# A Retrospective Study on Classifying Gait Signals Using Entropy Measures

Wael Suliman[1], Vinayakumar Ravi[1], Tuan D. Pham[2]
[1]*Center for Artificial Intelligence, Prince Mohammad Bin Fahd University,* Khobar 31952, Saudi Arabia
[2]*Barts and The London School of Medicine and Dentistry, Queen Mary University of London,* London E1 2AD, United Kingdom

***Abstract*—The ability to differentiate sensor-induced physiological signals between healthy and diseased subjects is useful for developing an e-health system. Patients with neurodegenerative disorders are among those who can benefit from the use of e-health. Entropy methods have been utilized to quantify the complexity of such physiological signals for pattern classification. To date, these methods have been applied individually. In this retrospective study, several entropy methods are examined and used as feature extraction methods for machine learning to classify gait patterns in neurodegenerative diseases. Experimental results show that the combination of entropy methods and standard statistical measures performed much better than the individual measures for physiological pattern differentiation. Several machine learning models were also evaluated for learning on these features. This study also found that the one-dimensional convolutional network model trained with the combined features provided the most favorable results, where the best entropy measures depend on certain values for time delays and embedding dimensions.**

***Index Terms*—Physiological signals, entropy measures, machine learning, pattern classification.**

## I. Introduction

The advancements in wearable devices have made them promising to improve global health. A wearable device is a device equipped with sensors that can measure, process, or analyze one or several health indicators [1]. Such devices collect physiological data, and hence the user can monitor the health condition. Physiological data, such as heart rate, can be acquired using wearable devices. Physiological time series classification is an important task to assess and treat patients.

Neuroscience research has gained a comprehensive understanding of the important role of brain signal irregularity in its functions. This irregularity results from the interaction of individual neurons and their neural system [2]. Entropy methods are mathematical approaches that are known to quantify the irregularity in time series. Thus, entropy methods have been applied in medical research to describe insights into physiological conditions [3].

Neurodegenerative diseases (NDDs) are a group of conditions that are clinically and pathologically diverse [4]. The spread of NDDs is anticipated to rise with the increasing life expectancy [5]. Amyotrophic Lateral Sclerosis (ALS), Parkinson's Disease (PD), and Huntington's Disease (HD) among other NDDs are characterized by the accumulation of misfolded proteins into insoluble aggregates [4], [6]. Gait analysis assesses and treats individuals with NDDs that affect their walk. Gait analysis is based on temporal/spatial aspects and pressure measures, such as stride, stance, or swing intervals; Vertical Ground Reaction Force (VGRF) [7]. Gait signals analysis is useful to assess and treat individuals with NDDs that affect their walking [7].

There are few studies done on the VGRF signals. The goal of this study is not to promote entropy methods to outperform existing methods; however, the purpose is to study several entropy methods on physiological time series (NDDs is a case study) classification. Moreover, there are parameters that affect entropy methods, such as embedded dimension and time delay. In this study, entropy methods are the features of several machine learning models. Based on the classification accuracy, the results are discussed. As NDDs alter the shape, symmetry, and flatness of the VGRF signals, the common linear features are extracted from the time series. The classification results obtained from linear information are compared with those results obtained from complexity information. The result of combining both information to classify different NDDs is evaluated in this study.

The remaining sections of this paper are arranged as follows. Several entropy methods are described in section II. Section III describes the NDDs data set. Section IV presents the method used in the study. Results and discussion are addressed in section V. Finally, Section VI is the conclusion of the findings reported in this paper.

## II. Entropy Methods

Over the years, many entropy methods have been developed. Generally, entropy methods can be categorized into several groups: base entropy, cross-entropy, multiscale entropy, and bi-dimensional entropy [8]. In this study, several base entropy and multiscale entropy methods are considered.

### A. Approximate Entropy (ApEn)

ApEn is a well-known entropy method, and it is a logarithmic ratio of the probability that two patterns of the same embedded dimension $m$ are within the tolerance $r$. The calculation of ApEn depends on setting two parameters the embedded dimension and a tolerance level [9]. The equation is defined as follows:

$$ApEn(m, r, N) = ln\frac{C(m, r, N))}{C(m+1, r, N)} \quad (1)$$

where $N$ is the length of the time series data. The number of patterns of length $m$ that are within the tolerance $r$ in the time series data is defined as $C(m, r, N)$. While $C(m+1, r, N)$ is the number of patterns of length $m+1$ that are within the tolerance $r$ in the time series. A lower ApEn value indicates predictability in the data.

### B. Sample Entropy (SampEn)

SampEn is similar to ApEn in that it measures the likelihood that patterns of a certain length and tolerance will repeat in the time series. It is similar to the ApEn measure, yet with some modifications in the calculation [10]. The formula for SampEn is defined as follows:

$$SampEn(m, r, N) = -log\frac{C(m, r))}{C(m+1, r)} \tag{2}$$

where $C(m, r)$ is the number of patterns of length $m$ that are within the tolerance $r$ in the time series data, and $C(m+1, r)$ is the number of patterns of length $m+1$ that are within the tolerance $r$ in the time series data.

