


# A method to measure data complexity of a complicated medical data set

Martti Juhola<sup>1</sup>  | Henry Joutsijoki<sup>1</sup> | Kirsi Penttinen<sup>2</sup> | Disheet Shah<sup>3</sup> | Katriina Aalto-Setälä<sup>2,4</sup>

<sup>1</sup>Faculty of Information Technology and Communication Sciences, Tampere University, Tampere, Finland

<sup>2</sup>Faculty of Medicine and Health Technology, Tampere University, Tampere, Finland

<sup>3</sup>Department of Pharmacology, Northwestern University, Chicago, Illinois, USA

<sup>4</sup>Heart Center, Tampere University Hospital, Tampere, Finland

## Correspondence

Martti Juhola, Faculty of Information Technology and Communication Sciences, Tampere University, 33014 Tampere, Finland.

Email: [martti.juhola@tuni.fi](mailto:martti.juhola@tuni.fi)

## Funding information

Academy of Finland

## Abstract

In this article, we consider data complexity in the context of calcium transient signal data collected from induced pluripotent stem cell-derived cardiomyocytes. We present a novel way to measure data complexity based on the nearest neighbour searching method. Data complexity here is seen as overlapping and mixed data classes in addition to a relatively great number of data cases. Complexity affects classification results, which were run with nearest neighbour searching, feedforward artificial neural networks and random forests for seven genetic cardiological disease classes and healthy controls. The data are obtained from individuals carrying mutations for genetic cardiac diseases with induced pluripotent stem cell (iPSC) technology and the diseases include hypertrophic cardiomyopathy with two different founder gene mutations, dilated cardiomyopathy, long QT syndrome type 1 and 2, Brugada syndrome, a severe genetic ventricular arrhythmia (CPVT) and healthy controls. The data are from calcium transients from spontaneously beating iPSC-derived cardiomyocytes cultured in a biotechnology laboratory. When the genotype of the iPSC-derived cardiomyocytes is the same as the donor of the tissue sample and based on the characteristics of the calcium transients, it was possible to classify the seven diseases and healthy controls with machine learning. Peak data first detected before actual pre-processing from calcium transient signals corresponded to beats (repeating excitation–contraction coupling) of induced stem cell-derived cardiomyocytes and formed the basis of classification. During pre-processing of the calcium transient signals, we found that such techniques among others as even strong outlier cleaning or class size balancing by generating artificial cases improved only slightly or not at all classification accuracies. Therefore, the current data set was sufficiently complicated for our data complexity study. Random forests produced the best classification accuracies, 68% for all eight classes.

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2022 The Authors. *International Journal of Imaging Systems and Technology* published by Wiley Periodicals LLC.

**KEYWORDS**

calcium transient signals, classification, data complexity, genetic cardiac cardiomyocytes, induced pluripotent stem cell-derived cardiomyocytes, machine learning

## 1 | INTRODUCTION

Subject to data complexity, we understand here how data items or points are distributed in the attribute space of a data set. Different data classes may be quite well separable from each other at their best or rather overlapping and even mixed in difficult situations. In this sense, data complexity affects so that the more complex, the lower classification results are probably obtained while executing classification tasks with machine learning methods. Nevertheless, we may improve classification results by using appropriate pre-processing techniques. We studied these questions in the present article.

Since our first research articles<sup>1–3</sup> with induced pluripotent stem cell-derived cardiomyocyte (iPSC-CMs) data, that is, data extracted from calcium transient signals, we extended our data collection comprising seven genetic cardiologic diseases and controls. Machine learning can be used to classify calcium transient signals by using their peaks as data.<sup>2,3</sup>

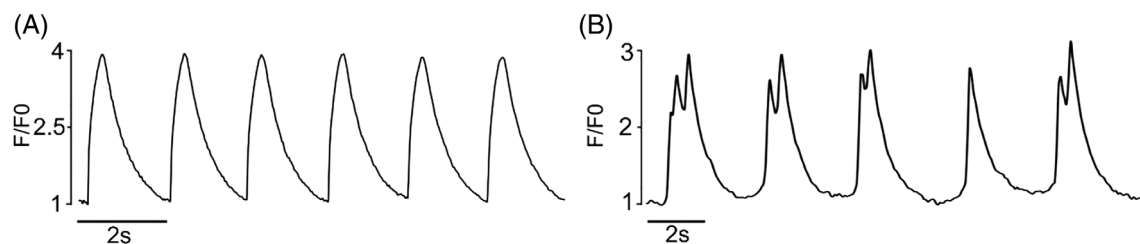
The function of calcium cycling is central in excitation–contraction coupling of cardiomyocytes. Abnormal calcium cycling is connected to arrhythmia associated with cardiac disorders and the study of cardiomyocyte calcium cycling offers tool to study cardiac functionality and diseases. Patient-specific iPSC-derived cardiomyocytes are used to study genetic cardiac diseases, and these include, for example, long QT syndrome 1 and 2 (LQT1 and LQT2),<sup>4–6</sup> electric disorders of the heart that predisposes patients to arrhythmias and sudden cardiac death,<sup>7</sup> dilated cardiomyopathy (DCM),<sup>8</sup> a disease of the heart muscle, hypertrophic cardiomyopathy (HCM),<sup>9,10</sup> disorder that affects the structure of heart muscle tissue leading to arrhythmias and progressive heart failure, Brugada syndrome (BrS) that predisposes patients to fatal cardiac arrhythmias,<sup>11,12</sup> and catecholaminergic polymorphic tachycardia (CPVT), an exercise-induced malignant arrhythmogenic disorder.<sup>13–15</sup> In addition, data of controls (wild type, WT) were included in the study.

