algorithms

MDPI

Article

SAIN: Search-And-INfer, a Mathematical and Computational Framework for Personalised Multimodal Data Modelling with Applications in Healthcare

Cristian S. Calude [1], Patrick Gladding [2], Alec Henderson [3] and Nikola Kasabov [4,*]

[1] Department of Computer Science, University of Auckland, Auckland 1142, New Zealand; c.calude@auckland.ac.nz
[2] Bioengineering Institute, University of Auckland, Auckland 1142, New Zealand; pgone11@gmail.com
[3] University of Queensland Center for Clinical Research, Brisbane, QLD 4072, Australia; alec.henderson@uq.edu.au
[4] Department of Computer Science, Auckland University of Technology , Auckland 1142, New Zealand
* Correspondence: nkasabov@aut.ac.nz

Abstract

Personalised modelling has become dominant in personalised medicine and precision health. It creates a computational model for an individual based on large data repositories of existing personalised data, aiming to achieve the best possible personal diagnosis or prognosis and derive an informative explanation for it. Current methods are still working on a single data modality or treating all modalities with the same method. The proposed method, SAIN (Search-And-INfer), offers better results and an informative explanation for classification and prediction tasks on a new multimodal object (sample) using a database of similar multimodal objects. The method is based on different distance measures suitable for each data modality and introduces a new formula to aggregate all modalities into a single vector distance measure to find the closest objects to a new one, and then use them for a probabilistic inference. This paper describes SAIN and applies it to two types of multimodal data, cardiovascular diagnosis and EEG time series, modelled by integrating modalities, such as numbers, categories, images, and time series, and using a software implementation of SAIN.

Keywords: search in multimodal data; inference in multimodal data; personalised modelling; precision health

Academic Editor: Roberto Montemanni

Received: 26 June 2025
Revised: 10 September 2025
Accepted: 18 September 2025
Published: 26 September 2025

Citation: Calude, C.S.; Gladding, P.; Henderson, A.; Kasabov, N. SAIN: Search-And-INfer, a Mathematical and Computational Framework for Personalised Multimodal Data Modelling with Applications in Healthcare. *Algorithms* **2025**, *18*, 605. https://doi.org/10.3390/a18100605

1. Introduction

Multimodal data has been gathered at a personalised level in large quantities, worldwide, for many applications, such as neuro-imaging analysis, personalised health diagnosis and prognosis, environmental modelling, and financial modelling, to mention only a few of them. Still, there are no efficient methods to integrate various multimodal data for a new subject and derive a more accurate and explainable diagnosis or prognosis based on existing multimodal data of many other subjects. The goal of this paper is to create such a method.

There are three main approaches to multimodal data integration in machine learning, explored so far [1]:

1. Early integration, where a common vector represents all modalities used for training a model and for its recall. This approach has been used for integrating time series and

textual information [2,3], for image integration [4], and for the integration of clinical, social, and cognitive data modalities to predict psychosis in young adults [5]. While the method in [4] is based on deep, feedforward neural networks, the methods in [2,5] use a brain-inspired spiking neural network architecture NeuCube [6].

2. Late integration, where a model is created and trained for each of the modalities of data, and the results from all models are integrated to calculate the output. This approach has been demonstrated in [1] on integrating clinical, genetic, cognitive, and social data for medical prognosis.

3. Hybrid, early, late, and intermediate integration of data modalities, where the two approaches above are combined [1].

The proposed SAIN method in this paper is designed for early integration of data modalities, where specific encoding and distance metrics are suggested for different types of data, along with novel algorithms for search in a multimodal database and inference. These search and inference algorithms are related to building a personalised model for individual outcome assessment and its explanation.

Personalised modelling is concerned with the creation of an individual model for a new personal (individual) record of data X, using an already existing repository D of many other personal records for which the outcomes are known, to assess the outcome of the new record X [7]. Methods for personalised modelling have been developed to work mainly on a single modality of data [8,9]. These methods have been used in many applications and constitute the state of the art in the field (e.g., refs. [1,9–13]). In [9,14], a personalised modelling is proposed based on static and temporal modalities of data, which are used to train a spiking neural network model.

The enormous growth of personal multimodal data worldwide demands more advanced methods for personal modelling with the use of multimodal data. This paper offers such a method, called SAIN, where the specificity of each data modality is considered and new algorithms are proposed for the encoding of multimodal data, for search in a multimodal data repository, and for multimodal inference, along with its explanation and visualisation. In contrast with the statistical solutions used in [15–17], we adopt a probabilistic framework that gives more precise evaluations of probabilities of outcomes for an individual.

2. Mathematical Description

In this section, we present the mathematical method. In detail, we start with the database coding and a class of distances (metrics) to be used in this article, followed by the list of tasks (problems) and their solutions. Detailed examples illustrate the solutions. Detailed solutions to three critical problems, namely survival analysis, heart disease diagnosis, and time series classification, are presented in detail and illustrated with numerical examples.

2.1. Database

We will work with the multidimensional data described as follows:

1. $m > 1$ objects (samples) $o_1, \ldots, o_m$;
2. Each object o_i $(1 \leq i \leq m)$ is defined by $n > 1$ criteria (variables) $c_1, \ldots, c_n$ with values in linearly ordered domains D_i with $\min D_i$ and $\max D_i$; if some value $a_{i,j} \in D_i$ $(1 \leq i \leq m, 1 \leq j \leq n)$ is either missing or uncertain, then its value is recorded as ∞;
3. $n > 1$ weights $w_1, \ldots, w_n$ in $(0,1)$ with $\sum_{i=1}^{n} w_i = 1$, where each w_i $(1 \leq i \leq n)$ quantifies the importance of the criterion c_i; if $w_i = \frac{1}{n}$ for all $1 \leq i \leq n$, then all criteria are equally important; a criterion c_i is ignored if $w_i = 0$.

Data of independent variables are organised as in Table 1.

Table 1. Unlabelled database.

Objects/Criteria	c_1	c_2	...	c_j	...	c_n
o_1	$a_{1,1}$	$a_{1,2}$	...	$a_{1,j}$	...	$a_{1,n}$
$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	...	$\vdots$
o_i	$a_{i,1}$	$a_{i,2}$	...	$a_{i,j}$	...	$a_{i,n}$
$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	...	$\vdots$
o_m	$a_{m,1}$	$a_{m,2}$	...	$a_{m,j}$	...	$a_{m,n}$
w	w_1	w_2	...	w_j	...	w_n

2.2. Distance Metrics

A distance metric on a space X is a positive real-valued function $d : X \times X \to \mathbf{R}_+$ satisfying the following three conditions for all $x, y, z \in X$: (a) $d(x, y) = 0$ if and only if $x = y$, (b) $d(x, y) = d(y, x)$, (c) $d(x, z) \leq d(x, y) + d(y, z)$.

The multicriteria metrics [18,19] (used in multicriteria recommendation systems [20]) presented in this part can be used on a variety of domains $X = D_i$: they can be sets of logical values, rational numbers, percentages, digitally codified images, sounds, videos, and many others. We use a bounded distributive complemented lattice $(L, \vee, \wedge, \bar{}, 0, 1)$ to describe uniformly the domains D_i. We rank all objects according to their aggregated distance to a new one; based on that, we calculate probabilities of the new object belonging to different classes, represented in the object repository.

Here is a list with illustrative, but far from exhaustive, examples of domains D_i:

- Logical Boolean domain: $(\{0,1\}, \max, \min, \bar{}, 0, 1)$, where $\bar{x} = 1 - x, x \in \{0,1\}$.
- Logical non-Boolean domain: $\left(\left\{0, \frac{1}{N-1}, \frac{2}{N-1}, \cdots, \frac{N-2}{N-1}, 1\right\}, \max, \min, \bar{}, 0, 1\right)$, where $x \in \left\{0, \frac{1}{N-1}, \frac{2}{N-1}, \cdots, \frac{N-2}{N-1}, 1\right\}$ and $\bar{x} = 1 - x,$.
- Numerical domain with natural values: $(\{0, 1, \ldots, N\}, \max, \min, \bar{}, 0, 1)$, where $\bar{x} = N - x, x \in \{0, 1, \ldots, N\}$.
- Numerical domain with rational values: $(\{x \mid a \leq x \leq A\}, \max, \min, \bar{}, a, A)$, where $\bar{x} = A - x, a \leq x \leq A$.
- Binary code: $(\{0,1\}^n, \max, \min, \bar{}, 00\ldots0, 11\ldots1)$, where the domain consists of all binary strings of length n, $\{0,1\}^n = \{x_1x_2\ldots x_n \mid x_i \in \{0,1\}\}$ and for all $x_1x_2\ldots x_n, y_1y_2\ldots y_n \in \{0,1\}^n$, $\max(x_1x_2\ldots x_n, y_1y_2\ldots y_n) = \max(x_1, y_1)\max(x_2, y_2)\ldots\max(x_n, y_n)$, $\min(x_1x_2\ldots x_n, y_1y_2\ldots y_n) = \min(x_1, y_1)\min(x_2, y_2)\ldots\min(x_n, y_n)$, $\overline{x_1x_2\ldots x_n} = (1 - x_1)(1 - x_2)\ldots(1 - x_n)$.