### C. Fuzzy Entropy (FuzzEn)

FuzzEn measures the complexity of a time series and considers the fuzziness in a time series. It is based on the concept of fuzzy sets, which allow for partial membership of an element in a set [11], [12]. A lower FuzzEn value indicates greater regularity in the time series. The formula for FuzzEn is defined as follows:

$$FuzzEn(m, r, N) = -log(\sum \frac{P(i, j))}{P(i)}) \tag{3}$$

where $i$ and $j$ are $1, 2, \ldots, N-m+1$. $P(i, j)$ is the similarity measure between two fuzzy patterns $i$ and $j$, which is defined as $exp(\frac{-d(i,j)^2}{r^2})$, where $d(i, j)$ is the distance between the fuzzy patterns $i$ and $j$. $P(i)$ is the probability of a fuzzy pattern $j$, which is calculated as the number of fuzzy patterns with a membership degree greater than or equal to a certain threshold divided by the total number of fuzzy patterns.

### D. Multiscale Sample Entropy (MSE)

MSE was initially developed to calculate SampEn of time series at multiple scales, then it got generalized to calculate any base entropy method by dividing the original time series into segments of different resolutions [13], [14]. This method reveals patterns of regularity in the time series that are not apparent at a single scale. The formula for calculating MSE is as follows:

$$MSE(m, r, N, S) = -log\frac{C(m, r, N, S))}{C(m, r, N, S+1)} \tag{4}$$

where $S$ is the scale at which the entropy is measured. $C(m, r, N, S)$ is the number of patterns of length $m$ that are within the tolerance $r$ in the time series at the given scale.

### E. Permutation Entropy (PE)

PE measures the complexity of a time series based on the ordinal patterns of the time series [15]. The basic idea behind PE is to convert the time series into a sequence of ordinal patterns by ranking the values in each embedded dimension of the time series. The resulting ordinal patterns are then counted and used to calculate the entropy of the time series. The formula for calculating PE is as follows:

$$PeEn(m) = -\sum p(i)log(p(i)) \tag{5}$$

where $i = 1, 2, \ldots, m!$ and $m!$ are the number of possible ordinal patterns of length $m$. While $p(i)$ is the probability of the $i^{th}$ ordinal pattern appearing in the data.

### F. Bubble Entropy (BubE)

BubEn is a measure of the complexity of a network or graph based on the distribution of bubbles or topological motifs in the network [16]. The advantage of this method is that it does not depend on parameters unlike other entropy methods [17]. The basic idea behind bubble entropy is to count the number of bubbles of different sizes or types in the network and use this information to calculate the entropy of the network. The resulting entropy value reflects the degree of randomness in the distribution of bubbles in the network. The formula for calculating bubble entropy is as follows:

$$BubEn = \frac{H_{swaps}^{m+1} - H_{swaps}^{m}}{log(\frac{m+1}{m-1})} \tag{6}$$

where $H_{swaps}^{m}$ is the second-order Renyi entropy of the probability mass function.

### G. Distribution Entropy (DisE)

DisEn is a measure of the randomness of a probability distribution. The basic idea behind distribution entropy is to calculate the entropy of a probability distribution by using the probabilities of the individual outcomes in the distribution [18]. The formula for calculating distribution entropy is as follows:

$$DisEn(m, M) = \frac{-\sum_{t=1}^{M} p_t log_2(p_t)}{log_2(M)} \tag{7}$$

where $M$ is the distribution bins $p_t$ is the probability of each distribution bin.

### H. Dispersion Entropy (DispE)

DispEn is an irregularity measure of a time series based on the distribution of its values. DispEn relies on the dispersion of the data points within a sliding window. The resulting entropy value reflects the degree of irregularity in the distribution of the data points [19]. The formula for calculating DispEn is as follows:

$$DispEn(m, c, d) = -\sum_{\pi=1}^{c^m} p(\pi_{v_i}).ln(p(\pi_{v_i})) \tag{8}$$

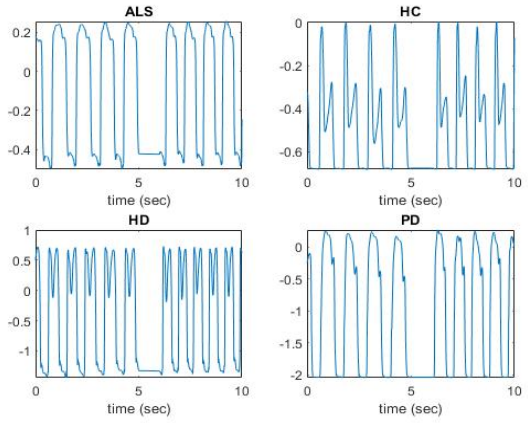
where $c$ is the number of classes, and $d$ is time delay. $\pi_{v_i}$ is dispersion pattern.

