- It introduces a Python and Matlab library which provide implementations of bubble entropy in linear time. The core development has been done in C, which further

reduces the execution time. Even though bubble entropy is based on sorting, its computation can be done by a linear algorithm [1].

The rest of the paper is structured as follows. Section II outlines bubble entropy and summarizes our previous work. Section III discusses in detail an algorithm which allows computation of bubble entropy in (mn) time. In section IV bubble entropy is compared with other popular entropy definitions. Then, in specific, section V shows why it is not important to estimate the size of the embedding space, when using bubble entropy with stationary series, and presents comparative experimental results on where it outperforms sample entropy and permutation entropy. The last section concludes this work.

II. Bubble Entropy

A time series $x = x_1, x_2, \ldots, x_N$ of size N is embedded into an m dimensional space leading to a second time series of size $N{-}m{+}1$ of the vectors:

$$\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N, \text{ where } \boldsymbol{X}_j = (x_j, x_{j+1}, ..., x_{j+m-1}).$$

Using *bubble sort*, the elements in each vector $\boldsymbol{X}_j$ are sorted, counting the number of swaps performed. The probability mass function (*pmf*) p_i of having i swaps is used to evaluate the second-order Rényi entropy:

$$H^m_{\text{swaps}} = -\log \sum_{i=0}^{\binom{m}{2}} p_i^2. \tag{1}$$

Bubble entropy is the normalized difference of the entropy of the sorting effort (swaps) required to sort vectors of length $m{+}1$ and m:

$$bEn = \left(H^{m+1}_{\text{swaps}} - H^m_{\text{swaps}}\right) / N^m_{\text{norm}}, \tag{2}$$

where N^m_{norm} is a normalization factor.

The normalization factor: Many different choices can be performed (and indeed we proposed in our works different alternatives). The most effective is to select the difference of the values of H^m_{swaps} for *White Gaussian Noise* (WGN) in these embedding spaces (which we termed W^m_{swaps} instead of H^m_{swaps} for clarity):

$$N^m_{\text{norm}} = W^{m+1}_{\text{swaps}} - W^m_{\text{swaps}}. \tag{3}$$

While an exact value of W^m_{swaps} and how it is derived can be found in [21], an approximated estimate, displaying an error $<1‰$ for $m > 30$ [21], is:

$$W^m_{\text{swaps}} \approx \frac{1}{2} \log \left(\pi \frac{m(m-1)(2m+5)}{18}\right). \tag{4}$$

In the first definition ("original") of bubble entropy [1], the normalization factor was instead set to:

$$N^m_{\text{norm}} = \log \left(\frac{m+1}{m-1}\right), \tag{5}$$

which is related to $\log[(m{-}1)m/2{+}1]$, the maximum value that H^m_{swaps} can reach for a uniform distribution of swaps.

In the Python/Matlab library publicly available (please refer to section III), the programmer can select among the exact, the approximated or the original value of N^m_{norm}.

The two-step ahead estimator: In our paper [21] we showed that, due to basic combinatorial reasons, for strongly antipersistent stochastic processes (*e.g.*, $x_n = a x_{n-1} + w_n$, where $a \to -1$ and w_n is a white Gaussian noise), the *pmf* p_i of having i swaps displays two peaks when m is even, while only one when it is odd. Thus, $H^{m+1}_{\text{swaps}} < H^m_{\text{swaps}}$ and negative values of bubble entropy are possible (as well as oscillations between negative and positive values of bubble entropy while m grows).

In these situations, a theoretically solid workaround is to consider bubble entropy as the difference in swap entropy between two embedding spaces of size $m{+}2$ and m, rather than between $m{+}1$ and m:

$$bEn_{+2} = \frac{H^{m+2}_{\text{swaps}} - H^m_{\text{swaps}}}{W^{m+2}_{\text{swaps}} - W^m_{\text{swaps}}}. \tag{6}$$

However, when the process is only mildly anti-correlated as it happens in real-world signals, this effect is not significant, since it does not affect the discriminating power of bubble entropy.

Values for certain known stochastic processes: Our initial experiments with bubble entropy were performed on heart rate variability (HRV) time series, which are commonly modelled with autoregressive (AR) processes, for their short-term behavior, or long memory processes, like fractional Brownian motion (fBm) or fractional Gaussian noise (fGn), for their long-term one. To better understand the characteristics of bubble entropy we therefor studied the values it assumes (theoretically) for these classes of stochastic processes. We also verified with careful numerical simulations that these results were holding in practice.

Summarizing, we verified that it is possible to derive an analytical value of bubble entropy given by an integral expression for any AR process. The formula in [19] is however computationally expensive when m grows, and only for $m \leq 3$, given the existence of an analytical formulation for the quadrant probability, an explicit formula can be derived. As expected, when an analytical closed-form formula can be computed, like for $m = 2$, the theoretical value of bubble entropy differs from the entropy rate of the AR process. For example, for the AR process $x[n] = \rho x[n{-}1] + w[n]$ with $|\rho| < 1$ and $w[n] \sim \text{WGN}(0, \sigma_w^2)$,

$$bEn = \frac{\log \pi^2 - \log \left[\left(\arccos \frac{1-\rho}{2}\right)^2 + 4\left(\arccos \frac{\sqrt{1+\rho}}{2}\right)^2\right]}{\log \frac{9}{5}},$$

while the entropy rate is $\frac{1}{2}\log\left[2\pi e(1{-}\rho^2)\right]$. Also, the value of bubble entropy did not change much with growing values of m, for a wide range of values of ρ (please refer to fig. 4 in [21]).

Considering fGn [20], which is a stationary process like the AR process of the previous paragraph, bubble entropy also showed a very small, if minimal, dependence on m. Empirically, it scaled as $H/2{+}3/4$, where H is the Hurst or scaling parameter. On the contrary, for fBm, a non-stationary process, when m grew, bubble entropy approached a constant (larger) value of $4/3$. In fact, the continuously growing or decaying ephemeral trends in fBm render the range of swaps necessary for sorting very broad, leading to values of swap entropy H^m_{swaps} very close to the maximum possible value of $\log[(m-1)m/2+1]$. Therefor, bubble entropy behaves like a scaling estimator for stationary Gaussian long-memory processes, but this is not the case when non-stationarity becomes relevant.