In the lattice $(L, \vee, \wedge, \bar{}, 0, 1)$ we introduce, following [18], the metric:

$$d(x, y) = \begin{cases} (x \wedge \bar{y}) \vee (\bar{x} \wedge y), & \text{if } x \neq y, \\ 0, & \text{otherwise}, \end{cases}$$

for $x, y \in L$. This metric d can be extended to $L \cup \{\infty\}$ as follows:

$$d_\infty(x, y) = \begin{cases} d(x, y), & \text{if } x, y \in L, \\ \sigma(x), & \text{if } x \in L \text{ and } y = \infty, \\ \sigma(y), & \text{if } y \in L \text{ and } x = \infty, \\ 0, & \text{otherwise}, \end{cases}$$

where $\sigma(x) = \max(x, \bar{x})$.

The metrics $d_{\infty,i}$ on $L_i \cup \{\infty\}$, $1 \leq i \leq n$, can be extended to $(L_i \cup \{\infty\})^n$, i.e., to n-dimensional vectors, as follows:

$$d_\infty(x_1x_2 \dots x_n, y_1y_2 \dots y_n) = \sum_{i=1}^{n} d_{\infty,i}(x_i, y_i), \tag{1}$$

where $x_i, y_i \in L_i \cup \{\infty\}$, $1 \leq i \leq n$.

In what follows, we write d for d_∞ when the meaning is clear from the context.

2.3. Tasks Specification

Data organised as in Tables 2–4 consists of independent objects augmented with a column of labels, the weights of criteria, and a new unlabelled object, respectively.

Additional information associated with data in Table 2 may include the range of each criterion c_j and the associated specific distance, e.g., the Euclidean distance for real numbers and the distance d for binary strings or strings over a non-binary alphabet (e.g., for images or colours).

Table 2. The labelled database.

Objects/Criteria	c_1	c_2	...	c_j	...	c_n	**Class Label**
o_1	$a_{1,1}$	$a_{1,2}$	...	$a_{1,j}$	...	$a_{1,n}$	l_1
⋮	⋮	⋮	...	⋮	...	⋮	⋮
o_i	$a_{i,1}$	$a_{i,2}$	...	$a_{i,j}$	...	$a_{i,n}$	l_i
⋮	⋮	⋮	...	⋮	...	⋮	⋮
o_m	$a_{m,1}$	$a_{m,2}$	...	$a_{m,j}$	...	$a_{m,n}$	l_m

Table 3. Weights.

Criteria Weights	c_1	c_2	...	c_j	...	c_n
w	w_1	w_2	...	w_j	...	w_n

Table 4. A new unlabelled object.

Object/Criteria	c_1	c_2	...	c_j	...	c_n
x	x_1	x_2	...	x_j	...	x_n

We consider the following tasks:

Task 1: Calculate the distance (or similarity metric) between the new object and each object in Table 2. If the distance corresponding to c_i is d_i, then

$$d(o_j, x) = \sum_{i=1}^{n} w_i \cdot d_i(a_{i,j}, x_j).$$

Task 2: Given a threshold $\delta > 0$, calculate all objects o_i at a distance at most δ to x.
Task 3: Calculate the probability of a new object belonging to a labelled class (e.g., low risk vs. high risk) using a threshold δ and Table 2.
Task 4: Rank the criteria in Table 2 and calculate the marker or markers criterion/criteria that are the most important/ones.
Task 5: Assign alternative weights to criteria.
Task 6: Test the data accuracy and method for Task 4.

2.4. Tasks Solutions

For Task 1, we calculate the distances $d_\infty(o_i, x)$ between each object o_i in Table 2 and x in Table 4.

For Task 2, given a threshold $\delta > 0$, we calculate all objects in Table 2 at a distance at most δ to x, that is, the objects which are δ-similar to x:

$$C_{\delta,x} = \{o_i \mid d(x, o_i) \leq \delta, 1 \leq i \leq m\},$$

and its complement $\overline{C_{\delta,x}}$.

For Task 3, we calculate the probability that x is in class label l_t, which is the ratio of the number of objects in $C_{\delta,x}$ with the label l_t to the size of the cluster $C_{\delta,x}$:

$$Prob\,(x \text{ has label } l_t) = \frac{\#\{o_i \in C_{\delta,x} \mid l_i = l_t\}}{\#(C_{\delta,x})},$$

where $\#\{\dots\}$ means the number of elements in the set $\{\dots\}$.

For Task 4, we work with Table 2. Recall that, for each criterion c_i, we have a domain D_i augmented with information "high" or "low," indicating whether higher or lower values are desirable. Based on this information, we can construct a hypothetical object (see Table 5) which has the most desirable values for each criterion: one could see this object as an "exemplar" one.

Table 5. The hypothetical object.

Object/Criteria	c_1	c_2	$\dots$	c_j	$\dots$	c_n	**Class Label**
o_E	n_1	n_2	$\dots$	n_j	$\dots$	n_n	l_h

Sometimes, criteria are interrelated or correlated. This means that, in some cases, there is no unique "exemplar object", but a couple of them have to be studied in ranking the importance of criteria.

For example, fix an "exemplar object" o_E.

1. Compute the distances $d_\infty(o_i, o_E)$ between each object o_i in Table 2 and o_E, so obtain a vector with n non-negative real components $V_0 = (d_1^0, \dots d_n^0)$.
2. For each $1 \leq t \leq m$, compute the distances $d_\infty(o_i, o_E)$ taking into consideration all criteria in Table 2 except c_t: obtain the vector $V_t = (d_1^t, \dots d_n^t)$.
3. Compute the distances between $dist(V_0, V_t)$, $1 \leq t \leq m$ using the formula

$$dist(V_0, V_t) = \sum_{i=1}^{n} |d_{0,i} - d_{t,i}|,$$

and sort them in increasing order. The criterion c_t is a marker if $dist(V_0, V_t) \geq dist(V_0, V_j)$, for every $1 \leq j \leq m$.

We repeat this procedure for each "exemplar object" and study possible variations.

For Task 5, normalise the distances $dist(V_0, V_t)$ and use these values to construct the weights w_i^*, $1 \leq t \leq m$.

For Task 6, assume we have weights (w_i) associated to Table 2 (see Table 1). To test the accuracy of the data and method used for Task 4, compare the original weights (w_i) with (w_i^*). Serious discrepancies should signal issues either with the data or the choices made in the applications of the method.

2.5. An Example

We illustrate the above tasks with an example of a labelled database in Table 6 and a new object (see Table 7), all having the following seven characteristics (the last column has the label classes 1 and 2):

c_1: real number $\{0-100\}$, e.g., age, weight, BMI etc.;
c_2: Boolean value $\{0,1\}$, e.g., gender;
c_3: integer number $\{0-10,000\}$, e.g., gene expression;
c_4: categorical {small, med, large }, e.g., size of tumour, body size, keywords;
c_5: colour {red, yellow, white, black}, e.g., colour of a spot on the body, on the heart;
c_6: spike sequence of $\{-1,0,1\}$ e.g., encoded EEG, ECG;
c_7: black and white image, e.g., MRI, face image.

Table 6. An example of labelled data.