We were interested in studying the complexity of data and particularly in the context of our current calcium transient signal data being clearly complicated while containing eight classes and varying data. Data complexity depends on properties of data sets such as numbers of data items, attributes, distributions of values of different attributes, classes, possible imbalance of classes and possible outliers or

missing values. Different data complexity measures<sup>16</sup> are given as follows. Overlaps in attribute values of different classes can be measured with maximum Fisher's discriminant ratio applying the squared difference of means of two classes divided by the sum of their variances, volume of overlap region by using minima and maxima of two classes of each attribute for computing the length of overlap region, and maximal attribute efficiency showing how much every attribute affects the separation of two classes. Another approach is to measure separability of classes<sup>16</sup> where there are (1) fraction of points on class boundary, (2) ratio of average intra-/inter-class nearest neighbour distance and (3) error rate of 1-nearest neighbour classifier. For these techniques, in (1), a class-independent minimum spanning tree is computed for a data set counting the number of items incident to a boundary between two classes. The tree connects all items to their nearest neighbours. Fractions incident to different classes versus all items are measured values. In (2), first, the Euclidean distance is computed from every item to its nearest neighbour. The average of all distances to intra-class nearest neighbours and the average of all distances to inter-class nearest neighbours are calculated. The ratio of two averages is the measured value, which compares the intra-class dispersion with the gap between classes. Measure in (3) is the error rate of a nearest neighbour classifier subject with the training set used according to leave-one-out. The measure indicates how close the items of different classes are. Furthermore, data complexity<sup>17</sup> was first approached theoretically (e.g., minimization) for binary classification and then different two-dimensional example distributions were presented by using one to three nested circles or other curved areas dividing the space into varying parts so that several parts could be given to the same class. An instance level analysis was studied, particularly considering such instances that are frequently misclassified.<sup>18</sup>

## 2 | DATA

The research was approved by the Ethics Committee of Pirkanmaa Hospital District as to culturing and differentiation of human iPSC lines (R08070). Patient-specific iPSC lines were established and cultured as previously described.<sup>2</sup> iPSC lines used in this study were derived from two LQT1 and two LQT2 patients, two HCM patients carrying a mutation in  $\alpha$ -tropomyosin gene



**FIGURE 1** (A) A normal BrS calcium transient signal with regular peak shapes and sizes. (B) An abnormal BrS transient signal containing multiple peaks.

(HCMT) and two HCM patients carrying a mutation in myosin binding protein C gene (HCM) patients, six CPVT patients carrying mutations in ryanodine receptor 2 gene for CPVT, one BrS patient carrying a mutation in SCN5A gene, two DCM patients carrying a mutation lamin A/C gene, and two healthy control individuals (WT). The studied iPSC lines were UTA.05605.CPVT, UTA.05208.CPVT, UTA.07001.CPVT, UTA.03701.CPVT, UTA.05503.CPVT and UTA.05404.CPVT generated from CPVT patients carrying cardiac ryanodine receptor (RyR2) mutations; UTA.14004.SCN5A generated from BrS patient carrying SCN5A mutation, UTA.07801.HCMM, and UTA.06108.HCMM generated from HCM patients carrying myosin-binding protein C (MYBPC3) mutations and UTA.02912.HCMT and UTA.13602.HCMT generated from HCM patients carrying  $\alpha$ -tropomyosin (TPM1); and UTA.00208.LQT1 and UTA.00118.LQT1 generated from LQT1 patients carrying potassium voltage-gated channel subfamily Q member1 (KCNQ1) mutation; UTA.03412.LQT2, UTA 03417.LQT2, UTA.03809.LQT2 and UTA.03810.LQT2 generated from LQT2 patients carrying the human ether-a-go-go-related gene (HERG) mutation; UTA.12619.LMNA and UTA.12704.LMNA generated from DCM patients with lamin A and lamin C (LMNA) mutations and UTA.04602.WT and UTA.04511.WT generated from healthy control individuals. These iPSCs were differentiated into spontaneously beating cardiomyocytes and dissociated as single cells for calcium imaging studies, in which cardiomyocytes were loaded with calcium indicators Fura-2 AM (Invitrogen, Molecular Probes) or Fluo-4 AM (Thermo Fisher Scientific) as described earlier.<sup>4,6,8,9,13</sup> Calcium transient signals were recorded from spontaneously beating cardiomyocytes and background noise was subtracted before further calcium analysis processing. A signal was determined to be abnormal if one or more of its peaks were determined as abnormal. A biotechnological expert had evaluated every entire signal to be either abnormal or normal to guarantee them as surely annotated as possible even if the algorithm

developed<sup>2</sup> could have been used for this purpose. See example signals in Figure 1.