A physical meaning for the sorting effort: There is a physical meaning behind the selection of the sorting effort as the basis of the distribution. The *sorting effort*, as an amount, expresses *task* or *work*. In Physics, *work* is responsible for energy transfer or change. *Energy* is closely correlated to *entropy*: the more "*disordered*" a system, the less of a system's energy is available to do work. Rényi entropy of order 2 counts and reports the logarithm of the energy of the system (equation 1). If we consider that in a sorted vector the "disorder" is minimal and in a unsorted one much larger, the more effort/work we need to bring the vector into the minimal "disorder", the higher the "disorder" in it. Thus, the distribution on which the entropy is computed is based on a quantity closely related to what finally is estimated, giving a physical meaning to the selection of the sorting tasks as the basis of the distribution.

III. Software Library with a Linear Implementation of Bubble Entropy

Bubble sort is an $O(n^2)$ algorithm. Even though sorting can be performed in $O(n \log n)$ steps using Quick sort, sorting still remains expensive.

The computation of bubble entropy requires only $O(mn)$ steps, where m is the size of the embedding space and n the size of the signal. This is not a paradox, even though bubble entropy depends on bubble sorting. Taking a closer look at what is sorted, one can easily notice that sorting is performed on the vectors and not on the whole signal. The quadratic computation is on m and, based on this observation, the algorithm complexity is $O(m^2 n)$.

Even though m is very small, compared to n and in theory we do not consider it significant, in practice the nested iteration on m still consumes valuable processing resources. We can exploit here, in an attempt to further reduce the computational cost, that vectors v_i and v_{i+1} have in common $m-1$ elements. We call $v_i^{\neq}$ the element of v_i which does not belong in v_{i+1}. The element $v_i^{\neq}$ is the first element of the vector v_i. We call $v_{i+1}^{\neq}$ the element of v_{i+1} which does not belong in v_i. This element is the last element of the vector v_{i+1}.

Having sorted the vector v_i we have already done most of the sorting effort for the vector v_{i+1}. Let us assume that sorting v_i costs v_i^{cost} steps. If we move the element $v_i^{\neq}$ at the beginning of the vector v_i and count the required swaps, we can subtract the necessary swaps from v_i^{cost}. The resulting cost, let's call it v_i^{cost-}, is the cost to sort the $m-1$ last elements of v_i, i.e. the first $m-1$ elements of v_{i+1}. Having sorted the $m-1$ first elements of v_{i+1} we insert $v_{i+1}^{\neq}$ from the right hand side of the $m-1$ sorted elements of v_{i+1} and move it as many positions as necessary, until v_{i+1} is sorted. We add the cost of this moving to the cost v_i^{cost-} and we get the sorting effort v_{i+1}^{cost} for the vector v_{i+1}.

The above algorithm needs at maximum $O(m)$ steps to bring $v_i^{\neq}$ in the first position of v_i and $O(m)$ steps to bring $v_{i+1}^{\neq}$ at the required position. We have reduced the complexity over m from quadratic to linear: $O(mn)$. Since $m \ll n$ and m is independent from the length of the signal, we can consider the complexity as linear.

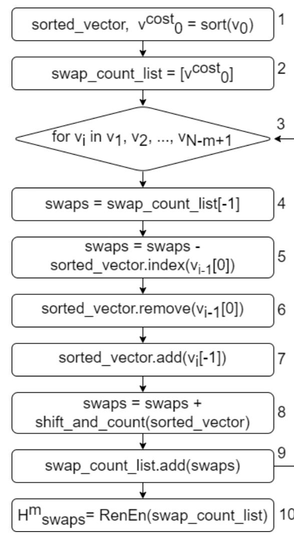

Fig. 1. A flowchart of the algorithm.

The algorithm is described in the flowchart on figure 1. In block 1, the sorting of the first vector is performed. Apart from this point, using bubble sort is no longer necessary, only swaps between neighboring elements in the vector are performed. In block 2, we initialize the list $swap\_count\_list$ with its first element, i.e. the cost to sort v_0, performed in block 1. After that, in blocks 3-9, we repeatedly remove from $sorted\_vector$ the first element of v_{i-1}, i.e. $v_{i-1}[0]$ (block 6). The variable $swaps$ is initialized in block 4 with the number of swaps necessary to sort element v_{i-1}, i.e. the last element in the $swap\_count\_list$. In block 5 we subtract the index of $v_{i-1}[0]$ from the $swaps$ counter. In block 7, the new element of $sorted\_vector$ is added at the last position of $sorted\_vector$. At this point, all elements of the $sorted\_vector$ are sorted, except from the last one. To sort the vector $sorted\_vector$ we only have to move the most recently added element $v_i[-1]$ in the proper position and count the necessary steps. This is done in block 8 with the $shift\_and\_count$ function. The list $swap\_count\_list$ is updated with the new element in block 9. After the end of the loop, in block 10, the algorithm reports the computed entropy.

Code for the linear algorithm is available at: [link][1] and can be freely used. There are two libraries available there, one for the Python programming language and one for Matlab. The core implementation has been done in C, to ensure minimal overhead. Wrappers on top of C provide the interfaces for Python and Matlab. The C source code is also available. It can also be called by C programs or additional wrappers can be implemented for other language, according to the programmers' needs.

[1]link will be available when/if the paper is accepted

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2025.3593153

IV. Comparison with Popular Entropy Definitions

Instead of summarizing the advantages of bubble entropy in a dedicated section, we found it more interesting to illuminate the advantages by comparison with other popular entropy definitions. We selected two: (a) permutation entropy, since both bubble entropy and permutation entropy are based on sorting and present superficial similarities and, (b) sample entropy, since it is the most popular definition of entropy, especially in the biomedical engineering field.