68.2	0	6789	small	red	0, 1, −1, −1, 1,1, 0, 0, 1, −1	1, 1, 0 0, 0, 1 0, 0, 1	1
93	1	98,000	medium	yellow	0, −1, −1, −1, −1, 0, 0, 1, −1, 1	1, 0, 0 0, 0, 1 0, 0, 1	1
44.5	1	5600	large	red	0, 1, −1, 1, −1, 1, 0, 0, 1, −1	1, 1, 0 1, 0, 1 1, 1, 1	1
56.8	0	89	small	white	1, −1, −1, −1, −1, 1, 0, 0, 1, −1	1, 1, 0 0, 1, 1 1, 0, 1	1
26.3	0	9456	large	black	1, −1, −1, −1, 0,1, 0, 0, 1, −1	1, 1, 0 1, 1, 1 1, 0, 1	2
81.5	1	78, 955	medium	red	0, 1, −1, 1, −1, −1, 0, 0, 1, −1	1, 1, 0 0, 0, 1 1, 1, 1	2
56.7	1	68, 900	small	black	1, −1, −1, 1, −1, 1, 0, 0, 1, 1	1, 1,1 0, 0,1 1, 1,1	2
20	0	7833	large	yellow	1, 1, −1, −1, 1,1, 0, −1, −1, 1	1, 0, 0 0, 0, 1 1, 1, 1	2
20	0	7833	∞	yellow	1, 1, −1, −1, 1,1, 0, −1, −1, 1	1, 0, 0 0, 0, 1 1, 1, 1	2

Table 7. An example of new unlabelled object.

48.5	1	45,679	large	red	1, 0, 0, −1, 1, −1, 1, 0, 0, 1	1, 1, 0 0, 0, 1 1, 0, 1

In this fictitious example, for simplicity, we did not use weights.

The first step is to code the data in Tables 6 and 7. The new data is in Tables 8 and 9.

Then, we normalise the data in Tables 8 and 9—the entries in the first, third, and fourth columns have been divided by 100, 10,000, and 2, respectively. The entries in the last three

columns have been transformed into reals in the unit interval, and the column of labels has been removed. In this way, we have obtained Tables 10 and 11.

Table 8. Coded labelled data.

o_1	68.2	0	6789	0	FF0000 111111110000000000000000	0122110012	110001001	1
o_2	93	0	98,000	1	FFFF00 111111111111111100000000	0222200121	110001001	1
o_3	44.5	1	5600	2	FF0000 111111110000000000000000	0121210012	110101111	1
o_4	56.8	0	89	0	FFFFFF 111111111111111111111111	1222210012	110011101	1
o_5	26.3	0	9456	2	000000 000000000000000000000000	1222010012	110111101	2
o_6	81.5	1	78,955	1	FF0000 111111110000000000000000	0121220012	110001111	2
o_7	56.7	1	68,900	0	000000 000000000000000000000000	1221210011	111001111	2
o_8	20	0	7833	2	FFFF00 111111111111111100000000	1122110221	100001111	2
o_9	20	0	7833	∞	FFFF00 111111111111111100000000	1122110221	100001111	2

Table 9. The new unlabelled object coded.

x	48.5	1	45,679	2	FF0000 111111110000000000000000	1002121001	110001101

Table 10. Coded labelled normalised data.

o_1	0.682	0	0.06789	0	0.2	0.0122110012	0.110001001
o_2	0.93	1	0.98	0.5	0.6	0.0222200121	0.100001001
o_3	0.445	1	0.056	1	0.2	0.0121210012	0.110101111
o_4	0.568	0	0.00089	0	1	0.1222210012	0.110011101
o_5	0.263	0	0.09456	1	0	0.1222010012	0.110111101
o_6	0.815	1	0.78955	0.5	0.2	0.0121220012	0.110001111
o_7	0.567	1	0.689	0	0	0.1221210011	0.111001111
o_8	0.2	0	0.07833	1	0.6	0.1122110221	0.100001111
o_9	0.2	0	0.07833	∞	0.6	0.1122110221	0.100001111

Table 11. The new unlabelled object coded normalised.

x	0.485	1	0.45679	1	0.2	0.1002121001	0.110001101

Then, we choose an appropriate distance according to each criterion. In this example, we used the Euclidean distance for all criteria (see Tables 12 and 13).

We can compute $C_{\delta,x} = \{o_i \mid d(o_i, x) \leq \delta\}$ and, accordingly, the probability that x would be labelled in class 1 or class 2.

If $\delta = 3.5$, then $C_{3.5,x} = \{o_1, o_2, o_3, o_5, o_6, o_7, o_8\}$ so the probability that x is in class 1 is $2/7$ and the probability that x is in class 2 is $5/7$. If $\delta = 2.5$, then its closest cluster is $C_{2.5,x} = \{o_2, o_3, o_5, o_6, o_7, o_8\}$, so the probability that x is in class 1 is $1/3$ and the probability that x is in class 2 is $2/3$.

Table 12. Normalised distances from the new object to all objects.

$d(o_1, x)$	0.197	1	0.3889	1	0	0.4	0.11111111	3.09701111
$d(o_2, x)$	0.445	0	0.52321	0.5	0.33333333	0.6	0.22222222	2.62376556
$d(o_3, x)$	0.04	0	0.40079	0	0	0.5	0.22222222	1.16301222
$d(o_4, x)$	0.083	1	0.4559	1	0.66666667	0.45	0.11111111	3.76667778
$d(o_5, x)$	0.222	1	0.36223	0	0.33333333	0.45	0.22222222	2.58978556
$d(o_6, x)$	0.33	0	0.33276	0.5	0	0.45	0.11111111	1.72387111
$d(o_7, x)$	0.082	0	0.23221	1	0.33333333	0.45	0.22222222	2.31976556
$d(o_8, x)$	0.285	1	0.37846	0	0.33333333	0.45	0.22222222	2.66901556
$d(o_9, x)$	0.285	1	0.37846	1	0.33333333	0.45	0.22222222	3.66901556

Table 13. Ranking of distances in increasing order in Table 12.

$d(o_3, x)$	0.04	0	0.40079	0	0	0.5	0.22222222	1.16301222
$d(o_6, x)$	0.33	0	0.33276	0.5	0	0.45	0.11111111	1.72387111
$d(o_7, x)$	0.082	0	0.23221	1	0.33333333	0.45	0.22222222	2.31976556
$d(o_5, x)$	0.222	1	0.36223	0	0.33333333	0.45	0.22222222	2.58978556
$d(o_2, x)$	0.445	0	0.52321	0.5	0.33333333	0.6	0.22222222	2.62376556
$d(o_8, x)$	0.285	1	0.37846	0	0.33333333	0.45	0.22222222	2.66901556
$d(o_1, x)$	0.197	1	0.3889	1	0	0.4	0.11111111	3.09701111
$d(o_9, x)$	0.285	1	0.37846	1	0.33333333	0.45	0.22222222	3.66901556
$d(o_4, x)$	0.083	1	0.4559	1	0.66666667	0.45	0.11111111	3.76667778

Which induces the following ranking of the objects in Table 8: o_3, o_6, o_7, o_5, o_2, o_8, o_1, o_9, o_2.

For Task 4, assume that the criteria $c_1, \ldots, c_7$ in Table 10 have the additional information (m, m, m, m, m, M, M), where m (M) means that the exemplar value is the minimum (maximum) value. Based on this vector, we compute the exemplar object (see Table 14).

Table 14. An exemplar object.

o_E	0.2	0	0.00089	0	1	0.1222210012	0.100001001

Next we calculate $V_0, \ldots, V_t$, (see Table 15), and finally the distances $Dist(V_0, V_t)$, $t = 1, 2, \ldots, 7$ and the weights as their normalised values (see Table 16). The marker, in this case, is the criterion c_5.

Table 15. Vectors $V_0, \ldots, V_m$, rounded to two decimals.

V_0	V_1	V_2	V_3	V_4	V_5	V_6	V_7
1.469	0.987	1.469	1.402	1.469	0.669	1.359	1.459
3.709	2.979	2.709	2.730	3.209	3.309	3.609	3.709

Table 15. *Cont.*

V_0	V_1	V_2	V_3	V_4	V_5	V_6	V_7
3.220	2.975	2.220	3.165	2.220	2.420	3.110	3.210
0.378	0.010	0.378	0.378	0.378	0.378	0.378	0.368
2.167	2.104	2.167	2.073	1.167	1.167	2.167	2.157
3.824	3.209	2.824	3.035	3.324	3.024	3.714	3.814
3.066	2.699	2.066	2.378	3.066	2.066	3.066	3.055
1.487	1.487	1.487	1.410	0.487	1.087	1.477	1.487
1.487	1.487	1.487	1.410	0.487	1.087	1.477	1.487

Table 16. Distances $Dist(V_0, V_t)$ and (normalised) weights.

Distances	2.870	4.00	2.826	5.00	5.60	0.450	0.061
Weights	0.137	0.192	0.135	0.240	0.269	0.021	0.002

2.6. Complexity Estimation of the SAIN Method

The proposed method to compute the similarity between a new object and N objects in a data repository is linear in N, so very fast.