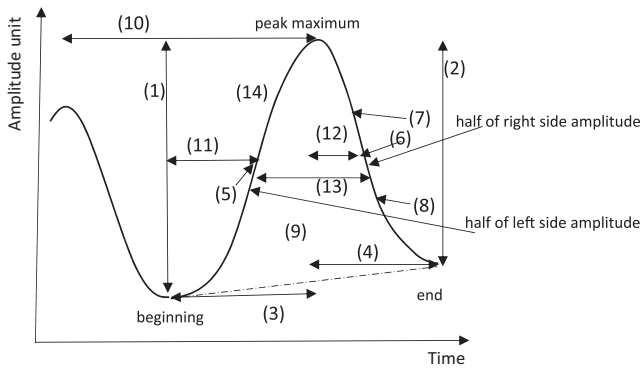
To detect peaks from signals, the first derivative of the signal was computed to detect the beginning, maximum and end of every peak. In those three locations the first derivative values are close to zero, but elsewhere mainly greater or less than zero. After the detection of potential peaks from a signal, small peaks less than 8% of an estimated average peak amplitude of the entire signal were removed and evaluated to be probable noise.<sup>2,3</sup> The minimum number of valid peaks was 1, the maximum 123 and the average 17.

The data set was comprised of seven disease classes and controls. The total of the calcium transient signals was 1393: 233 in disease CPVT, 69 in DCM, 270 in HCMM, 149 in HCMT, 90 in LQT1, 138 in LQT2, 218 BrS and 226 in controls WT. Respectively, the total of accepted peaks was 23 720: 2279 acceptable peaks were found in CPVT, 1169 in DCM, 4416 in HCMM, 2128 in HCMT, 1617 in LQT1, 3712 in LQT2, and 5577 in BrS and 2822 in WT signals. Compared with our recent study,<sup>3</sup> diseases DCM, LQT2 and BrS and a part of WT signals were new.

Next, values of suitable peak attributes from the accepted peaks were computed for the classification of peaks and finally signals into different classes. The attributes named in Figure 2 were computed for every accepted peak.

### 3 | METHODS

After having increased the number of diseases to seven and the class of controls of the current data set, we noticed that their classification was more difficult compared with three diseases only and controls of our earlier study.<sup>2</sup> The only pre-processing techniques that could improve classification accuracies a little (~0%–2%) depending on ways to build models and test these were standardization (zscore in Matlab) and weighting



**FIGURE 2** Peak attributes: (1) left and (2) right amplitudes, (3) left and (4) right durations, approximate location for (5) first derivative maximum of the left side, location for (6) absolute first derivative minimum, locations for (7) second derivative absolute minimum and (8) second derivative maximum, (9) surface area bounded by the peak curve and line from the peak beginning to the end, (10) time interval from the maximum of the preceding peak to the current one, (11) duration from the peak beginning to the first derivative maximum, (12) duration from the peak maximum to the first derivative absolute minimum, (13) mean peak duration computed from the mean of the left and right amplitudes, and (14) peak curve length

attributes with weight values computed with Relief algorithm.<sup>19</sup> Standardization changes the values of all attributes to an approximately same interval. This is important for such machine learning methods as nearest neighbour searching that use distance measures. In the current research all programming was made with Matlab.

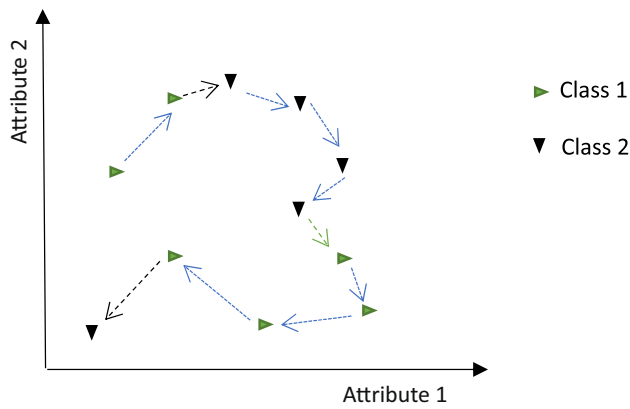
Using mainly nearest neighbour searching, we experimented with several pre-processing techniques beginning from data cleaning of training data by modifying the algorithm<sup>20</sup> that was purposed to outlier recognition so that it was effective to clean “boundary areas” of data classes. Nevertheless, this did not affect positively, although we iterated cleaning one time or even a few times so that ultimately approximately a half of all peak data were left out. We ran the common 10-fold cross-validation with nearest neighbour searching and then with artificial feedforward neural networks for the original data and then weighted with Relief algorithm. In 10-fold cross-validation, all data signals are divided randomly into 10 subsets of the approximately same size and one by one each subset is the current test set and the other 9 subsets jointly form the corresponding training set. Neural networks were also run based on one-versus-all (OVA) and one-versus-one (OVO) principles, that is, by dividing training data according to eight classes and operating with these separate parts of the data. After one cleaning iteration, we still classified by using neural

networks and according to OVO principle. Lastly, in OVO, we experimented with our novel idea of “predictive cleaning” where we used misclassified peaks of training sets and searched for 1-nearest neighbours in the test sets and removed these. Since various cleaning approaches did not improve classification results, next we applied Synthetic Minority Over-sampling Technique (SMOTE) algorithm,<sup>21</sup> with preceding cleaning as described above or without cleaning, to balance small and large data classes by generating artificial data items into small classes less than roughly a half of the largest of BrS class: CPVT, DCM, HCMT and WT. After SMOTE runs classifications were performed with nearest neighbour searching. Neither did the class balancing improve classification results. All those runs described were executed according to 10-fold cross-validation. All nearest neighbour runs were executed with Mahalanobis measure, because this produced better results than Euclidean, cityblock, correlation, cosine, Hamming, Jaccard, or Spearman measures.