Other interesting, but less popular, entropy definitions which could constitute the basis for bubble entropy to establish its position in the research field, include: corrected conditional entropy [27], a measure which adjusts the calculation of conditional entropy to account for biases that arise in finite or noisy data; dispersion entropy [28], an entropy measure which uses symbolization, grouping into patterns and Shannon entropy to assess the probability of the occurrence of each pattern; fuzzy entropy [29], where a fuzzy entropy function is used to evaluate the similarity between points; distribution entropy [30], where data points are assigned to bins and the relative frequency of data points is assessed using Shannon entropy; measures that modified permutation entropy [31]–[33]. Detailed comparison with the less popular entropy definitions is in our plans for a future work. An interesting review of all those methods and entropy analysis of univariate biomedical signals can be found in [34]

A. Comparison with permutation entropy

Bubble entropy and permutation entropy might look quite similar. However, even though they are based on sorting elements inside the vectors, the two definitions present remarkable differences, which are analyzed in the following.

Distribution of the sorting effort: permutation entropy estimates the entropy on the distribution of the different sorting patterns, formed by the initial positions of the elements in the vector, *i.e.*, the positions before sorting. Bubble entropy, instead, computes the distribution of the sorting effort for each vector and estimates the entropy on the distribution of these efforts. This small difference is very important and affects the discriminating capability and the range of m values, where each method is applicable.

Employing the sorting effort results into grouping of several sorting patterns under one common factor: the task performed. This leads to more effective grouping. A swap between two successive samples in the original signal, produces $m-1$ different sorting patterns, since these two samples are common in $m-1$ vectors in the embedding space. In addition, a swap between the i_{th} and the i_{th+1} element of a vector or between the j_{th} and the j_{th+1} elements ($i \neq j$) contribute towards different patterns, even though it is a swap between the same two elements. The place in the signal, at which the swap has been performed, is of less importance for the estimation of entropy than the fact that this swap has been actually performed. In all the above cases, permutation entropy produces patterns, while bubble entropy just adds 1 to the sorting task of the vector, estimating the energy necessary to reach the minimal entropy state.

A physical meaning: As stated earlier, there is a physical meaning behind the selection of the sorting task as the basis of the distribution in bubble entropy. In contrary, employing the sorting patterns is something with a less clear physical meaning, which appears more clearly only when $m \to \infty$, as permutation entropy then it provides an upper bounds for the Kolmogorov-Sinai Entropy in several circumstances [35].

The number of states: Creating different patterns for simple swaps has also one more effect: it increases the number of possible states (the number of different possible patterns) in the distribution. For permutation entropy and a vector or size m the possible number of states is $m!$. This is a very large number with more than 3 million states, for $m=10$. The large number of possible states leads to a less descriptive distribution for practical signals with a much smaller number of samples, learning to low height, large width and relatively equally distributed small number of patterns in each state. On the other hand, bubble entropy performs at the maximum $m(m+1)/2$ swaps for a vector of size m (55 states for $m=10$), leading to a more descriptive distribution with frequent values concentrated in the middle and rare ones at the sides.

Range of m: The large number of sorting patterns leads to a limitation in the range of the examined values of m. This is connected to the previous issue: it is simply practically unfeasible to consider $m!$ states when m is large. This is not the case in bubble entropy where the range of useful values can be much larger. For example, going back to HRV, when on entire day of data is considered or about 10^5 samples, probability mass functions for $m=10$ are estimated with about 1.78×10^3 samples for possible swap (bubble entropy) compared to 0.0276 samples per state (permutation entropy). For nonstationary signals, with a larger m, it is possible to explore a wider variety of embedding spaces and to produce a larger number of features, when a machine learning classifier is to be employed. The larger the number of features, the more flexible the feature selection and the more likely the classifier to approach the optimal solution.

Difference of entropy between spaces with different embedding dimensions: While permutation entropy computes and reports the entropy of the distribution in the m-dimensional space, bubble entropy follows the popular sample entropy [24], [25] and approximate entropy [36], which both report the difference between the entropy estimated in embedding spaces with dimensions $m+1$ and m.

Bubble entropy computes the difference in the swap entropy, or as expressed in equation 1, the entropy of the distribution of swaps necessary to sort the vectors $\boldsymbol{X}_j$ of length $m+1$ and m. In accordance with popular and well established entropy definitions, Swap entropy expresses complexity in a way similar to Permutation entropy, i.e. by examining ordering. In our previous papers [1], [21] we showed that bubble entropy is not the entropy rate of a stochastic process; also, it is not maximal for white noise [21]. Nevertheless, similarly to approximate entropy, sample entropy and sorting entropy, it reflects the new information that is available in a embedding space of size $m+1$, when the information in the embedding space of size m is a given, or else, it reflects the average uncertainty that remains in space $m+1$, with respect to the uncertainty of space $m1$. However, bubble entropy does not directly quantify the conditional entropy (entropy rate) of the series, like sample entropy and approximate entropy or the corrected conditional entropy [27] and the equivalent measure based on nearest neighbor metrics [37], but it is a practical estimator which assesses the change in entropy of the ordering of portions of the samples of length m, when an extra element is added.

Sorting entropy [26] is a definition similar to permutation entropy. It estimates the difference between permutation entropy computed in spaces with dimensions $m+1$ and m. Theoretical comparison with bubble entropy is not necessary here, due to the similarities between permutation entropy and sorting entropy. We will include sorting entropy, though, in our experimental analysis later in this paper.

Rényi entropy: How entropy is estimated from the distribution is also an interesting difference. While permutation entropy uses the definition of Shannon, bubble entropy prefers the definition of Rényi of order 2 (equation 1), since: a) Rényi entropy of order 2 expresses energy, in accordance to the rest of the philosophy of bubble entropy and, b) Rényi entropy suppresses the more rare instances and emphasizes on the most frequent ones, minimizing the influence of rare occurrences or noise, with the latter being very common in biomedical signals.

Similarity - lack of r: Even though, at first sight, the most important similarity between bubble entropy and permutation entropy seems to be the use of sorting, this is not true. The discussion above shows that sorting is a non significant similarity, since it is employed and exploited in a very different way in the two definitions. The interesting common characteristic is the lack of the threshold distance r, something that make them less depended on parameters and human/subjective estimations.