3. Survival Analysis in SAIN

Medical survival analysis evaluates the time until an event of interest occurs, like death or disease recurrence, in a group of patients. This analysis is often used to compare treatment outcomes or predict prognosis. In contrast with the statistical solutions used in [15–17], we adopt a probabilistic framework that gives more precise evaluations of probabilities.

3.1. Data and Tasks

We are given the following data:

1. Table 17, in which the first column lists the patients treated for the same disease with the same method under strict conditions, and wherein the last column records the times until the patients' deaths.

Table 17. Survival database.

Patients/Criteria	c_1	c_2	...	c_j	...	c_n	**Units of Time**
p_1	$a_{1,1}$	$a_{1,2}$	...	$a_{1,j}$	...	$a_{1,n}$	t_1
⋮	⋮	⋮	⋯	⋮	⋯	⋮	⋮
p_i	$a_{i,1}$	$a_{i,2}$	...	$a_{i,j}$	...	$a_{i,n}$	t_i
⋮	⋮	⋮	⋯	⋮	⋯	⋮	⋮
p_m	$a_{m,1}$	$a_{m,2}$	...	$a_{m,j}$	...	$a_{m,n}$	t_m

2. Table 18, which includes the record of the new patient p.

Table 18. The new patient record.

Patient/Criteria	c_1	c_2	...	c_j	...	c_n
p	x_1	x_2	...	x_j	...	x_n

3. A threshold δ which defines the acceptable similarity between p and the relevant p_i's in the Survival database (i.e., $d(p, p_i) \leq \delta$).

We consider the following tasks:

Task 1: What is the life expectancy of p?

Task 2: What is the probability that the life expectancy of p is greater than or equal to a given T?

3.2. Tasks Solutions

Using a standard method of survival analysis

1. For Task 1,
   (a) Compute the set of patients that are similar up to δ to p:

$$C_{\delta,p} = \{p_i \mid d(p, p_i) \leq \delta, 1 \leq i \leq m\}. \tag{2}$$

   (b) Using $C_{\delta,p}$, compute the probability that p will survive the time t_j:

$$Prob_\delta\,(p \text{ survives time } t_j) = \frac{\#\{p_i \in C_{\delta,p} \mid t_i = t_j\}}{\#(C_{\delta,p})}. \tag{3}$$

   (c) Compute the life expectancy of p using the formula:

$$LE_\delta(p) = \sum_{j=1,\, t_j \in C_{\delta,p}}^{m} t_j \times Prob_\delta\,(p \text{ survives time } t_j). \tag{4}$$

2. For Task 2, calculate the probability that the life expectancy of p is at least time T:

$$Prob_\delta(LE(p) \geq T) = \sum_{j=1,\, t_j \in C_{\delta,p},\, t_j \geq T}^{m} Prob_\delta\,(p \text{ survives time } t_j). \tag{5}$$

3.3. An Example

We illustrate the above tasks with an example of a database in which columns 2–8 record patients' medical test results, and the last column records time to death (see Table 19) and a new patient (see Table 20):

Table 19. Patient records.

Patients	c_1	c_2	c_3	c_4	c_5	c_6	c_7	Units of Time
$p1$	0.682	0	0.06789	0	0.2	0.012211001	0.110001001	12.3
$p2$	0.93	1	0.98	0.5	0.6	0.022220012	0.100001001	15
$p3$	0.445	1	0.056	1	0.2	0.012121001	0.110101111	68
$p4$	0.568	0	0.00089	0	1	0.122221001	0.110011101	1.4
$p5$	0.263	0	0.09456	1	0	0.122201001	0.110111101	40.5
$p6$	0.815	1	0.78955	0.5	0.2	0.012122001	0.110001111	97.2
$p7$	0.567	1	0.689	0	0	0.122121001	0.111001111	97.2
$p8$	0.2	0	0.07833	1	0.6	0.112211022	0.100001111	55.7
$p9$	0.2	0	0.07833	∞	0.6	0.112211022	0.100001111	63.7

Table 20. New patient records.

xp	0.485	1	0.45679	1	0.2	0.1002121001	0.110001101

The distance for column 4 is $d(x,y) = |x-y| =$ and $d_\infty(x,\infty) = \max(x, 1-x)$. For example, $d_\infty(1,\infty) = \max(1, 1-1) = 1$. For all other columns, the distance is $d(x,y) = |x-y|$. Finally, the total distance is the sum of individual distances (7 terms), with the results in Table 21.

Table 21. Distances between all patients and the new patient.

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	**Distance** d
$d(p_1,p)$	0.1970	1	0.388900	1.0	0.0	0.08800109890	0.000000100	2.67390119890
$d(p_2,p)$	0.4450	0	0.523210	0.5	0.4	0.07799208800	0.010000100	1.95620218800
$d(p_3,p)$	0.0400	0	0.400790	0.0	0.0	0.08809109890	0.000100010	0.52898110890
$d(p_4,p)$	0.0830	1	0.455900	1.0	0.8	0.02200890110	0.000010000	3.36091890110
$d(p_5,p)$	0.2220	1	0.362230	0.0	0.2	0.02198890110	0.000110000	1.80632890110
$d(p_6,p)$	0.3300	0	0.332760	0.5	0.0	0.08809009890	0.000000010	1.25085010890
$d(p_7,p)$	0.0820	0	0.232210	1.0	0.2	0.02190890100	0.001000010	1.53711891100
$d(p_8,p)$	0.2850	1	0.378460	0.0	0.4	0.01199892200	0.009999990	2.08545891200
$d(p_9,p)$	0.2850	1	0.378460	1.0	0.4	0.01199892200	0.009999990	3.08545891200

The results for Task 1, (a), (b), and (c) are listed below:

1. For $\delta \geq 3.37$, $C_{\delta,p} = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$, that is the entire database. Then,
   (a) $LE_\delta(p) = 50.11$,
   (b) i. $Prob_\delta\,(p \text{ survives time } = 12.3) = 1/9$,
   ii. $Prob\,(p \text{ survives time } = 15) = 1/9$,
   iii. $Prob_\delta\,(p \text{ survives time } = 68) = 1/9$,
   iv. $Prob_\delta\,(p \text{ survives time } = 1.4) = 1/9$,
   v. $Prob_\delta\,(p \text{ survives time } = 40.5) = 1/9$,
   vi. $Prob_\delta\,(p \text{ survives time } = 97.2) = 2/9$,
   vii. $Prob_\delta\,(p \text{ survives time } = 55.7) = 1/9$,
   viii. $Prob_\delta\,(p \text{ survives time } = 63.7) = 1/9$.
   (c) i. $Prob_\delta(LE_\delta(p) \geq 1.4) = 1$,
   ii. $Prob_\delta(LE_\delta(p) \geq 12.3) = 8/9$,
   iii. $Prob_\delta(LE_\delta(p) \geq 15) = 7/9$,
   iv. $Prob_\delta(LE_\delta(p) \geq 40.5) = 6/9$,
   v. $Prob_\delta(LE_\delta(p) \geq 55.7) = 5/9$,
   vi. $Prob_\delta(LE_\delta(p) \geq 63.7) = 4/9$,
   vii. $Prob_\delta(LE_\delta(p) \geq 68) = 3/9$,
   viii. $Prob_\delta(LE_\delta(p) \geq 97.2) = 2/9$,

   We can calculate other probabilities; for example, $Prob_\delta(LE_\delta(p) \geq 60) = Prob_\delta(LE_\delta(p) \geq 63.7) + Prob_\delta(LE_\delta(p) \geq 68) = 3/9 + Prob_\delta(LE_\delta(p) \geq 97.2) = 2/9 = 1/9 + 1/9 + 2/9 = 4/9$.
2. For $\delta \geq 2.5$, $C_{\delta,p} = \{v_2, v_3, v_5, v_6, v_7, v_8\}$. Then,
   (a) $LE_\delta(p) = 62.27$,
   (b) i. $Prob\,(p \text{ survives time } = 15) = 1/6$,
   ii. $Prob\,(p \text{ survives time } = 68) = 1/6$,

iii. $Prob\,(p \text{ survives time } = 40.5) = 1/6$,
iv. $Prob\,(p \text{ survives time } = 97.2) = 2/6$,
v. $Prob\,(p \text{ survives time } = 55.7) = 1/6$,

(c) i. $Prob_\delta(LE_\delta(p) \geq 15) = 1$,
ii. $Prob_\delta(LE_\delta(p) \geq 40) = 5/6$,
iii. $Prob_\delta(LE_\delta(p) \geq 55.7) = 4/6$,
iv. $Prob_\delta(LE_\delta(p) \geq 68) = 3/6$,
v. $Prob_\delta(LE_\delta(p) \geq 97.2) = 2/6$.