After having noticed in the foregoing pre-processing experiments that the peak data of these calcium transient signals are relatively complicated while consisting of data items from seven different disease classes and controls as the eighth class, we designed a technique to measure the complexity of data. The technique is based on nearest neighbour searching that is utilized in several data mining algorithms, for example, SMOTE and Relief mentioned above. First, the path of nearest neighbours is computed so that starting from a randomly chosen peak (data item) of the whole data, the process searches for its nearest neighbour, which is not yet visited. The found one is assigned to be the next neighbour of the path. This is continued until the last unvisited peak is chosen, see Figure 3. Thereafter, for every class it is counted along with the neighbour path how many times the class label changes from the current class to some other class. Then for the whole peak data it is counted along with the path how many times a class changes to some other class. Ultimately, the classwise numbers of class changes are related to the sizes of classes and the latter class changes quantity is related to the size of the entire data set. The algorithm is given in Figure 4.

If there are relatively few changes from one class to another, complexity is low, close to 0. If there are frequent changes, complexity is clearly higher than in situations of infrequent class changes. Thus, data complexity in this context measures how overlapping or mixed the classes of the data set are.

It is necessary to repeat the process, for example, at least 10 times and average the complexity values computed, because the choice of the starting item affects somewhat how neighbour path is constructed. For our later tests, we first standardized the values of each data



**FIGURE 3** A simple hypothetical nearest neighbour path (in reality, the dimension of 14 attributes and thousands of data items originating from also up to eight classes) where two black dashed arrows indicate the changes from Class 1 to Class 2 and one green dashed arrow indicates the change from Class 2 to Class 1, but the blue dotted arrows show no class changes. Altogether, three changes were encountered.

Algorithm for data complexity evaluation:

- (1) Choose random starting item  $s$  from among  $n$  data items.
- (2) Repeat  $n - 1$  times:
  - Search for the nearest unvisited neighbor.
  - Set the found neighbor to the path of nearest neighbors.
- (3) Repeat for every class  $i$ :
  - (3.1) Start from the beginning  $s$  of the neighbor path.
  - (3.2) Set the number of changes  $c_i = 0$ .
  - (3.3) Repeat  $n - 1$  times:
    - Choose the next neighbor from the neighbor path.
    - if the preceding neighbor originates from the class  $i$  and the current neighbor from some other class then
    - $c_i = c_i + 1$ .
  - (3.4) For class  $i$  set complexity value
 
$$cv_i = \frac{c_i}{|s_i|}$$
 where  $|s_i|$  is the size of class  $i$ .
- (4) Start from item  $s$  of the neighbor path.
- (5) Set the number of changes  $d = 0$ .
- (6) Repeat  $n-1$  times:
  - Choose the next neighbor from the neighbor path.
  - if the class of the preceding neighbor differs from that of the current one then
  - $d = d + 1$ .
- (7) For the whole data set complexity value is as follows.
 
$$CV = \frac{d}{n}$$

**FIGURE 4** Algorithm for the computation of proposed data complexity

attribute applying zscore and then computed weights for the attributes by using Relief function.

The minimum complexity value of a class in Equation (1) would be given by one change only after having found first all items  $C_i$  of the class  $i$  one after another and only after that the change to some other class. Here  $|s_i|$  is the size of class  $i$ .

$$cv_i^{\min} = \frac{1}{|s_i|}. \tag{1}$$

However, if the class contained all the last neighbours of the neighbour path, the minimum would be equal to 0. The maximum would be obtained if after every item of the current class an item of some other class were encountered. Thus, the maximum complexity value would be 1. This has also the exception in Equation (2) for the last element of the neighbour path, which has no successor and gives the following when  $l$  is the last class of the neighbour path.

$$cv_l^{\max} = \frac{|c_l - 1|}{|c_l|}. \tag{2}$$

Minimum complexity value  $CV^{\min}$  of the whole data in Equation (3) is obtained when the neighbour path is comprised of unbroken parts, each of these representing one class only. Then there is a queue of the united classes, but after the last neighbour of the last class there is no comparison.  $C$  is the number of classes.

$$CV^{\min} = \frac{C - 1}{n}. \tag{3}$$

Since there are  $n - 1$  comparisons for  $n$  data items in the neighbour path, Equation (4) gives the maximum.

$$CV^{\max} = \frac{n - 1}{n}. \tag{4}$$

Naturally, these minima and maxima are hardly met in real-world data, but they are the limits for these complexity values.