B. Comparison with sample entropy

Sample entropy is the most popular definition of entropy and more than a thousand of papers have based their scientific results on sample entropy. Nevertheless, bubble entropy present some significant advantages over sample entropy, which are listed below.

Dependency on parameters: sample entropy is heavily based on parameters. The difficulty in estimating the parameters has led to the common practice of using standard, stereotyped values, i.e. $m=2$, $r=0.2$, in all cases. On the other hand, bubble entropy depends only on one integer parameter: the dimension of the embedding space m, a fundamental amount in all definitions. Later in this paper, we will show how and why bubble entropy can be used, without the selection of m being important.

Computational cost: A second disadvantage of sample entropy is the high computational cost. The definition of the method implies a quadratic algorithm: $O(n^2)$. Even though fast implementations or implementations with smaller complexity have been proposed in the literature [38], [39], the computation of sample entropy is still very expensive. Bubble entropy uses bubble sort, which is also a quadratic algorithm $O(m^2)$. However, we showed earlier, in section III that bubble entropy can be computed in linear time. A linear algorithm allows experiments with a large number of different m values and produce a large number of features for a machine learning classifier, each describing the behavior in a different embedding space. A fast algorithm also allows the computation of all meaningful values of m, reducing or eliminating the dependency of the method from parameter m.

Range of values of m: For a stationary process, also sample entropy, like bubble entropy, would be independent from the value of m selected. In practice, due to the slow convergence of the estimator, with increasing value of m the value of sample entropy diverges from its expected value (or cannot be estimated). For example, for a white gaussian noise WGN(0,1) and a sequence length of 10^5, values of sample entropy are very difficult to obtain for $m \geq 9$ and $r=0.2$; when $r=0.1$, already $m=7$ is critical. On the contrary, in the same situation, bubble entropy can be estimated with ease still for $m=50$. A workaround for sample entropy would be to increase further the length of the series to foster convergence, but the high computational costs limit substantially this possibility.

As a consequence, sample entropy computation usually involves small values of m, $m=1$ or $m=2$, even $m=3$, but for $m>4$ the discriminating capability is significantly reduced or lost, with the typical length of practical signals. Thus, sample entropy is estimated based on very small vectors, which describe very local information. On the other hand, bubble entropy can extract information based on both short and larger vectors, extending the base of the analysis to longer relations between samples. While this might be irrelevant in stationary situations, it otherwise opens up new exploratory scenarios.

Tolerance to spikes: In [23] sample entropy is claimed to be more robust to noise and spikes than approximate entropy. Presented in [23] is a thorough tolerance to spikes comparison between sample entropy and bubble entropy. Spikes were artificially and randomly inserted into HRV series and the values of the two metrics were compared before and after the insertion. The mean relative error for sample entropy ranged from 0.084 to 0.435, while the corresponding mean relative error for bubble entropy was much lower, between 0.069 and 0.122. This is a strong hint that bubble entropy might present an increased tolerance to spikes compared to sample entropy.

V. Dependence and Independence on the Size of the Embedding Space

Commonly, studies employing popular definitions of entropy investigate the time series dynamics in relatively small embedding spaces. Increasing the size of the embedding space, after a certain value, decreases the discriminating power of the method (unless un-practically long signals are considered) due to some of the limitation highlighted in the previous sections.

For nonstationary signals, investigating the relations in larger dimensions could reveal much different information. For example, in a heartbeat time series, a value of $m=2$ captures inner-vector relationships only between neighboring beats. However, when $m=30$ is considered, the base of the comparison is a window of approximately 30 seconds, in

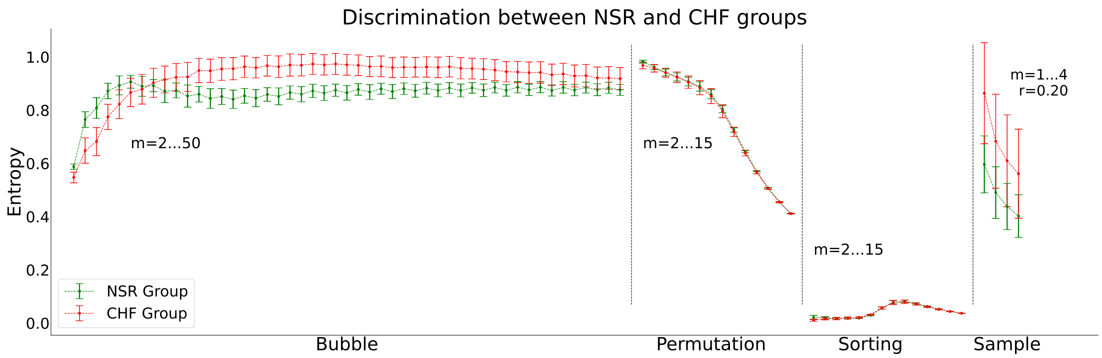

Fig. 2. Comparison of discriminating capability of the four examined entropy definitions.

which more complicated relations may be hidden and more interested phenomena may be encrypted.

The analysis of time series in larger dimensions has significant advantages, which makes bubble entropy especially interesting. On the other hand, the dependence on m is contrary to the initial and fundamental target of bubble entropy, *i.e.*, the independence from the parameters. But in nonstationary conditions, m is an inherent quantity that cannot be eliminated, therefor it could be beneficial not having to select a specific value for the size of the embedding space, while using multiple values of m. Combined with a feature selection method, we can produce a vector with all entropy values in the range between $m=2$ and a large value of m. Using this vector, we can choose as many features as we think fit, picking the most valuable ones. In doing so, the selection of a specific m is no more necessary.

Machine learning technology can be part of the solution. Computing bubble entropy for all reasonable values of m and stacking these values to a machine learning classifier can lead to accurate classifications, investigating various possible complex relations. Of course, any definition of entropy can be used to provide features to a classifier. However, bubble entropy has again some interesting advantages. Using sample entropy as reference, bubble entropy can:

- investigate all reasonable values of m, without having to deal also with an infinite number of pairs m, r.
- be computed in reasonable time, since the algorithm is $O(n)$, when the computation of sample entropy is $O(n^2)$. The high complexity of sample entropy makes the computation of a large number of m, r pairs expensive or even impossible .
- employ a large range of values of m possibly providing more independent features, when both sample entropy (and also permutation entropy) are limited in much smaller ranges and possibly less independent features.