Similarly, we can calculate the probabilities $Prob_\delta(LE_\delta(p) \geq 45) = 4/6$, $Prob_\delta(LE_\delta(p) \geq 100) = 0$.

In contrast with the statistical solutions used in [14–16], we adopted a probabilistic framework that gives more precise evaluations of probabilities. The SAIN algorithms also include some statistically established methods, such as the *t*-test, for ranking variables before applying the inference method.

4. SAIN: A Modular Diagram and Functional Information Flow

The SAIN framework consists of the following modules (Figure 1):

1. Multimodal data of a new object X.
2. An existing repository D of multimodal data of many objects, labelled with their outcome.
3. A module of algorithms for searching in the database D and based on the distance between X and each object in D.
4. Defining a subset D_x from D, so that X is closer to the objects in D_x based on a given threshold.
5. A module of algorithms for building a model M_x in D_x.
6. An inference algorithm to derive the output for X from the model M_x and to visualise it for explanation purposes. Figure 1 gives a modular view of the SAIN framework and Figure 2 shows the information processing flow:
   (a) Encoding the multimodal data of X and D.
   (b) Choosing a distance matrix and similarity search in the dataset D.
   (c) Calculating the aggregated difference between the new data vector X and the closest vectors in D_x.
   (d) Creating a model M_x in D_x.
   (e) Applying inference by calculating the $X_{c,j}$ for each class C_j (or output value), using the wwkNN method in [5].
   (f) Reporting and visualisation of results of the individual model Mx. This is illustrated in Figure 3.

The inference method is based on the wwkNN (weighted variables, weighted samples *k*-nearest neighbour) proposed by Kasabov [7]. This method first ranks the impact of the variables (multimodal ones) to estimate their weights/impact towards the output using a *t*-test; then, it measures the distance between the new object X and the ones in the database D_x and weighs it. For each class C_j, the higher a variable is ranked, the closer the samples/objects belong to class C_j. The closer to X, the higher the calculated value of C_j is. The new object X is classified in class C_l if X_{cl} is the highest among all X_{cj} values in Figures 1 and 2, we present the modular diagram and the functional information flow of SAIN.

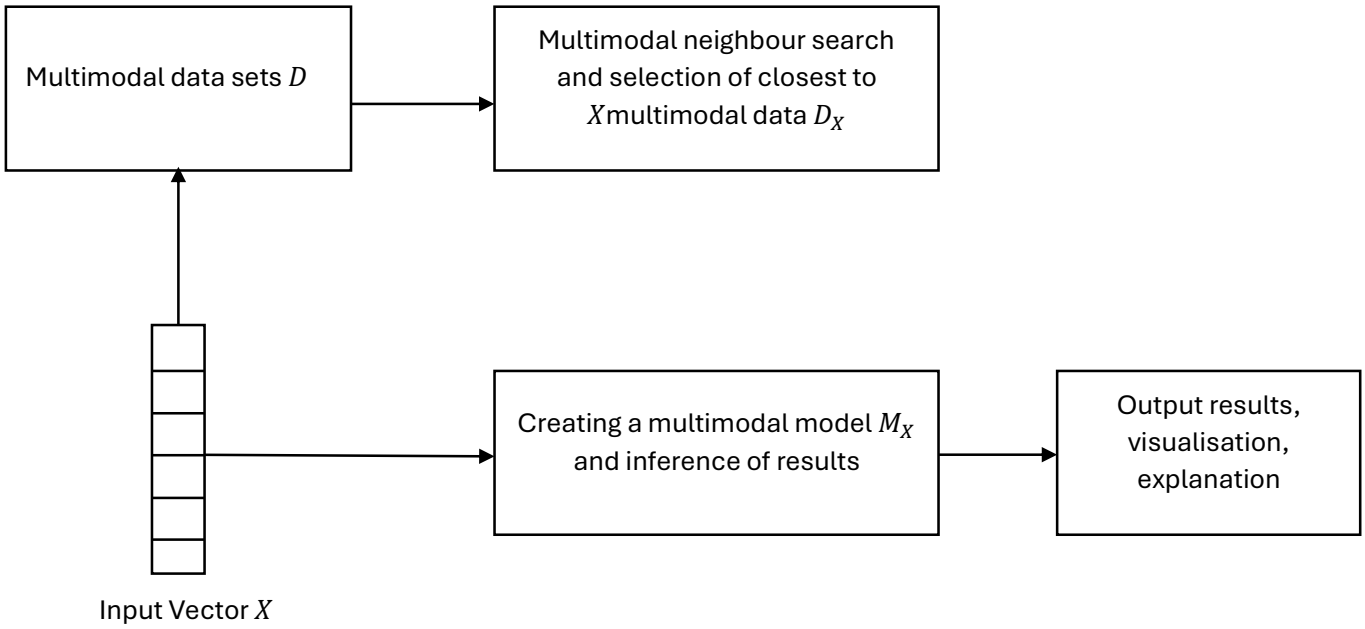


Figure 1. A modular diagram of the proposed SAIN computational framework.

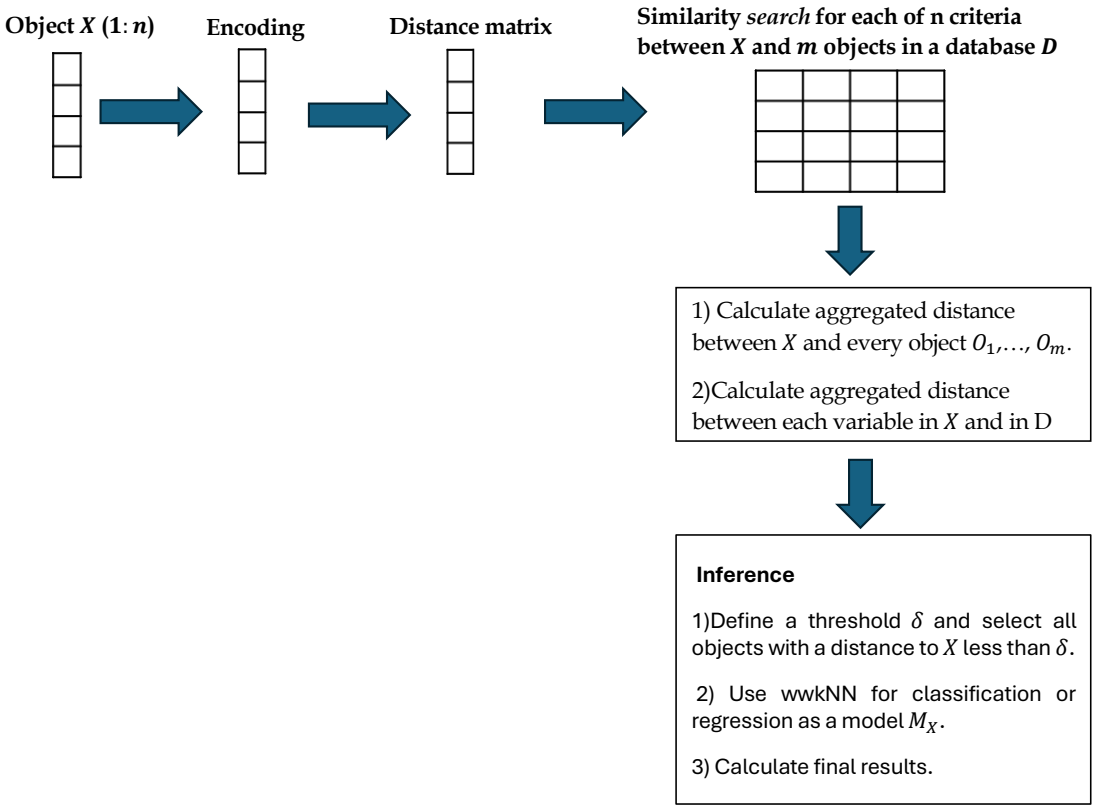


Figure 2. A flow of data and information processing in the SAIN computational framework.