### I. Sample-Dependence Recurrent Sample Entropy (SDRSE)

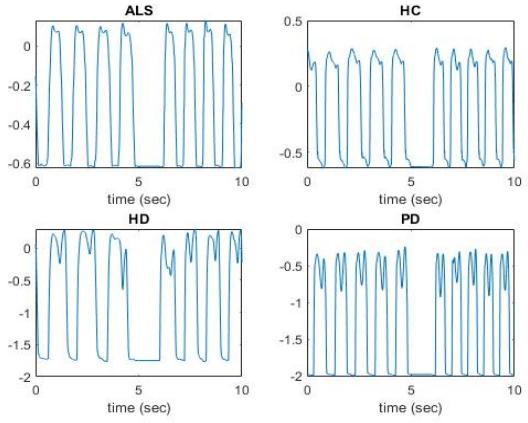
SDRSE captures the sequential order of the time series. The total number of matches with other points within a similarity tolerance of the point represented by the row or column of the recurrence plot (RP), where the diagonal element is removed is the sum of each row. Then the BLCM can be obtained from the binary image of RP, which quantifies the sample similarity according to spatial orientations [20]. The formula for calculating SDRSE is as follows:

$$SDRSE(m, \epsilon, \delta) = -log(\frac{s^{m+1}(\epsilon, \delta)}{s^{m}(\epsilon, \delta)}) \quad (9)$$

where the spatial offset is denoted as $\delta$, and the probability of the total number of the recurrence pairs is denoted as $S(\epsilon, \delta)$.




(a) male




(b) female

Fig. 1. Signals of 10 seconds recorded from different NDD subjects.

## III. Data Set

This study examined entropy methods on a publicly available gait data set, which consists of three NDDs. The NDDs' gait data was contributed by Hausdorff et al. [21]. The data set contains VGRF of gait recordings obtained from 15 patients with PD, 20 patients with HD, 13 patients with ALS, and 16 healthy control subjects (HC). Table I displays the demographic parameters (age, height, weight, speed, and severity) of the participants. The data were collected using force-sensitive resistors while all subjects walked at their own pace along a predetermined path (77-meter-long straight hallway). Each recording contained 5 minutes obtained from the foot at a sampling frequency of 300 Hz. Fig. 1 depicts ten seconds of VGRF signals from male and female participants of HC controls and patients with ALS, PD, and HD.

TABLE I
NDD DEMOGRAPHY

| Group | PD | HD | ALS | HC |
|---|---|---|---|---|
| **Age** | 66.8 ± 10.9 | 46.7 ± 12.6 | 55.6 ± 12.8 | 39.3 ± 18.5 |
| **Height** | 1.9 ± 0.2 | 1.8 ± 0.1 | 1.74 ± 0.1 | 1.74 ± 0.10 |
| **Weight** | 75.1 ± 16.9 | 72.1 ± 17.1 | 77.1 ± 21.1 | 66.8 ± 11.1 |
| **Speed** | 1.0 ± 0.2 | 1.2 ± 0.4 | 1.1 ± 0.2 | 1.4 ± 0.2 |
| **Severity** | 2.8 ± 0.9 | 6.9 ± 3.8 | 18.3 ± 17.8 | 0 ± 0 |

## IV. Experiment

The used data set will go through several stages in the study. Fig. 2 depicts the block diagram of the study. The stages are (1) Signal acquisition (2) signal preprocessing (3) feature extraction, and (4) classification. Each block of the study is explained in this section.

### A. Signal Preprocessing

The VGRF records the movement of the foot. Each record contains several "not number values". To tackle this issue, interpolation was used. In this study, non-overlap 1000 sample points'segmentation was used. After the segmentation, the number of learning samples has increased, thus the classifier has more information to learn from. Originally the number of samples was 64 with 90000 data points, while after the segmentation the data increased to 5760 samples (90 segments per record).

### B. Feature Extraction

Entropy methods were used to extract the nonlinearity in the sequence of movement and pattern of force fluctuations. Entropy is a complexity metric that is useful for non-stationary signals since it focuses on recurring patterns of variations in a time series. In this study, the aforementioned entropy methods were considered. In the study, two parameters, embedded dimensions (m = 2,3,4), and time delay (t = 1,2,3), were considered. Since NDDs change the gait signals characteristics, linear features like mean, variance, skewness, and kurtosis are extracted to reveal important information about the dispersion, symmetry, and flatness of VGRF signal distribution across time [22].

### C. Classifiers

In this study, supervised machine learning classifiers, such as the support vector machine (SVM) model, the long short-term memory (LSTM) network model, and the one-dimensional convolutional neural network (1D-CNN) model were used. The kernel considered in SVM is Radian Basis Function (RBF).