In the following, we will present experimental results to support the above claims. As test data, we used HRV series from two groups of subjects. The first consists of 44 patients with congestive heart failure (*CHF group*). The second consists of 72 controls in normal sinus rhythm (*NSR group*). Data are publicly available on Physionet [40] (such as chf, chf2, nsr and nsr2 data sets).

We trained and tested the accuracy of four machine learning models: Gaussian Naive Bayes classifier (*GNB*), Support Vector Machine (*SVM*) [41], k-Nearest Neighbor (*k-NN*) [42] and Logistic Regression (*LR*) [43]. In an attempt to make a fair comparison, we selected standard classifiers, decoupling the problem from the classifier and the estimation of the classification parameters. Thus, we used the default or reasonably simple values for all classification parameters (GNB: no parameters; SVM: Linear, Sigmoid, Radial Basis Function Kernels, gamma="scale" and C = 1; k-NN: $k=3,\cdots,25$, odd values only; LR: $C=1$, penalty=none). The optimal number and set of features was selected by checking the classification accuracy, after examining all possible feature subsets and all parameters, where necessary. Even though a high amount of features may lead to over-fitting, we used 10-Fold Cross-Validation in order to verify its absence through the insignificant variance of test accuracy throughout the 10 folds, paired with proper training accuracy. 49 features were computed for bubble entropy ($2\leq m\leq 50$). 14 features were computed for permutation entropy and sorting entropy ($2\leq m\leq 15$), although the discriminating power of the methods was significantly reduced after $m=10$, as our experiments showed. 12 features were selected for sample entropy (all m, r pairs when $m=1,\ldots,4$ and $r=0.15, 0.2, 0.25$).

TABLE I
CLASSIFICATION ACCURACY FOR EXAMINED DEFINITIONS

	GNB	SVM	k-NN	LR
Sample entropy	67.27%	71.55%	75.90%	70.53%
Permutation entropy	69.84%	74.31%	73.33%	71.66%
Sorting entropy	69.09%	72.27%	74.16%	68.93%
Bubble entropy	90.53%	88.93%	88.86%	87.80%
All	89.27%	88.93%	89.69%	88.10%

A comparison of the discriminating capability of each feature, when used independently of the others, is contained in figure 2. For each feature, we reported the average value for each of the two populations (with confidence bars) to quickly highlight those metrics for which the two distributions are distinct.

Bubble entropy presents an interesting behavior throughout different values of m. Starting off with low m-values the means of the entropy values of the NSR patients are higher than those of CHF. Around $m=10$ those two means intersect making this area the least valuable in terms of discriminatory ability. For even higher m-values, the means of the groups switch, bringing the CHF means to the top and the NSR means underneath. Additionally, around the area of $m = 20$, bubble entropy reaches its peak in discriminating the two classes. We cannot say if bubble entropy is higher or lower, since there is different behavior in small and in large spaces. In small values of m the behavior is similar to all other entropy metrics. For large numbers of m the means of the groups switch, suggesting changes on the HRV series itself at larger scales.

Our results are, however, not in contradiction with the statement that Bubble Entropy is only mildly dependent on m. As we showed for AR processes in [1] (fig. 4) and for fractional Gaussian noises in [20] (fig. 1), when the process is stationary the dependence on m is very weak and bubble entropy converges quickly. For example, for a white Gaussian noise, for growing values of m and using the first normalization we proposed in [1], the difference in bubble entropy between two nearby scales m and $m+1$ is asymptotic to $\approx \frac{3}{4m^2}$ which for $m = 7$ and $m = 8$ is about 2% off the theoretical value $3/4$.

On the other hand, when the process is nonstationary or a composition of processes acting at different scales, it is reasonable that any metric, if capable of exploring that range of scales, might be able to detect the changes in complexity which might happen with m. For example, with respect to HRV of CHF patients, the short term fractal exponent $\mathrm{DFA}\alpha_1$ computed with detrended fluctuation analysis (DFA) is smaller for CHF patients who are less likely to survive (thus more "far" apart from normal subjects) [44], while the long term DFA fractal exponent $\mathrm{DFA}\alpha_2$ (which is linked to the α power-law exponent in the low frequencies by the algebraic scaling relation $\alpha = 2\mathrm{DFA}\alpha_2 - 1$) is larger for CHF patients [45]. Thus, the fractality of HRV is higher at larger scales for CHF patients with respect to normal (the Hurst exponent grows), while the opposite is true when considering scales smaller than about 11 beats. A similar variation with scales happens with respect to the complexity of the HRV series when analyzed with bubble entropy (or other metrics capable of exploring a large range of scales like multi-scale entropy [46]).

The best accuracy achieved for each classifier and each entropy definition is reported in table I. Please note that bubble entropy reached remarkably better accuracy levels than all other metrics. When all the features pooled together were fed to the classifier (last line of the table), best accuracy slightly differed from bubble entropy, suggesting that bubble entropy had been considered significant.

To further support elaborate on this last conclusion, we provided in table II the number of features selected and used in the classification, when the best accuracy (as presented in table I) for each method was achieved. Please note that the features extracted from bubble entropy being considered together are more likely to be independent among them, than those derived by the other definitions, reinforcing our claim that the investigation of the problem in large dimensions reveals valuable information.

In an attempt to further evaluate the 89 features, we used two feature ranking methods: ANOVA and mutual information. Based on them, each feature was given a rank between 1 and 89. The inverse of this ranking was plotted in figure 3, grouping together the features of each definition. We used different colors for each definition and similar colors for ANOVA and mutual information. We noted that bubble entropy ranked the highest with a remarkably higher concentration of relevant values than other definitions, suggesting that: (a) with bubble entropy more useful for classification and more independent features had been extracted, and (b) computing features in higher embedding spaces is an advantage and enrich the description of the series.