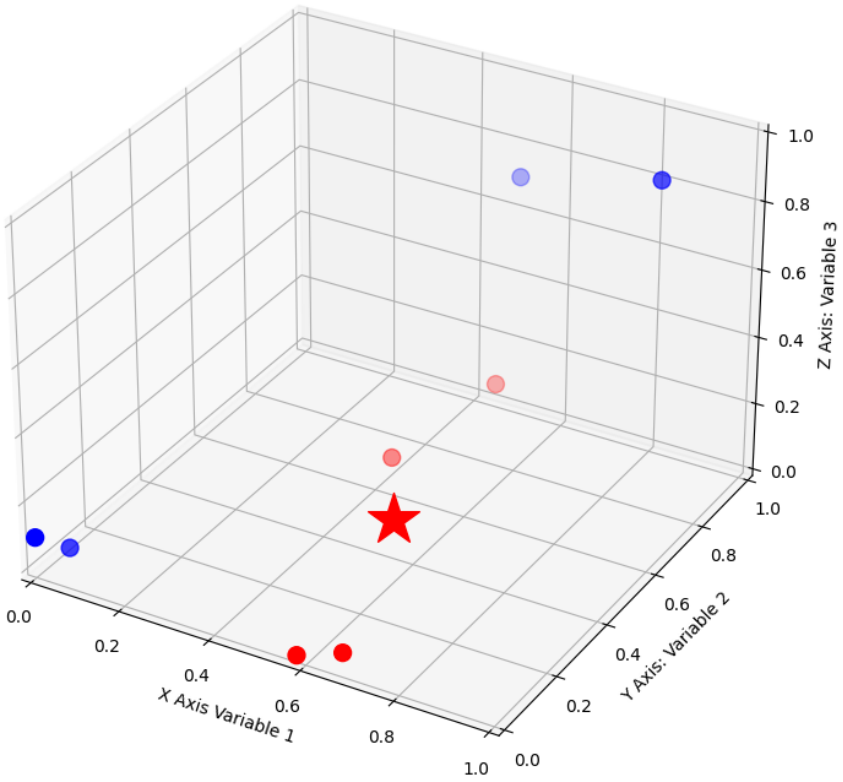


Figure 3. An example of visualisation of a personalised SAIN model. The closest four samples (out of 6) to the new object (star) are from class 1 (in red, and class 2 in blue) using the top three informative variables. Each sample is a multimodal one, and the top 3 variables can be of different modalities.

5. Case Studies for Medical Diagnosis and Prognosis

We present three case studies in which we applied SAIN.

5.1. Heart Disease Diagnosis

We worked with the well-known Cleveland dataset, which contains multiple data types [21]. The UCI Heart Disease dataset includes 76 attributes. As in most articles, the attributes in our experiment data were restricted to 14, see Table 22.

Table 22. The 14 variables used in the heart disease diagnosis case.

Name	Data Type	Definition
age	integer	age in years
sex	binary	sex
cp	{1,2,3,4}	chest pain type
trestbps	integer	resting blood pressure
chol	integer	serum cholesterol in mg/dL
fbs	binary	fasting blood sugar > 120 mg/d
restecg	{0,1,2}	resting electrocardiographic results
thalach	integer	maximum heart rate achieved
exang	binary	exercise-induced angina
oldpeak	float	ST depression induced by exercise relative to rest
slope	{1,2,3}	the slope of the peak exercise ST segment
ca	{0,1,2,3 }	number of major vessels colored by flourosopy
thal	{3,6,7}	heart status
num	{0,1,2,3,4}	diagnosis of heart disease

The problem is a binary classification of whether the patient has or does not have heart disease.

First, we selected suitable distance metrics and weights to classify the attributes. For binary objects, the distance metric is simply whether they are equal; for non-binary discrete objects such as resting electrocardiographic results, the appropriate distance measure is not obvious and should be informed by an expert. We give the electrocardiographic results of 0 for normal, 1 for having ST-T wave abnormality, and 2 for showing probable or definite left ventricular hypertrophy following Estes' criteria.

Many studies with the Cleveland dataset have been tested with different machine learning techniques. For example ref. [21] lists different algorithms and performances ranging from 47% to 80% accuracy. SAIN achieved an 82% accuracy score. Why SAIN? The search is fast, uses appropriate distances chosen by a medical expert, and provides explainability at a personal level, including probabilities. It offers different scenarios for modelling by experimenting with different sets of features, parameters, and preferred outcome visualisations.

The SAIN experiment used binary and numerical representation for each variable as described. We have used the same data representation (recommended by medical experts) as in the original paper [21]. The accuracy of the SAIN experiment was 82%, the same accuracy as in [6], which used classical machine learning. In addition, SAIN allows visualising each personalised model as shown in the examples in Figures 3 and 4.

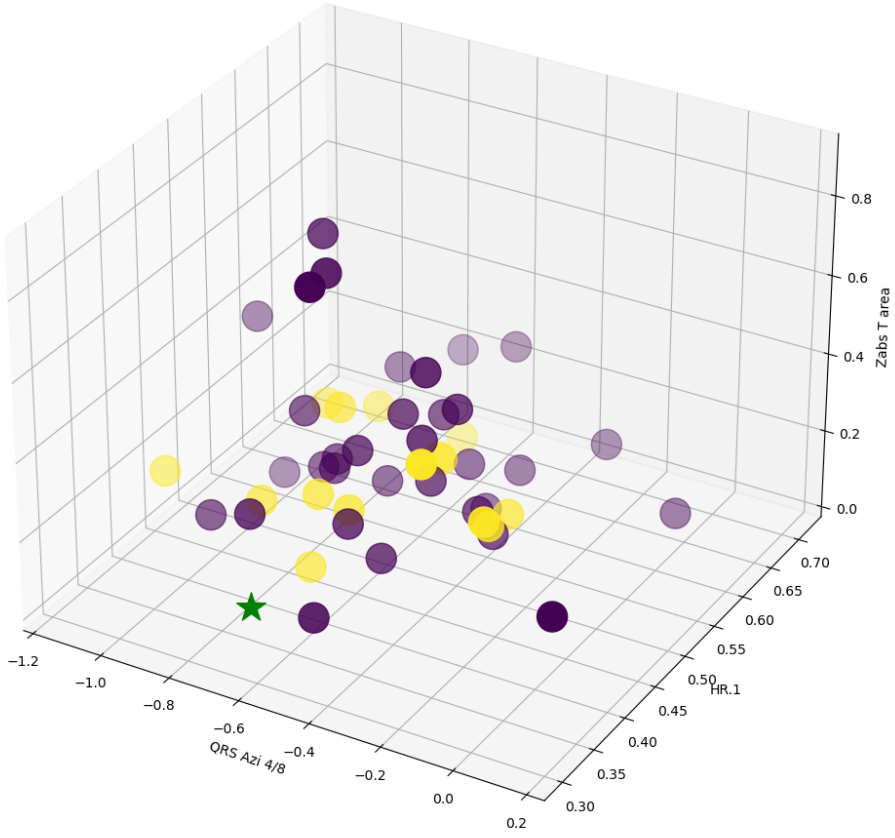


Figure 4. An individual model of survival showing the closest neighbouring samples in the top 3 ranked variables. Where the green star is the individual, the purple denotes class 0 and the yellow class 1.

5.2. Time Series Classification

The proposed SAIN framework can incorporate time series data, as another modality, in addition to other modalities of data for a person, making a joint multimodal personal

vector. A time series data is encoded into the binary vector by using spike encoding algorithms [6], where if there is a positive change from one discrete time point to the next one in the time series, there will be a positive spike (encoded as 1); a negative change will result in a negative spike (−1) and no change will result in 0 value. This is illustrated on a hypothetical time series in Figure 5. This approach applies to any time series raw data, at any time scales, and here we show just two hypothetical examples of brain EEG data (Figure 6) and cardio data (Figure 7).

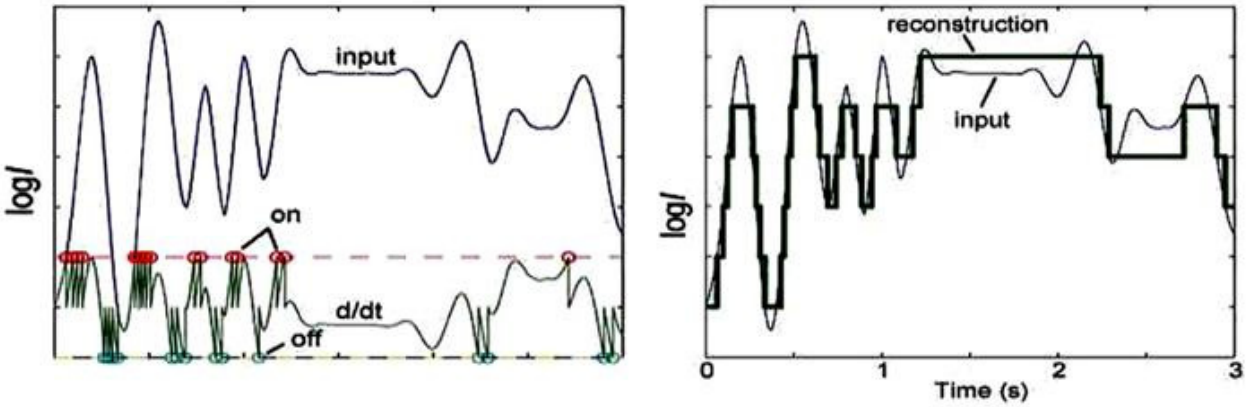


Figure 5. Every time series can be represented as a 3-value vector through a spike encoding method over time [11]. If at a time t the time series is increasing in value, there will be a positive spike (1), if decreasing—a negative spike (−1), and if no change—no spike (0) (**left figure**). Each element in this vector represents the signal change at a time. The original signal can be recovered over time using this vector (**right figure**) if necessary. The length of the vector is equal to the time points measured (reproduced from [11]).