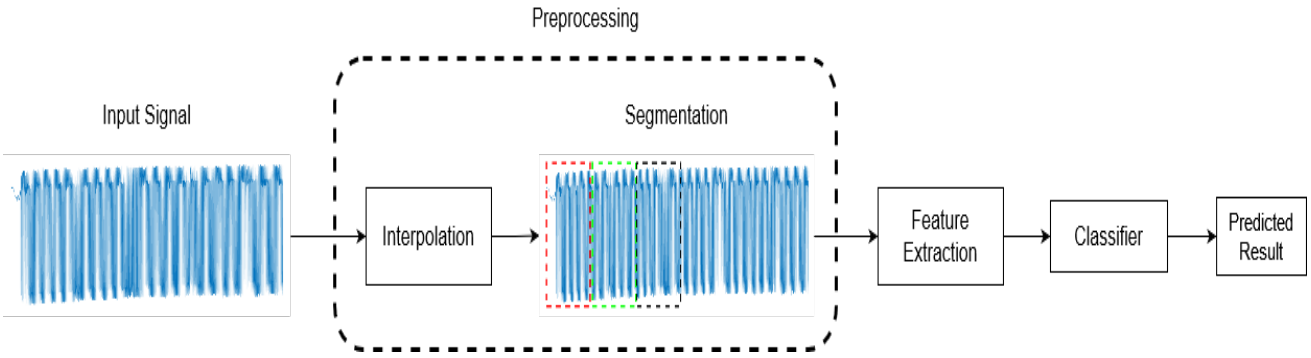



Fig. 2. Procedure for classification of physiological signals.

TABLE II
ACCURACY RESULTS OF USING ONLY ENTROPY MEASURES

| | SVM | | | LSTM | | | 1DCNN | | |
|---|---|---|---|---|---|---|---|---|---|
| **t = 1** | | | | | | | | | |
| | *m=2* | *m=3* | *m=4* | *m=2* | *m=3* | *m=4* | *m=2* | *m=3* | *m=4* |
| **Approximate** | 21.68±5.33 | 21.16±3.66 | 21.68±5.33 | 31.05±11.87 | 37.05±9.32 | 35.05±10.67 | 25.58±10.04 | 43.05±31.93 | 36.63±15.18 |
| **Sample** | 21.68±5.33 | 21.68±5.33 | 21.68±5.33 | 37.05±9.32 | 35.05±10.67 | 33.05±11.48 | 32.11±21.44 | 32.11±21.44 | 33.16±18.86 |
| **Fuzzy** | 20.63±2 | 21.68±5.33 | 22.21±6.99 | 37.05±9.32 | 35.05±10.67 | 31.05±11.87 | 49.68±14.42 | 46.11±18.82 | 44.11±18.23 |
| **Multiscale** | 20.11±0.33 | 18±6.32 | 21.68±5.33 | 33.05±11.48 | 31.05±11.87 | 31.05±11.87 | 35.58±18.83 | 26.11±16.43 | 33.16±18.86 |
| **Permutation** | 21.16±3.66 | 20.11±0.33 | 18.11±11.38 | 37.05±13.26 | 37.05±9.32 | 29.05±11.88 | 35.58±16.3 | 37.05±13.26 | 34.21±13.62 |
| **Bubble** | 17.58±6.32 | 20.21±9.93 | 16.11±8.49 | 31.05±15.16 | 37.05±13.26 | 27.05±11.51 | 34.11±21.11 | 36.63±15.18 | 39.58±19.39 |
| **Distribution** | 25.68±9.19 | 21.68±14.36 | 21.16±16.74 | 35.05±10.67 | 35.05±10.67 | 31.05±11.87 | 31.05±11.87 | 33.05±14.85 | 33.05±14.85 |
| **Dispersion** | 28.74±11.48 | 25.16±20.73 | 19.16±7.65 | 31.05±11.87 | 33.05±14.85 | 31.05±11.87 | 29.05±11.88 | 33.58±14.04 | 27.05±14.88 |
| **SDR** | 19.05±3 | 19.05±3 | 21.68±5.33 | 29.05±11.88 | 35.05±10.67 | 33.05±11.48 | 22.11±6.3 | 33.16±21.09 | 32.11±16.79 |
| **t = 2** | | | | | | | | | |
| | SVM | | | LSTM | | | 1DCNN | | |
| | *m=2* | *m=3* | *m=4* | *m=2* | *m=3* | *m=4* | *m=2* | *m=3* | *m=4* |
| **Approximate** | 21.68±5.33 | 21.16±3.66 | 22.21±6.99 | 33.05±14.85 | 35.05±10.67 | 33.05±11.48 | 49.58±20.19 | 41.68±14.84 | 37.26±19.1 |
| **Sample** | 20.11±0.33 | 20.63±2 | 21.68±5.33 | 35.05±10.67 | 33.05±11.48 | 35.05±10.67 | 26.63±18.86 | 33.58±21.54 | 41.58±15.5 |
| **Fuzzy** | 21.16±3.66 | 22.21±6.99 | 22.21±6.99 | 25.05±10.72 | 33.05±11.48 | 31.05±11.87 | 49.26±21.35 | 40.21±18.87 | 49.68±21.79 |