TABLE II
NUMBER OF FEATURES SELECTED FOR EACH DEFINITION

	GNB	SVM	k-NN	LR
Sample entropy	1	12	4	7
Permutation entropy	1	11	5	1
Sorting entropy	3	12	2	12
Bubble entropy	48	33	11	2

TABLE III
PUBLISHED LITERATURE

Study	ML Models	Features	Accuracy
[47]	SNN, kNN, DT	T-F Domain, Entropy	93.10%
[48]	RF, SVM plus more	AR Burg	100%
[49]	kNN	T-F Domain and more	96.39%
[50]	SONN	Spectral Analysis	83.65%
This work	GNB,SVM,kNN,LR	Bubble entropy	90.53%

The key factor in our choice to limit the number of features for sample entropy to 12 was the execution time. Please recall that the computation of sample entropy needs $O(n^2)$ time. Indicatively, for the whole data set of 116 signals with a maximum size 100.000 samples, the computation of sample entropy for 12 pairs of m and r values took approximately 64 minutes, while the computation of bubble entropy for the same signals and $m = 2, \cdots, 50$ required only about 1.5 minutes. Please also note that for the computation of sample entropy the fastest available implementation [38] was used (written in C programming language), while a straightforward implementation would require significantly more time. The implementation of bubble entropy was also in C.

Many recent studies have tried to investigate how well we can discriminate patients using various feature extraction methods and machine learning models, often using more than one feature in classification. In our paper, the purpose was not to study the NSR-CHF problem, but to examine the significance of the features extracted from bubble entropy and compare them with the features extracted from other entropy definitions. Furthermore, the machine learning models and parameters we chose were as simple as possible, to decouple our investigation from parameters selection. We successfully proved, though, that bubble entropy produces more significant features compared to its competitors.

VI. DISCUSSION ON THE CONCLUSIONS

Bubble entropy is a recently proposed definition of entropy, which presents significant advantages over popular definitions,

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2025.3593153

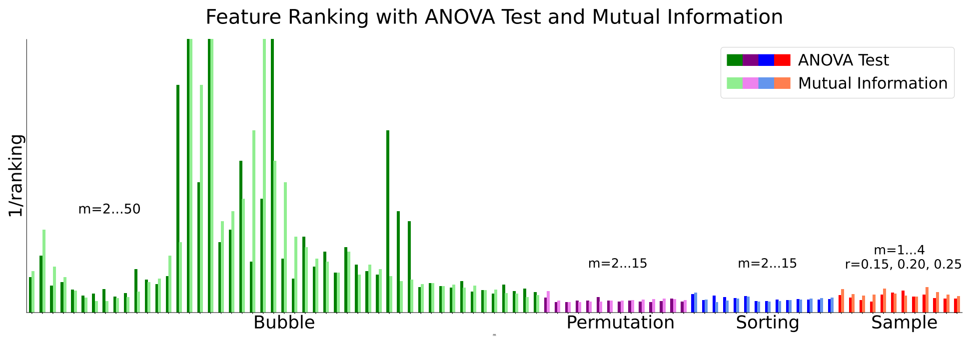

Fig. 3. Visualization of feature ranking. The features with the best ranking have been extracted with bubble entropy.

both in theoretical and experimental analyses. We showed that there is a physical meaning behind each fundamental structural point of the method, *i.e.*, the selection of the sorting task as the basis of the distribution and the selection of Rényi entropy as an estimator. Apart from other advantages, they both express energy, a physical quantity closely connected to entropy. We compared bubble entropy with popular entropy definitions. We pointed out that the similarity with permutation entropy is only superficial, since the distributions on which the computations are based are quite different and express completely different descriptors. We emphasized on the reduced dependence on parameters, since bubble entropy, not only has eliminated the parameter r, as we have showed in our previous works, but also does not require the selection of the parameter m, as we extensively discuss in this paper. Using experimental analyses, and two data sets with HRV series of healthy controls and congestive heart failure patients, we compared the classification accuracy reported by four classifiers, where bubble entropy outperformed sample entropy, permutation entropy and sorting entropy. We ranked the features extracted from the four examined entropy definitions and concluded that the features extracted from bubble entropy were the most significant ones. In order to facilitate the researchers in using bubble entropy, we provide a Matlab / Python library. The library, can be freely downloaded and used. It provides a fast linear implementation of bubble entropy, the core of which is written in C, to ensure minimal overhead.

After having discussed the main points of the paper, we would like, once more, to illuminate the main contributions. In our previous work we emphasized on the importance of limiting the dependence on parameters, focusing on r, but not giving enough attention to m. Bubble entropy computation was based on a single value, by selecting one value of m. In this paper we move to a more thorough investigation, in which all reasonable values of m are considered, significantly improving the discriminating power.

In this paper, we explain why and how the selection of m can be avoided. The parameter m is important and fundamental for bubble entropy and for any similar entropy definition. However, with bubble entropy, we do not need to estimate the optimal value of m or select specific sizes for the m-dimensional space. The idea is to consider all possible reasonable values of m, as this is a small, finite set, and the computational overhead is minimal.

Another significant contribution is the software library distributed with the paper. This is also important and interesting to the research community, since the several implementations of bubble entropy supported by available software libraries are slow and written in Python or other not fast languages. We provide a fast algorithm in C language, wrapped up in Python and Matlab for convenience, the two most popular languages for applications in which bubble entropy can be useful. Something also interesting for the research community, which has not been investigated in our previous work and which will reinforce the value of the method and significantly increase its impact, is the study on the feature selection. Feature importance is used to compare bubble entropy and other entropy definitions. We show, in this paper, that the features extracted with bubble entropy were proved much more significant than the corresponding features of other entropy definitions. We have also investigated a wider range of m than the values of m we chose in our previous work. This led to new important results, since we noticed that there is valuable information in these high embedding dimensions.

Given all the above, we believe that bubble entropy is a very promising definition of entropy, the limits of which have not been extensively examined yet. Currently we are working on different HRV data sets and electroencephalograms. Our future research directions include investigating bubble entropy in embedding spaces with huge dimensions. Given that entropy estimation can extract information from diverse scientific fields, including several biomedical signals, the challenge to investigate the specific problem seems especially interesting.