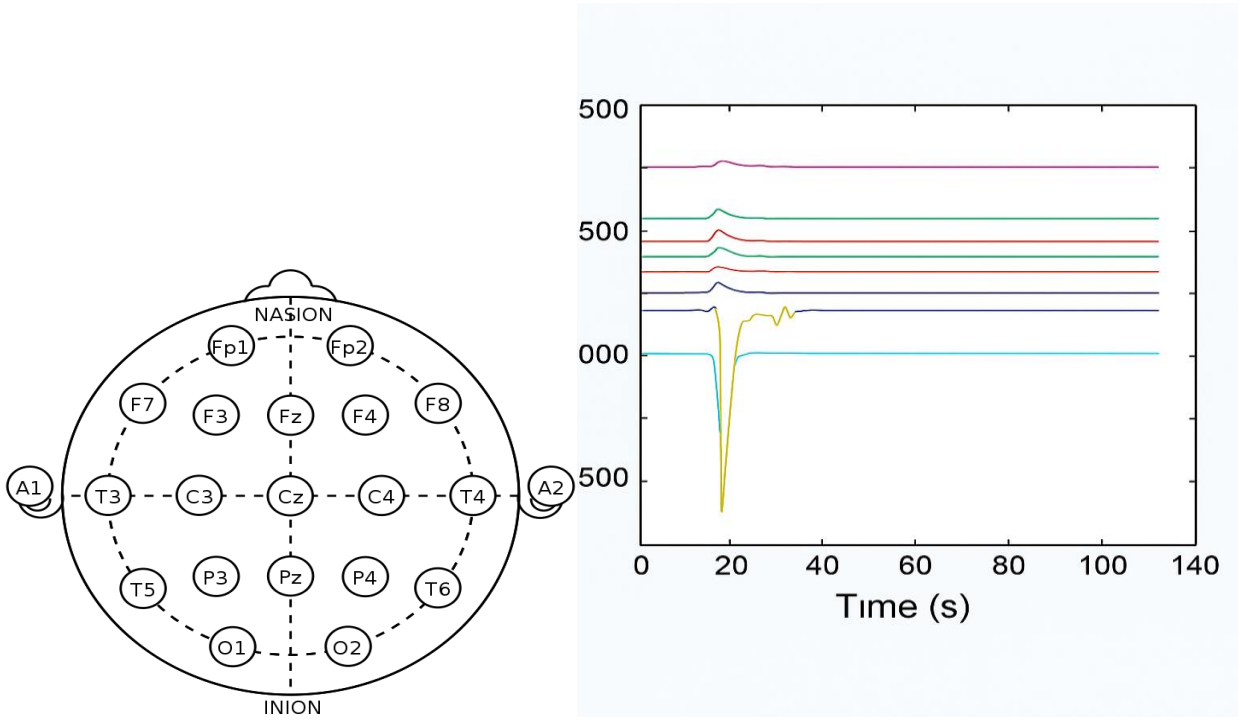


Figure 6. EEG signals from EEG electrodes spatially distributed on the scalp are spatio-temporal signals (**left figure**). Each time series signal from an electrode is measured every 1 millisecond. The figure on the right shows the measurements of 14 EEG electrodes over 124 milliseconds. Each signal can be encoded into a 124-element vector according to Figure 5, making altogether 14 such vectors to be processed in the SAIN framework (reproduced from [11]).

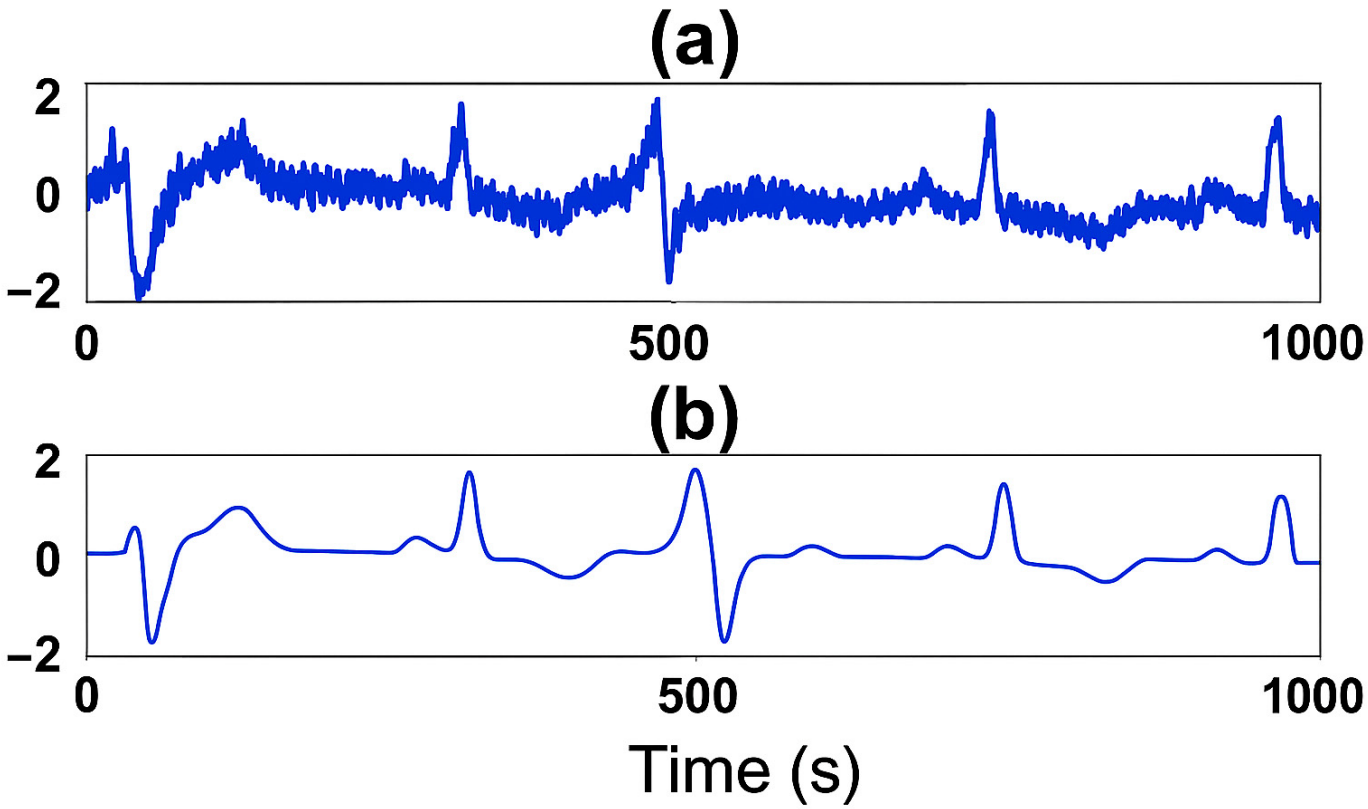


Figure 7. ECG (electrocardiogram) signals ((**a**)—nosy and (**b**)—filtered) can be encoded into binary vectors according to the spike-encoding methods from Figure 5. Spike encoding is robust to noise, as any noise below a threshold would not cause the generation of a spike (either positive or negative), and the encoder will act as a filter. This vector's length will equal the number of measurement time points. The vector data can be further processed in the SAIN framework.

Many datasets for classifying outcomes of events consist of multiple time series. Each variable in a time series may depend on other variables that change in time. The proposed model can deal with this problem by encoding time series (signal) into binary vectors, which can be processed for classification in the SAIN framework. The variables for this dataset are 14 channels of temporal EEG data, located at places of interest on the human scalp.