| **Multiscale** | 19.05±3 | 21.68±5.33 | 21.16±3.66 | 33.05±11.48 | 31.05±11.87 | 35.05±10.67 | 31.58±21.8 | 28.63±13.72 | 36.63±22.3 |
| **Permutation** | 19.05±3 | 21.68±5.33 | 19.05±3 | 31.05±11.87 | 33.05±11.48 | 35.05±10.67 | 33.58±14.04 | 38.11±14.62 | 29.16±9.97 |
| **Bubble** | 18.11±6.37 | 18.11±6.37 | 20.11±0.33 | 29.05±11.88 | 27.05±14.88 | 35.05±14.24 | 31.05±11.87 | 22.11±17.5 | 28.63±13.72 |
| **Distribution** | 29.68±13.93 | 19.68±15.93 | 17.16±16.51 | 35.05±10.67 | 35.05±10.67 | 33.05±11.48 | 27.05±14.88 | 31.05±11.87 | 32.53±12.68 |
| **Dispersion** | 26.21±13.76 | 25.68±13.17 | 25.16±12.77 | 35.05±10.67 | 33.05±11.48 | 29.05±11.88 | 23.05±13.32 | 25.05±17.11 | 29.05±11.88 |
| **SDR** | 21.68±5.33 | 19.05±3 | 21.68±5.33 | 35.05±10.67 | 37.05±9.32 | 33.05±11.48 | 29.68±19.28 | 32.63±18.99 | 37.16±19.97 |
| **t = 3** | | | | | | | | | |
| | SVM | | | LSTM | | | 1DCNN | | |
| | *m=2* | *m=3* | *m=4* | *m=2* | *m=3* | *m=4* | *m=2* | *m=3* | *m=4* |
| **Approximate** | 22.11±6.3 | 15.68±12.01 | 17.68±10.7 | 33.05±11.48 | 31.05±11.87 | 31.05±11.87 | 36.11±24.51 | 37.05±13.26 | 34.63±15.98 |
| **Sample** | 21.16±3.66 | 21.68±5.33 | 21.68±5.33 | 37.05±9.32 | 29.05±11.88 | 35.05±10.67 | 35.58±16.3 | 30.11±14.06 | 24.11±12.62 |
| **Fuzzy** | 24.74±10.14 | 25.68±13.17 | 28.63±9.97 | 35.05±10.67 | 31.05±15.16 | 29.05±15.17 | 34.21±9.83 | 42.63±21.39 | 38.63±16.89 |
| **Multiscale** | 19.05±3 | 19.05±3 | 21.16±3.66 | 35.05±10.67 | 33.05±11.48 | 35.05±10.67 | 40.11±13.16 | 38.11±23.86 | 34.11±16.37 |
| **Permutation** | 21.68±5.33 | 19.05±3 | 21.68±5.33 | 35.05±10.67 | 31.05±11.87 | 35.05±10.67 | 31.05±15.16 | 33.05±17.59 | 39.05±24.17 |
| **Bubble** | 17.58±6.32 | 18.63±6.84 | 18.11±6.37 | 29.05±17.86 | 31.05±11.87 | 31.05±15.16 | 32.63±18.99 | 26.11±16.43 | 29.05±20.19 |
| **Distribution** | 18.63±17.7 | 17.16±16.51 | 16.63±13.02 | 37.05±9.32 | 35.05±10.67 | 29.05±11.88 | 31.05±15.16 | 27.05±11.51 | 31.05±15.16 |
| **Dispersion** | 23.26±16.86 | 25.68±13.17 | 25.16±8.62 | 33.05±11.48 | 33.05±11.48 | 37.05±9.32 | 33.05±11.48 | 29.05±17.86 | 33.58±10.4 |
| **SDR** | 19.68±8.71 | 20.63±2 | 20.11±0.33 | 37.05±9.32 | 31.05±11.87 | 33.05±11.48 | 39.16±23.25 | 36.63±22.3 | 35.26±21.95 |