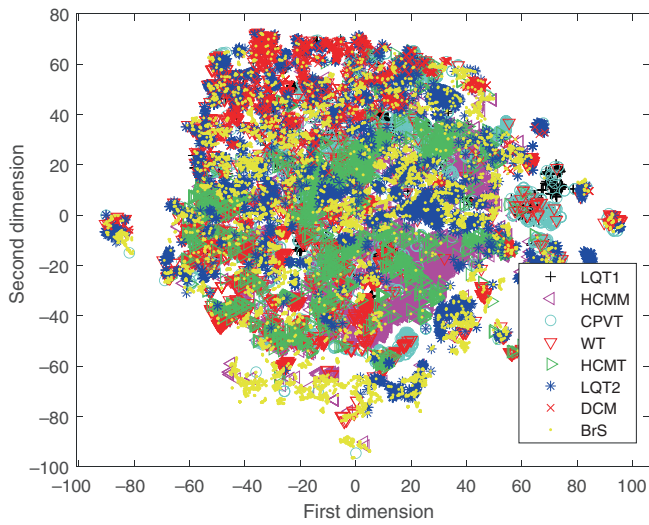
## 4 | RESULTS

In Table 1, there are complexity values computed from 2, 4, 6 and 8 classes that were comprised of 4950, 10 941, 16 526 and 23 720 calcium transient signal peaks, respectively. In Figures 5–8, there are visualizations computed from the whole data and its subsets of 6, 4 and 2 classes. Reducing two classes stepwise simplifies typically the classification problem when the attribute space areas or volumes of those classes still included become more separable from each other.

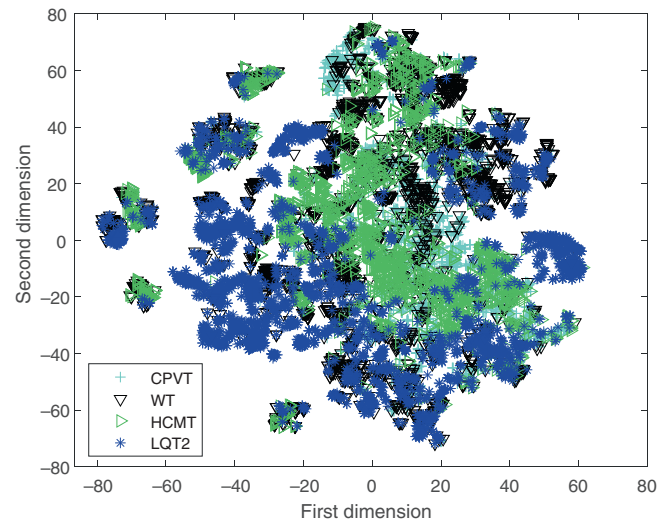
Machine learning modelling and tests with models were executed according to 10-fold cross-validation. The same cross-validation training and test folds were used for different methods and test set-ups, but while using less than all eight classes, those items representing classes left

**TABLE 1** Complexity values computed from the data of 2, 4, 6 and 8 classes when averages of 10 repetitions of complexity value computations were computed

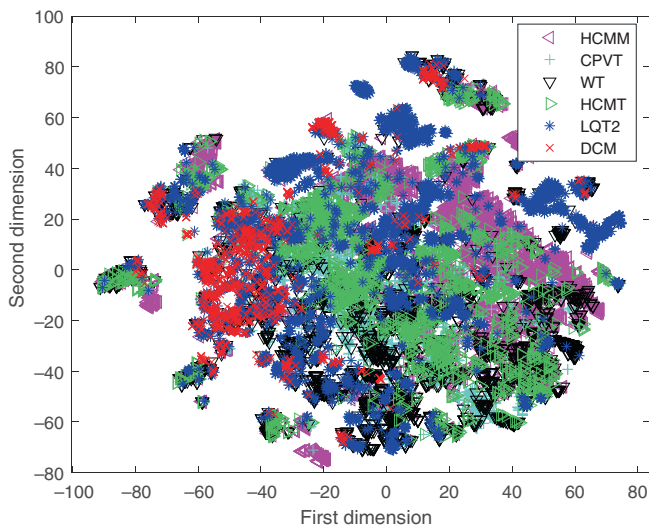
Number of classes	Classes								
	LQT1	HCMM	CPVT	WT	HCMT	LQT2	DCM	BrS	All classes
8	0.497	0.425	0.534	0.550	0.522	0.407	0.518	0.336	0.445
6		0.383	0.475	0.523	0.535	0.367	0.515		0.445
4			0.385	0.419	0.386	0.248			0.347
2				0.208	0.276				0.237



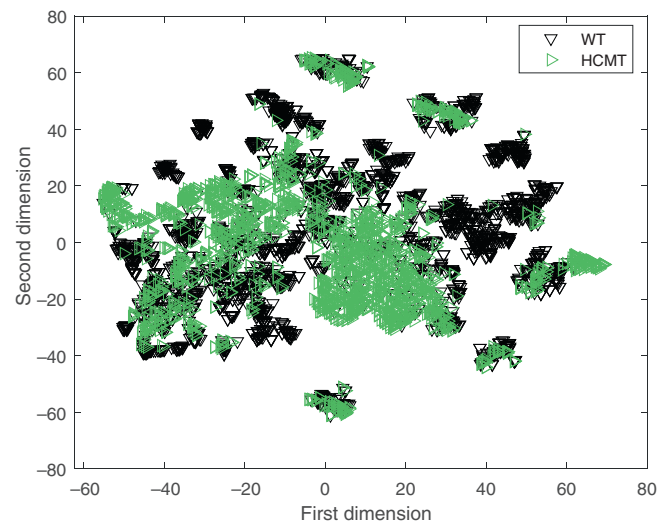
**FIGURE 5** Visualization computed with t-Distributed Stochastic Neighbour Embedding (t-SNE) algorithm in Matlab for all eight classes