REFERENCES

[1] G. Manis, M. Aktaruzzaman, and R. Sassi, "Bubble entropy: An entropy almost free of parameters," *IEEE Trans Biomed Eng*, vol. 64, pp. 2711–2718, 2017.

[2] P. Chawla, S. B. Rana, H. Kaur, K. Singh, R. Yuvaraj, and M. Murugappan, "A decision support system for automated diagnosis of Parkinson's disease from EEG using FAWT and entropy features," *Biomedical Signal Processing and Control*, vol. 79, p. 104116, Jan 2023.

[3] A. Rohila and A. Sharma, "Detection of sudden cardiac death by a comparative study of heart rate variability in normal and abnormal heart conditions," *Biocybernetics and Biomedical Engineering*, vol. 40, no. 3, pp. 1140–1154, Jul 2020.

[4] S. Gupta, A. Singh, A. Sharma, and R. K. Tripathy, "dSVRI: A PPG-based novel feature for early diagnosis of type-II diabetes mellitus," *IEEE Sensors Letters*, vol. 6, no. 9, pp. 1–4, Sep 2022.

[5] N. Padhye, D. Rios, V. Fay, and S. K. Hanneman, "Pressure injury link to entropy of abdominal temperature," *Entropy*, vol. 24, no. 8, p. 1127, Aug 2022.

[6] R. K. Tripathy, S. K. Ghosh, P. Gajbhiye, and U. R. Acharya, "Development of automated sleep stage classification system using multivariate projection-based fixed boundary empirical wavelet transform and entropy features extracted from multichannel EEG signals," *Entropy*, vol. 22, no. 10, p. 1141, Oct 2020.

[7] D. Cuesta-Frau, , and B. Vargas, "Permutation entropy and bubble entropy: Possible interactions and synergies between order and sorting relations," *Mathematical Biosciences and Engineering*, vol. 17, no. 2, pp. 1637–1658, 2020.

[8] F. N. del Amor, R. Beskhani, Y. Ye-Lin, J. Garcia-Casado, A. Diaz-Martinez, R. Monfort-Ortiz, V. J. Diago-Almela, D. Hao, and G. Prats-Boluda, "Assessment of dispersion and bubble entropy measures for enhancing preterm birth prediction based on electrohysterographic signals," *Sensors*, vol. 21, no. 18, p. 6071, Sep 2021.

[9] J. Gong, X. Yang, H. Wang, J. Shen, W. Liu, and F. Zhou, "Coordinated method fusing improved bubble entropy and artificial gorilla troops optimizer optimized KELM for rolling bearing fault diagnosis," *Applied Acoustics*, vol. 195, p. 108844, Jun 2022.

[10] Y. Mi, A. Lin, D. Gu, X. Zhang, and X. Huang, "Bubble transfer spectral entropy and its application in epilepsy EEG analysis," *Communications in Nonlinear Science and Numerical Simulation*, vol. 110, p. 106294, Jul 2022.

[11] F. Wang, Z. Chen, and G. Song, "Smart crawfish: A concept of underwater multi-bolt looseness identification using entropy-enhanced active sensing and ensemble learning," *Mechanical Systems and Signal Processing*, vol. 149, p. 107186, Feb 2021.

[12] M. Shahbakhti, M. Beiramvand, T. Eigirdas, J. Solé-Casals, M. Wierzchon, A. Broniec-Wójcik, P. Augustyniak, and V. Marozas, "Discrimination of wakefulness from sleep stage i using nonlinear features of a single frontal eeg channel," *IEEE Sensors Journal*, vol. 22, no. 7, pp. 6975–6984, 2022.

[13] Y. A. Khan, M. Tahreem, and O. Farooq, "Single channel eeg based binary sleep and wake classification using entropy based features," in *2023 International Conference on Recent Advances in Electrical, Electronics & Digital Healthcare Technologies (REEDCON)*, 2023, pp. 100–105.

[14] H. Pal and A. Kumar, "Stability analysis of multiscale bubble entropy and power metric based seizure detection technique with MLA," *IETE Journal of Research*, pp. 1–12, Apr 2021.

[15] W. Li, Y. Zhao, Q. Wang, and J. Zhou, "Twenty years of entropy research: A bibliometric overview," *Entropy*, vol. 21, no. 7, p. 694, Jul 2019.

[16] M. Ribeiro, T. Henriques, L. Castro, A. Souto, L. Antunes, C. Costa-Santos, and A. Teixeira, "The entropy universe," *Entropy*, vol. 23, no. 2, p. 222, Feb 2021.

[17] D. Mayor, D. Panday, H. K. Kandel, T. Steffert, and D. Banks, "CEPS: An open access MATLAB graphical user interface (GUI) for the analysis of complexity and entropy in physiological signals," *Entropy*, vol. 23, no. 3, p. 321, Mar 2021.

[18] M. W. Flood and B. Grimm, "EntropyHub: An open-source toolkit for entropic time series analysis," *PLOS ONE*, vol. 16, no. 11, p. e0259448, Nov 2021.

[19] M. Bodini, M. W. Rivolta, G. Manis, and R. Sassi, "Analytical formulation of bubble entropy for autoregressive processes," in *2020 11th Conference of the European Study Group on Cardiovascular Oscillations (ESGCO)*, 2020, pp. 1–2.

[20] G. Manis, M. Bodini, M. W. Rivolta, and R. Sassi, "Bubble entropy of fractional Gaussian noise and fractional Brownian motion," in *2021 Computing in Cardiology (CinC)*, vol. 48, 2021, pp. 1–4.

[21] ——, "A two-steps-ahead estimator for bubble entropy," *Entropy*, vol. 23, no. 6, 2021, article 761.

[22] G. Manis and R. Sassi, "Relation between fetal HRV and value of umbilical cord artery pH in labor, a study with entropy measures," in *IEEE CBMS*, 2017, pp. 272–277.