TABLE III
ACCURACY RESULTS OF STATISTICAL MEASURES ALONE

| | SVM | LSTM | 1DCNN |
|---|---|---|---|
| **Mean** | 38.63±19.35 | 27.05±14.88 | 39.68±24.96 |
| **STD** | 32.21±14.13 | 31.05±20.19 | 47.16±19.88 |
| **Skewness** | 24.11±12.62 | 37.58±15.38 | 35.16±20.65 |
| **Kurtosis** | 26.21±10.02 | 50.74±10.01 | 54.84±14.16 |

LSTM networks are a special case of Recurrent Neural Networks (RNN) and are designed to find patterns in time. In the study, the architecture is constructed with two bi-LSTM layers, a 20% dropout layer, a fully connected layer, and a softmax output layer. The build of 1D-CNN is the same as the conventional CNN, yet the size is one dimension. This study's architecture comprises numerous layers, including two convolutional layers, two dropout layers with a dropout rate of 0.2, one max pooling layer, a fully connected Multilayer Perceptron (MLP) layer, and a Softmax output layer. The

TABLE IV
ACCURACY RESULTS OF COMBINING BOTH INFORMATION

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **t = 1** | | | | | | | | | |
| | **SVM** | | | **LSTM** | | | **1D-CNN** | | |
| | *m=2* | *m=3* | *m=4* | *m=2* | *m=3* | *m=4* | *m=2* | *m=3* | *m=4* |
| **Approximate** | 49.68±14.42 | 37.68±14.74 | 43.68±18.48 | 46.74±13.34 | 55.16±16.91 | 46.21±9.54 | 65.89±13.38 | 72.84±13.42 | 60.32±18.88 |
| **Sample** | 53.68±19.26 | 39.16±21.25 | 53.16±21.96 | 50.74±10.01 | 49.79±13.99 | 51.26±9.95 | 70.84±19.14 | 74.84±18.46 | 62.32±23.94 |
| **Fuzzy** | 43.79±22.55 | 56.21±22.55 | 46.21±13.41 | 54.74±8.71 | 56.32±12.8 | 54.74±15.93 | 74.84±15.87 | 66.32±18.89 | 64.42±23.07 |
| **Multiscale** | 45.26±12.83 | 46.32±19.26 | 39.26±12.27 | 50.74±13.75 | 50.74±16.67 | 44.74±15.74 | 71.37±21.33 | 74.32±26.58 | 65.37±18.56 |
| **Permutation** | 38.74±24.13 | 37.68±19.88 | 37.26±28.45 | 50.21±16.87 | 57.26±6.49 | 46.74±18.86 | 71.79±14.2 | 65.37±15.98 | 58.32±11.46 |
| **Bubble** | 37.68±22 | 28.63±21.33 | 49.26±10.01 | 61.26±17.76 | 55.26±12.6 | 52.74±13.47 | 66.32±16.37 | 74.84±24.65 | 60.32±16.36 |
| **Distribution** | 46.21±13.41 | 44.21±24.56 | 37.68±22 | 47.68±10.64 | 56.21±12.37 | 46.74±16.33 | 67.89±10.2 | 72.42±17.14 | 59.79±24.95 |
| **Dispersion** | 47.79±30.02 | 46.21±21.13 | 44.21±15.73 | 48.74±13.71 | 47.16±14.75 | 41.79±14.48 | 66.32±21.11 | 62.84±17.61 | 64.32±12.58 |
| **SDR** | 44.63±27.43 | 46.74±29.82 | 44.74±20.63 | 54.21±18.81 | 50.74±13.75 | 48.74±9.95 | 67.37±16.48 | 71.89±13.92 | 66.74±15.3 |
| **t = 2** | | | | | | | | | |
| | **SVM** | | | **LSTM** | | | **1D-CNN** | | |
| | *m=2* | *m=3* | *m=4* | *m=2* | *m=3* | *m=4* | *m=2* | *m=3* | *m=4* |
| **Approximate** | 42.21±23.93 | 41.26±13.92 | 46.63±26.14 | 58.21±11 | 44.74±15.74 | 48.74±9.95 | 70.32±13.93 | 70.32±19.28 | 68.84±9.94 |
| **Sample** | 41.68±22.06 | 38.74±11.73 | 35.16±15.76 | 47.26±13.47 | 56.74±17.8 | 44.74±12.6 | 69.89±14.06 | 63.89±24.51 | 67.37±13.52 |
| **Fuzzy** | 45.68±28.45 | 50.21±23.48 | 48.21±16.77 | 54.21±13.27 | 48.21±10.17 | 56.74±11.8 | 66.84±18.86 | 65.37±20.81 | 63.37±15.18 |
| **Multiscale** | 48.74±25.15 | 47.79±21.38 | 48.63±20.66 | 50.74±10.01 | 44.74±18.35 | 50.74±10.01 | 66.84±13.34 | 71.89±13.92 | 71.26±19.96 |
| **Permutation** | 45.16±14.16 | 50.74±21.35 | 36.21±24.63 | 53.26±13.34 | 50.74±10.01 | 46.74±16.33 | 69.37±16.7 | 78.84±13.83 | 63.37±22.3 |
| **Bubble** | 35.26±19.82 | 36.74±22.97 | 51.68±21.72 | 40.11±16.19 | 50.74±10.01 | 50.74±13.75 | 67.37±13.52 | 68.84±13.7 | 67.37±16.48 |
| **Distribution** | 40.74±21.21 | 40.21±21.09 | 45.26±20.77 | 44.74±12.6 | 52.32±10.64 | 48.21±16.77 | 66.32±23.12 | 65.26±19.37 | 76.32±20.41 |
| **Dispersion** | 37.16±22.09 | 37.26±9.91 | 49.16±11.7 | 48.74±16.64 | 46.74±13.34 | 42.11±14.59 | 58.84±17.82 | 69.79±19.57 | 75.37±12.58 |
| **SDR** | 41.68±19.95 | 39.68±18.88 | 46.74±29.82 | 46.74±9.43 | 52.32±10.64 | 48.74±16.64 | 68.95±15.16 | 75.37±12.58 | 63.79±8.57 |
| **t = 3** | | | | | | | | | |
| | **SVM** | | | **LSTM** | | | **1D-CNN** | | |
| | *m=2* | *m=3* | *m=4* | *m=2* | *m=3* | *m=4* | *m=2* | *m=3* | *m=4* |
| **Approximate** | 41.68±19.95 | 48.21±23.41 | 47.68±27.12 | 42.11±17.37 | 56.21±12.37 | 50.74±13.75 | 65.79±16.56 | 74.32±13.17 | 66.84±26.67 |
| **Sample** | 41.68±14.84 | 44.63±25.76 | 45.26±18.51 | 46.74±13.34 | 46.74±13.34 | 52.21±19.19 | 67.37±26.76 | 76.84±20.1 | 66.84±13.34 |
| **Fuzzy** | 41.16±20.16 | 37.79±23.74 | 46.74±23.1 | 49.26±16.67 | 44.74±18.35 | 44.74±18.35 | 66.32±9.49 | 76.84±11.71 | 81.37±11.65 |
| **Multiscale** | 48.21±23.41 | 37.68±19.88 | 44.11±22.58 | 51.26±13.71 | 47.26±16.44 | 49.79±13.99 | 71.79±19.48 | 71.89±13.92 | 67.89±16.79 |
| **Permutation** | 49.68±17.23 | 33.26±20.29 | 37.68±14.74 | 57.26±11.45 | 39.68±9.48 | 50.74±13.75 | 62.84±17.61 | 68.74±24.49 | 72.42±19.56 |
| **Bubble** | 39.26±15.47 | 42.21±23.93 | 47.68±19.49 | 42.74±14.83 | 55.26±15.74 | 49.26±10.01 | 71.89±23.44 | 69.37±21.37 | 58.74±19.27 |
| **Distribution** | 35.68±12.58 | 47.79±23.37 | 44.21±22.67 | 48.74±16.64 | 46.21±9.54 | 50.74±13.75 | 69.37±25.19 | 77.79±20.11 | 69.37±16.7 |
| **Dispersion** | 39.16±21.25 | 43.16±18.77 | 44.74±15.74 | 43.26±11.8 | 54.74±12.83 | 46.74±16.33 | 64.32±15.72 | 71.89±19.28 | 72.32±9.96 |
| **SDR** | 42.74±17.57 | 46.21±21.13 | 46.21±18.91 | 48.74±9.95 | 48.74±19.12 | 55.16±14.04 | 63.89±22.63 | 64.84±12.63 | 69.89±10.43 |