**FIGURE 7** Visualization computed with t-SNE algorithm for four classes



**FIGURE 6** Visualization computed with t-SNE algorithm for six classes



**FIGURE 8** Visualization computed with t-SNE algorithm for two classes

**TABLE 2** Results of  $k$ -nearest neighbour searching classification: (A) Data of eight classes (standardized or not standardized with zscore and) classified with Mahalanobis distance, (B) data of eight classes when attributes were weighted with Relief algorithm and ties inside signals were solved with the squared inverses of their distances from nearest neighbours in different classes, (C) data of six classes classified with Mahalanobis distance similarly to those of eight classes, (D) then data of four classes classified similarly and (E) data of two classes classified similarly

Mode of classification	$k$	Average sensitivity (true positive rate) %								Accuracy %
		LQT1	HCMM	CPVT	WT	HCMT	LQT2	DCM	BrS	Mean and std
(a)	1	85.6	89.3	55.0	49.2	53.8	59.9	44.5	60.9	63.5 ± 4.1
	3	86.7	91.9	52.0	45.1	41.6	53.5	41.9	53.1	59.6 ± 2.8
	5	87.8	90.0	53.7	44.7	45.6	55.6	42.9	52.6	60.1 ± 3.9
(b)	1	84.4	87.0	56.8	48.7	59.1	59.2	46.0	64.2	64.2 ± 3.4
(c)	1		86.7	70.9	53.2	59.8	70.1	53.1		68.4 ± 3.7
(d)	1			73.5	60.3	73.9	81.2			70.9 ± 3.3
(e)	1				87.6	78.6				84.0 ± 4.8

out were removed from folds. The purpose was to make different test set-ups as comparable as possible. However, for artificial neural networks, five runs were executed for the whole cross-validation process (with the fixed folds), because feedforward neural networks use randomly selected initial values for weight values for connections of learning neurons.

It is noticeable to remember the character of classifying calcium transient signal data that the classification task is performed for every peak of signal, and in the present data, a signal may contain from 1 to 123 peaks. After having classified all peaks of a signal, the majority class of those peaks determines the class of the entire signal. Nonetheless, a tie as to the majority is then possible. Thus, for the nearest neighbour searching method, we ran first the tests without taking ties into consideration and second taking ties into consideration.

In Table 2, there are results computed with  $k$ -nearest neighbour searching method and Mahalanobis distance measure that produced better results than those with Euclidean, cityblock, correlation, Spearman, cosine, Hamming or Jaccard distance measures. In the beginning, the data items were also standardized with zscore function attribute by attribute since this was also used for other tests. Nevertheless, while using Mahalanobis distance measure, standardization would not have been necessary, when the covariance calculation of Mahalanobis measure has the similar effect to classification results. The data items were also weighted by coefficients computed with Relief algorithm. When other distance measures than Mahalanobis were applied, standardization pre-processing may be more or less useful depending on data. For nearest neighbours, number  $k$  searched for from a training set, we used values 1, 3, 5, 7, 9 and 11, but when  $k$  equal to 1 always gave the best result, these are only presented after Table 2 part (a). In Table 2 part (a), ties in nearest neighbour searching

inside signals were not considered, but they were considered in parts (b)–(e). If there was a tie, that is, equally many nearest neighbours from two or more classes subject to all peaks of a test signal, the following computation was performed. A weight was computed for every test peak  $i$  of the test signal as to its  $j$ th nearest neighbour

$$w_{ij} = \frac{1}{d_{ij}^2}, i = 1, \dots, p, j = 1, \dots, k, \tag{5}$$

where  $d_{ij}$  is the distance value from a test peak  $i$  to its  $j$ th nearest neighbour in the training set. Let  $C(i, j)$  be the class of neighbour  $j$ . Vector  $V$  containing  $g$  components here according to the number of eight classes of the present data is used to store the sums for classes predicted by nearest neighbours of test case  $i$ , which is repeated for every peak  $i = 1, \dots, p$  of the test signal:

$$W(h) = \sum_{i=1}^p \sum_{j=1}^k V(C(i, j) = h) \cdot w_{ij}, h = 1, \dots, g, \tag{6}$$

$$\text{where } V(C(i, j) = h) = \begin{cases} 1, & \text{if } C(i, j) = h \\ 0, & \text{if } C(i, j) \neq h \end{cases}$$

Finally, the class label of the test signal was determined by the majority vote:

$$\text{argmax}_{m=1, \dots, g} \{W(m)\}. \tag{7}$$

When thinking all cases also being test items in some fold in modelling and testing 131 ties occurred, rather many, for all 1393 signals. Obviously, the relatively high number of ties reflects the complexity of the data. Including the above consideration of ties improved the uppermost mean accuracy of Table 2 part (a) only around 0.7%

**TABLE 3** Results of multilayer feedforward (perceptron) neural network when the data standardized with zscore: (A) Data of eight classes, (B) data of eight classes with weights computed with Relief algorithm in Matlab, data of six classes when classified as in the preceding alternative, (C) then data of four classes classified similarly and (D) data of two classes similarly

Mode of classification	Hidden layer size	Average sensitivity (true positive rate) %								Accuracy %
		LQT1	HCMM	CPVT	WT	HCMT	LQT2	DCM	BrS	Mean and std
(a)	36	88.7	82.9	43.4	33.6	47.2	41.0	65.2	62.7	56.9 ± 4.5
(b)	36		80.3	59.6	47.4	46.8	69.2	67.0		62.2 ± 4.5
(c)	36			62.5	59.5	63.5	76.6			64.4 ± 4.5
(d)	21				88.5	72.8				82.2 ± 4.2