[23] G. Manis and R. Sassi, "Tolerance to spikes: A comparison of sample and bubble entropy," in *Comput Cardiol*, 2017, pp. 1–4.

[24] J. S. Richman and J. R. Moorman, "Physiological time-series analysis using approximate entropy and sample entropy," *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 278, no. 6, pp. H2039–H2049, 2000.

[25] D. E. Lake, J. S. Richman, M. P. Griffin, and J. R. Moorman, "Sample entropy analysis of neonatal heart rate variability," *Am J Physiol Regul Integr Comp Physiol*, vol. 283, pp. R789–R797, 2002.

[26] C. Bandt and B. Pompe, "Permutation entropy: A natural complexity measure for time series," *Phys Rev Lett*, vol. 88, p. 174102, 2002.

[27] A. Porta, G. Baselli, D. Liberati, N. Montano, C. Cogliati, T. Gnecchi-Ruscone, A. Malliani, and S. Cerutti, "Measuring regularity by means of a corrected conditional entropy in sympathetic outflow," *Biological Cybernetics*, vol. 78, no. 1, pp. 71–78, 1998. [Online]. Available: https://doi.org/10.1007/s004220050414

[28] M. Rostaghi and H. Azami, "Dispersion entropy: A measure for time-series analysis," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 610–614, 2016.

[29] W. Chen, Z. Wang, H. Xie, and W. Yu, "Characterization of surface emg signal based on fuzzy entropy," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 2, pp. 266–272, 2007.

[30] P. Li, C. Liu, K. Li, D. Zheng, C. Liu, and Y. Hou, "Assessing the complexity of short-term heartbeat interval series by distribution entropy," *Medical & biological engineering & computing*, vol. 53, pp. 77–87, 2015.

[31] H. Azami and J. Escudero, "Amplitude-aware permutation entropy: Illustration in spike detection and signal segmentation," *Computer Methods and Programs in Biomedicine*, vol. 128, pp. 40–51, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169260715301152

[32] C. Bandt, "A new kind of permutation entropy used to classify sleep stages from invisible eeg microstructure," *Entropy*, vol. 19, no. 5, 2017. [Online]. Available: https://www.mdpi.com/1099-4300/19/5/197

[33] M. Tao, K. Poskuviene, N. F. Alkayem, M. Cao, and M. Ragulskis, "Permutation entropy based on non-uniform embedding," *Entropy*, vol. 20, no. 8, 2018. [Online]. Available: https://www.mdpi.com/1099-4300/20/8/612

[34] H. Azami, L. Faes, J. Escudero, A. Humeau-Heurtier, and L. Silva, *Entropy Analysis of Univariate Biomedical Signals:Review and Comparison of Methods*, ser. Contemporary Mathematics and Its Applications: Monographs, Expositions and Lecture Notes. World Scientific, Oct. 2022, vol. 4, p. 233–286.

[35] K. Keller, A. M. Unakafov, and V. A. Unakafova, "On the relation of ks entropy and permutation entropy," *Physica D: Nonlinear Phenomena*, vol. 241, no. 18, pp. 1477–1481, 2012.

[36] S. M. Pincus, "Approximate entropy as a measure of system complexity." *Proceedings of the National Academy of Sciences*, vol. 88, no. 6, pp. 2297–2301, 1991. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.88.6.2297

[37] W. Xiong, L. Faes, and P. C. Ivanov, "Entropy measures, entropy estimators, and their performance in quantifying complex dynamics: Effects of artifacts, nonstationarity, and long-range correlations," *Phys. Rev. E*, vol. 95, p. 062114, Jun 2017. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.95.062114

[38] G. Manis, M. Aktaruzzaman, and R. Sassi, "Low computational cost for sample entropy," *Entropy*, vol. 20, no. 1, 2018, article 61.

[39] Y. Jiang, D. Mao, and Y. Xu, "A fast algorithm for computing sample entropy," *Advances in Adaptive Data Analysis*, vol. 3, no. 01n02, pp. 167–186, 2011.

[40] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[41] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[42] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[43] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 20, no. 2, pp. 215–232, 1958.

[44] T. H. Mäkikallio, H. V. Huikuri, U. Hintze, J. Videbæk, R. D. Mitrani, A. Castellanos, R. J. Myerburg, and M. Møller, "Fractal analysis and time- and frequency-domain measures of heart rate variability as predictors of mortality in patients with heart failure," *The American Journal of Cardiology*, vol. 87, no. 2, pp. 178–182, 2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0002914900013126

[45] S. Guzzetti, M. T. L. Rovere, G. D. Pinna, R. Maestri, E. Borroni, A. Porta, A. Mortara, and A. Malliani, "Different spectral components of 24h heart rate variability are related to different modes of death in chronic heart failure," *European Heart Journal*, vol. 26, no. 4, pp. 357–362, 12 2004. [Online]. Available: https://doi.org/10.1093/eurheartj/ehi067

[46] D.-Y. Lee and Y.-S. Choi, "Multiscale distribution entropy analysis of short-term heart rate variability," *Entropy*, vol. 20, no. 12, 2018. [Online]. Available: https://www.mdpi.com/1099-4300/20/12/952

[47] L. Hussain, I. A. Awan, W. Aziz, S. Saeed, A. Ali, F. Zeeshan, and K. S. Kwak, "Detecting congestive heart failure by extracting multimodal

features and employing machine learning techniques," *BioMed Research International*, vol. 2020, no. 1, p. 4281243, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2020/4281243

[48] Z. Masetic and A. Subasi, "Congestive heart failure detection using random forest classifier," *Computer Methods and Programs in Biomedicine*, vol. 130, pp. 54–64, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169260715303369

[49] Y. İşler and M. Kuntalp, "Combining classical hrv indices with wavelet entropy measures improves to performance in diagnosing congestive heart failure," *Computers in Biology and Medicine*, vol. 37, no. 10, pp. 1502–1510, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482507000285

[50] N. Elfadil and I. Ibrahim, "Self organizing neural network approach for identification of patients with congestive heart failure," in *2011 International Conference on Multimedia Computing and Systems*, 2011, pp. 1–6.