optimizer algorithm used in the study is Adam with a mini-batch size of 32, and the learning rate is 0.01. For the validation stage, the data set was partitioned into 10 folds one fold is for testing, while the rest is for training.

## V. RESULTS AND DISCUSSION

In this section, the accuracy results of different features are discussed. Table II shows the accuracy results of the entropy methods as input to different classifiers. The results of the linear features are shown in Table III. The accuracy results of combining linear and nonlinear information are shown in Table IV.

Table II shows the results of different entropy methods with tuning two parameters (time delay and embedded dimension). Generally, entropy methods did not perform better across different classifiers. FuzzEn provided the highest accuracy when it is a feature to 1DCNN. There is no direct correlation between accuracy and entropy parameters. For example, time delay did not show a significant change in accuracy. When changing the time delay value, the accuracy did not significantly improve. SampEn showed a consistent performance across different sizes of embedded dimensions and the time delay. When comparing different classifiers, SVM showed the worst performance. This was not expected since SVM is known to perform well in classification tasks. Both LSTM and 1DCNN are known to handle time series classification, but the accuracy is low. It can be concluded that information of irregularity, such as entropy methods, cannot provide enough discriminatory information. Therefore, classifiers find difficulty differentiating NDDs. We compared the results obtained from entropy measures with statistical measures.

NDDs alter signal characteristics; therefore, studying linear features is appreciated. Table III provides the accuracy results of statistical measures. Moreover, using inexpensive features, such as kurtosis, which measures the peakedness of the time series, has shown better performance compared to entropy methods. Kurtosis performed better than other statistical measures with LSTM and 1DCNN classifiers. Moreover, using kurtosis as a feature with 1DCNN provided better performance with an accuracy of 54.84%. Overall, 1DCNN performed well with all statistical features. Statistical measures are computationally inexpensive; however, they did not provide enough information to classify different NDDs. Entropy methods depend on several parameters (embedded dimension and time delay), which makes it difficult to find the best criteria to quantify the irregularity in the time series. In contrast, statistical measures do not quantify the irregularity in the signals, instead, they measure the signal's characteristics. Based on the results from

Table II and Table III, NDDs affect the signal's characteristics and show a similar pattern across time. Table IV shows the accuracy performance when both measurement groups are combined.

Table IV shows the accuracy results of combining linear and nonlinear information. Generally, this combination showed a boost in performance. Overall, 1DCNN outperformed other classifiers when combining both pieces of information. 1DCNN showed accuracy above 60%. Again, there is no direct correlation between accuracy and entropy parameters. DispEn with statistical measures showed consistent performance when the time delay is 1. Again, changing the value of time delay did not show a significant effect on the performance. Tuning entropy parameters is crucial, yet the study did not show a difference in the performance. This is because the time series pattern among different NDDs is close to each other.