**TABLE 4** Results of random forests classifier when the data standardized with zscore: (A) Data of eight classes, (B) data of six classes, (C) data of four classes and (D) data of two classes

Mode of classification	Number of trees	Average sensitivity (true positive rate) %								Accuracy %
		LQT1	HCMM	CPVT	WT	HCMT	LQT2	DCM	BrS	Mean and std
(a)	112	93.3	85.6	67.9	51.8	55.8	52.7	75.2	69.3	68.1 ± 3.3
(b)	58		87.8	78.1	60.2	57.1	73.1	73.6		73.0 ± 4.9
(c)	32			81.1	66.0	72.6	79.7			74.5 ± 5.2
(d)	36				93.4	76.6				86.7 ± 3.8

compared with that of part (b). Table 2 parts (b) and (c) show how the results became better while decreasing the number of classes, that is, decreasing data complexity.

In Table 3, there are results computed with feedforward neural networks of three layers (one hidden layer). Levenberg–Marquardt learning algorithm was applied, which achieved approximately as good classification accuracies as those of Bayesian regularization, but slightly better (~1%) than those given by Resilient backpropagation algorithm. The parameter of the size of the hidden layer (the numbers of its neurons or nodes) was increased from 10 upward, since greater sizes than 10 gave typically slightly better (up to ~4%–5%) results tested up to 50 neurons. Results given by sizes 21 or 36 are presented depending on which gave a little better (~1%) classification accuracy. Pre-processing of zscore standardization and weights computation with Relief algorithm were also executed, but they would not have been necessary when feedforward neural networks are not sensitive to scales of attributes unlike nearest neighbour searching, but they “tune” network connection weights correspondingly in their learning processes based on optimization.

In Table 4, there are results obtained by the random forest classifier.<sup>22</sup> In all classification tasks ([a]–[d]), a random forests classifier was tested from 1 to 150 trees in a forest with stepsize of 1. Otherwise, we used the default

parameter settings. The same 10-fold cross-validation division was used with random forests as with other classification algorithms, and performance measures (accuracy and sensitivities) were computed from all folds in percentages and finally mean and standard deviation of the performance measures were evaluated. Performance measures were computed from signal-level predictions, which were obtained by taking the mode of peak-level predictions in the case of each signal data. The peak-level data from each signal was included only in one fold. In all classification settings ([a]–[d]), data were zscore standardized to have zero mean and unit variance. The best parameter setting was selected based on the topmost accuracy gained from the signal-level predictions. Table 4 results show that the random forests classifier was able to achieve the highest accuracies within all classification methods tested.

In Reference 17, an approach of data complexity on data items was used according to the division into either classified correctly or incorrectly data items and then k-nearest neighbours were used by computing for every test data item how many of its k-nearest neighbours did not have the same class label as that of the test data item. Since in Reference 17, complexity was based on either correctly or incorrectly classified data items, this principle differed from our approach. Yet, we apply this for a kind of rough comparison with the results of our method.



In Table 2, the classification accuracies of rows (b)–(e) give indirectly such values, since they contain the classification accuracies of nearest neighbour searching. By subtracting each accuracy value from 100 and dividing the difference by 100, we obtain complexity values 0.358, 0.316, 0.291 and 0.160 for the numbers of classes 8, 6, 4 and 2 to be compared with the rightmost complexity values in Table 1. Both series contain a series of decreasing data complexity, but these cannot be compared as commensurable when they were computed with two different principles.

## 5 | CONCLUSION

A method utilizing nearest neighbour searching was presented for elucidating data complexity. The method was tested with peak data derived from calcium transients measured from induced pluripotent cardiomyocyte-based origin. We showed how data complexity of classes and the subsets of data became smaller when classes were decreased from 8 to 6, 4 and 2. Data complexity could be seen in these data, such as classes are somewhat overlapping each other, and their dispersions somewhat intermingled without clear boundaries between classes. These overlapping and mixing were abundant enough so that there were probably difficult to determine any items to be outliers. Thus, data cleaning did not aid to improve classification accuracies of the current data set. On the other hand, when data cleaning did not aid, it showed that the current data are complicated and, for this reason, very suitable for the present study. In general, when data cleaning affects some other data, it may improve classification results.

The superiority of random forests classifier in this classification task compared with other methods tested is important. Random forests being tree-based is a transparent machine learning method compared with black box deep learning methods, which are a standard approach in numerous machine learning tasks nowadays. Transparency is a crucial issue in medical decision-making and in other domains as well. When a machine learning is used for diagnostic purpose, we need to have a way to justify or check afterwards, how the decision or prediction has been made. This can be also a mandatory requirement in practice due to legal purposes, for example, if a machine learning method is used in real-world situation, likewise in a hospital environment. Since a random forest is a tree-based transparent method, we can follow and back-track, basically, how the decision has been made. It is excellent from the practical point of view that a random forest classifier is the best method for classifying diseases and controls.

## ACKNOWLEDGEMENTS

The authors would like to thank Academy of Finland Centre of Excellence in Body-on-Chip Research.

## CONFLICT OF INTEREST

The authors declare no conflicts of interest.

## DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

## ORCID

Martti Juhola  <https://orcid.org/0000-0003-2298-9553>

## REFERENCES

- Juhola M, Penttinen K, Joutsijoki H, et al. Signal analysis and classification methods for calcium transient data of stem cell derived cardiomyocytes. *Comput Biol Med.* 2015;61:1-7. doi:10.1016/j.combiomed.2015.03.016
- Juhola M, Joutsijoki H, Penttinen K, Aalto-Setälä K. Detection of genetic cardiac diseases by Ca<sup>2+</sup> transient profiles using machine learning methods. *Sci Rep.* 2018;8:9355.
- Juhola M, Joutsijoki H, Penttinen K, Aalto-Setälä K. Differentiation of genetic cardiac diseases on the basis of artificial intelligence. *Eur J Biomed Inform.* 2019;15(3):43-52.
- Kiviahho AL, Ahola A, Larsson K, et al. Distinct electrophysiological and mechanical beating phenotypes of long QT syndrome type 1-specific cardiomyocytes carrying different mutations. *Int J Cardiol Heart Vasc.* 2015;25(8):19-31.
- Kuusela J, Larsson K, Shah D, Prajapati C, Aalto-Setälä K. Low extracellular potassium prolongs repolarization and evokes early after depolarization in human induced pluripotent stem cell-derived cardiomyocytes. *Biol Open.* 2017;6:777-784.
- Shah D, Prajapati C, Penttinen K, et al. hiPSC-derived cardiomyocyte model of LQT2 syndrome derived from asymptomatic and symptomatic mutation carriers reproduces clinical differences in aggregates but not in single cells. *Cell.* 2020;9(5):1153.
- Hwang H, Liu R, Maxwell JT, Yang J, Xu C. Machine learning identifies abnormal Ca<sup>2+</sup> transients in human induced pluripotent stem cell-derived cardiomyocytes. *Sci Rep.* 2020;10:16977.
- Shah D, Virtanen L, Prajapati C, et al. Modeling of LMNA-related dilated cardiomyopathy using human induced pluripotent stem cells. *Cell.* 2019;8(6):594.
- Ojala M, Prajapati C, Pölönen RP, et al. Mutation-specific phenotypes in hiPSC-derived cardiomyocytes carrying either myosin-binding protein C or  $\alpha$ -tropomyosin mutation for hypertrophic cardiomyopathy. *Stem Cells Int.* 2016;2016:1-16.
- Prajapati C, Ojala M, Aalto-Setälä K. Divergent effects of adrenaline in human induced pluripotent stem cell-derived cardiomyocytes obtained from hypertrophic cardiomyopathy. *Dis Model Mech.* 2018;11:dmm032896.
- Liang P, Sallam K, Wu H, et al. Patient-specific and genome-edited induced pluripotent stem cell-derived cardiomyocytes elucidate single-cell phenotype of Brugada syndrome. *J Am Coll Cardiol.* 2016;68:2086-2096.
- Penttinen K, Prajapati C. submitted. 2022.

13. Penttinen K, Swan H, Vanninen S, et al. Antiarrhythmic effects of dantrolene in patients with catecholaminergic polymorphic ventricular tachycardia and replication of the responses using iPSC models. *PLoS One*. 2015;10:5.
14. Pölönen RP, Penttinen K, Swan H, Aalto-Setälä K. Antiarrhythmic effects of carvedilol and flecainide in cardiomyocytes derived from catecholaminergic polymorphic ventricular tachycardia patients. *Stem Cells Int*. 2018;2018:1-11.
15. Pölönen RP, Swan H, Aalto-Setälä K. Mutation-specific differences in arrhythmias and drug responses in CPVT patients: simultaneous patch clamp and video imaging of iPSC derived cardiomyocytes. *Mol Biol Rep*. 2020;47:1067-1077.
16. Cano J-R. Analysis of data complexity measures for classification. *Expert Syst Appl*. 2013;40:4820-4831.
17. Smith MR, Martinez T, Giraud-Carrier C. An instance level analysis of data complexity. *Mach Learn*. 2014;95:25-256. doi: [10.1007/s10994-013-5422-z](https://doi.org/10.1007/s10994-013-5422-z)
18. Li L, Abu-Mostafa YS. Data complexity in machine learning, Caltech Computer Science Technical Report CaltechCSTR:2006.004. <http://resolver.caltech.edu/CaltechCSTR:2006.004>
19. Kira K, Rendell L. The feature selection problem: traditional methods and a new algorithm, AAAI-92 Proceedings of the Ninth International Workshop on Machine Learning. 1992. 249-256. <http://www.aaai.org/Library/AAAI/1992/aaai92-020.php>
20. Laurikkala J, Juhola M, Kentala E. Informal identification of outliers in medical data, 5th International Workshop on Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP-2000) (A workshop at the 14th European Conference on Artificial Intelligence, ECAI-2000). 2000. Proceedings, 20-24.
21. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res*. 2002;16:321-357.
22. Breiman L. Random forests. *Mach Learn*. 2001;45:5-32.

**How to cite this article:** Juhola M, Joutsijoki H, Penttinen K, Shah D, Aalto-Setälä K. A method to measure data complexity of a complicated medical data set. *Int J Imaging Syst Technol*. 2022;1-10. doi:[10.1002/ima.22760](https://doi.org/10.1002/ima.22760)