Based on the obtained results, the NDDs data set is challenging. Both features (linear and nonlinear) did not capture discriminate information to classify different NDDs. However, kurtosis performed better than entropy methods. This tells us that NDDs alter the shape of the VGRF signal, while the non-linearity of the time series is immune to the changes. The entropy methods rely on the embedded dimension, and choosing its value is crucial to capture the irregularity in the signal. However, some entropy methods like FuzzEn showed a consistent performance across different values of embedded dimension, yet the performance remains poor. Both ApEn and SampEn share similar computations, and thus, they perform similarly. This finding indicates that the information from both of them is redundant. Segmentation could be another reason for low performance. initially, the data set is limited and the number of subjects per disease is almost the same. However, the segmentation process increased the gap between classes, and hence the data set was imbalanced.

## VI. Conclusion

This study applied several entropy methods to quantify the irregularity in the VGRF signals of NDD patients and HC subjects. The irregularity information, alone, cannot differentiate between NDD patients and HC participants. NDDs change the signal's structure; therefore, four common statistical measures were selected. The study showed that by combining both structural and irregularity information, classifier accuracy improved. For future studies, we will investigate the severity classification of NDDs. It is believed that by having a larger database, machine learning algorithms will learn more then the performance will improve. Another limitation of the database is the imbalance between classes. The gap between classes increases after the segmentation stage.

## References


[1] M. Tahri Sqalli and D. Al-Thani, "Evolution of wearable devices in health coaching: Challenges and opportunities," *Frontiers in Digital Health*, vol. 2, p. 545646, 2020.

[2] S. Keshmiri, "Entropy and the brain: An overview," *Entropy*, vol. 22, no. 9, p. 917, 2020.

[3] L. C. Amarantidis and D. Abásolo, "Interpretation of entropy algorithms in the context of biomedical signal analysis and their application to eeg analysis in epilepsy," *Entropy*, vol. 21, no. 9, p. 840, 2019.

[4] C. Peng, J. Q. Trojanowski, and V. M.-Y. Lee, "Protein transmission in neurodegenerative disease," *Nature Reviews Neurology*, vol. 16, no. 4, pp. 199–212, 2020.

[5] O. Hansson, "Biomarkers for neurodegenerative diseases," *Nature medicine*, vol. 27, no. 6, pp. 954–963, 2021.

[6] L.-N. Schaffert and W. G. Carter, "Do post-translational modifications influence protein aggregation in neurodegenerative diseases: a systematic review," *Brain sciences*, vol. 10, no. 4, p. 232, 2020.

[7] F. Setiawan, A.-B. Liu, and C.-W. Lin, "Development of neurodegenerative diseases' gait classification algorithm using convolutional neural network and wavelet coherence spectrogram of gait synchronization," *IEEE Access*, vol. 10, pp. 38 137–38 153, 2022.

[8] M. W. Flood and B. Grimm, "Entropyhub: An open-source toolkit for entropic time series analysis," *PloS one*, vol. 16, no. 11, p. e0259448, 2021.

[9] 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.

[10] 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*, 2000.

[11] 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.

[12] H.-B. Xie, W.-X. He, and H. Liu, "Measuring time series regularity using nonlinear similarity-based sample entropy," *Physics Letters A*, vol. 372, no. 48, pp. 7140–7146, 2008.

[13] M. Costa, A. L. Goldberger, and C.-K. Peng, "Multiscale entropy analysis of complex physiologic time series," *Physical review letters*, vol. 89, no. 6, p. 068102, 2002.

[14] M. D. Costa and A. L. Goldberger, "Generalized multiscale entropy analysis: Application to quantifying the complex volatility of human heartbeat time series," *Entropy*, vol. 17, no. 3, pp. 1197–1203, 2015.

[15] C. Bandt and B. Pompe, "Permutation entropy: a natural complexity measure for time series," *Physical review letters*, vol. 88, no. 17, p. 174102, 2002.

[16] G. Manis, M. Aktaruzzaman, and R. Sassi, "Bubble entropy: An entropy almost free of parameters," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 11, pp. 2711–2718, 2017.

[17] G. Manis, M. Bodini, M. W. Rivolta, and R. Sassi, "A two-steps-ahead estimator for bubble entropy," *Entropy*, vol. 23, no. 6, 2021. [Online]. Available: https://www.mdpi.com/1099-4300/23/6/761

[18] 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.

[19] 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.

[20] T. D. Pham and H. Yan, "Spatial-dependence recurrence sample entropy," *Physica A: Statistical Mechanics and its Applications*, vol. 494, pp. 581–590, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437117312591

[21] J. Hausdorff, "Lertratanakul a, cudkowicz me, peterson al, kaliton d, and goldberger al," *Dynamic markers of altered gait rhythm in amyotrophic lateral sclerosis. J Appl Physiol*, vol. 88, pp. 2045–2053, 2000.

[22] M. Chakraborty, T. Das, and D. Ghosh, "Characterization of gait dynamics using fractal analysis for normal and parkinson disease patients," in *2015 IEEE Power, Communication and Information Technology Conference (PCITC)*. IEEE, 2015, pp. 367–372.