The signals measured over the same period are the EEG channels, fMRI voxels, ECG electrodes, seismic sensory signals, financial time series, gene expressions, voice, and music frequency bands [11]. Even when the variable (signal) measurements are independent, the signals may impact each other as they represent the same object/person at the same time period. The number N of these signals can vary from just a few for a short time window T (Figure 5) to hundreds and thousands when the time varies from a few milliseconds to minutes, hours, days, etc.

Figure 6 shows an EEG experiment, and Figure 7 shows a cardio-vascular disease signal.

Next, we present a simple example of how this search can be computed for a new record X consisting of only three variables/signals (e.g., EEG channels, ECG electrodes) over a short period of five time moments and the database D consisting of only six such records, which are labelled by outcome labels 1, 2, 3 (e.g., diagnosis, prognosis).

In addition to the record X, a weight vector is supplied with the weighted importance of the signals at different time points, e.g., $W = [0.1, 0.2, 0.4, 0.2, 0.1]$, meaning that the most important and informative part of the measurements is at time point 3.

The new record $X = [1, 1, -1, 0, 1]$ (signal, EEG channel 1) 0, 1, 1, 1, -1 (signal, EEG channel 2) 1, 1, -1, -1, 0 (signal, EEG channel 3), $W = [0.1, 0.2, 0.4, 0.2, 0.1]$.

The database contains records $(Records, R1, R2, R3, R4, R5, LabelsL)$ where (see Table 23):

Table 23. Database of EEG records.

Record	Channel 1	Channel 2	Channel 3	Label
R1	(1, 1, −1, 0, 1)	(0, 1, 1, 1, −1)	(1, 1, −1, −1, 0)	1
R2	(1, 0, −1, 0, 1)	(0, 1, 1, 1, −1)	(1, 0, −1, −1, 1)	1
R3	(1, 1, −1, 0, 1)	(0, −1, 1, 1, −1)	(1, 1, −1, 0, 1)	2
R4	(1, 1, −1, 0, 1)	(0, −1, 1, 0, −1)	(1, 1, −1, 0, 1)	2
R5	(1, 1, −1, 0, 0)	(0, −1, 0, 1, −1)	(1, 1, −1, 1, 1)	3
R6	(1, −1, −1, 0, 1)	(0, −1, 1, 0, −1)	(1, 1, −1, 0, 1)	3

The new record X of EEG signals will be classified in class 1 as it is closest according to the Euclidean distance, with class 1 data samples $R1$ and $R2$.

5.3. Predicting Longevity in Cardiac Patients

We utilised a dataset [22] in which we applied a binary classification on whether the patient had an event (e.g., death) and in addition to those that had an event, whether this would occur in the near future (within the next 180 days, e.g., approximately six months). The dataset contained a set of 150 variables and an outcome, with 295 patients in the first dataset and 49 in the second. The data included a mix of variables that could be grouped as follows:

- Demographics, risk factors, disease states, medication, and deprivation scores;
- Echocardiography, cardiac ultrasound measurements;
- Advanced ECG measurements.

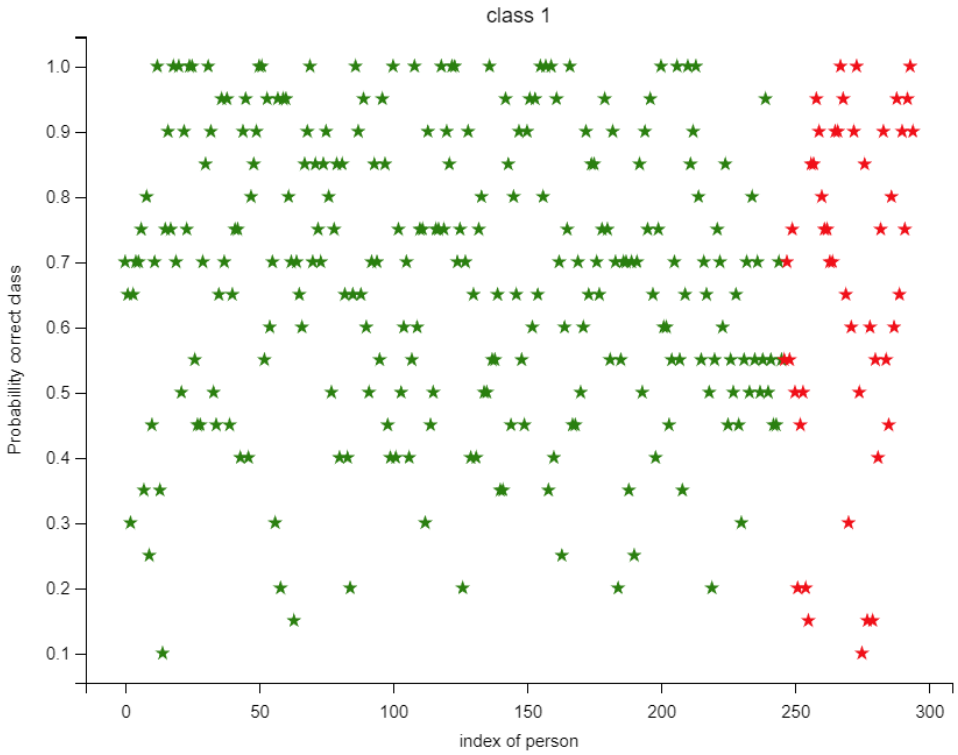
The other data includes the days until the event occurred and the censor date for the Cox proportional hazard monitoring.

The objectives are to predict an arrhythmic event or death.

Before running the algorithm, the data was normalised, and to account for the data being unbalanced, we utilised the SMOTE data balancing method [11] each time we left one out (ensuring that we did not SMOTE when the true data point was part of the dataset). For the event classification dataset, the model achieved an accuracy of 79%. This is broken down into classifying no event (198/247, 80%) and an event with (36/49, 73%) accuracy. It is worth noting that the confidence of each individual could be explored with a sample of the confidence for classification in Figure 8.

For the second experiment, we normalised the dataset and removed any columns with unknown values. We then applied a genetic algorithm to find the set of features to use for classification. We found a set of 34 variables which would provide an accuracy of 81% with (34/34) for class 0 and (6/15) for class 1. Alternatively, if we apply SMOTE and focus more on the accuracy of class 1, we obtain 69% accuracy; however, more evenly distributed with (24/34) for class zero and (10/15) for class 1.

Experiment one, in which we used a non-balanced complete dataset, showed satisfactory results of 80% for class 0 (no event) and 73% for class 1 (event). This demonstrates the ability of the SAIN method to work with imbalanced data. It also shows the superiority of utilising the missing values rather than removing them in experiment two. Furthermore, the selection of variables with a genetic algorithm also showed improvement. The genetic algorithm, included in the SAIN software version 0.1, can also help select biomarker variables in other cases (see Figure 4).



	0	1
0	198.0	13.0
1	48.0	36.0

accuracy: 0.7932203389830509

Figure 8. A sample of the classification breakdown (with class 0 in green and class 1 in red) and the confusion matrix.

6. Discussion

In Table 24, the features that are implemented in the proposed SAIN methodology and framework are compared qualitatively with similar features of already developed methods for personalised modelling. SAIN can deal with an unlimited number of modalities of data, including numbers, time series, images, videos, and digitally encoded elements. It also offers multicriteria metrics for distance measure across variables from different domains, e.g., logical Boolean domain, logical non-Boolean domains, numerical domains with natural and rational values, and binary codes. Furthermore, SAIN processes all modalities of data into a single individual (personalised) vector, which allows the early integration of the modalities and facilitates the discovery of causal associations across modalities to explain individual outcomes. This contrasts with the late integration of modalities [1], where for each modality there is a separate model and their outputs are weighted for the calculation of the final output. Explainability is available in all models in Table 24.

When tested on benchmark and domain problem data, all personalised models have exceeded in accuracy the corresponding global models, where one model is created on all data. In terms of speed of processing, the proposed SAIN method is superior, due to the early integration and the binary representation of most of the modalities.

Table 24. Comparison between SAIN and other existing methods for personalised modelling.

Source	Number of Modalities	Number of Metrics	Types of Data Sets	Type of Integration Explainability	Explainability	Machine Learning Method
[1]	3	1	Longitudinal time series data	Late	No	Liquid Sate Machine
[2]	2	1	Time series; on-line text	Early	Reveals the impact of news on time series	SNN
[3]	2	1	Time series; on-line text	Late	No	DeepNN
[4]	3	1	Brain images – pixel values	Early	Moderate	Deep NN
[5]	2	1	Social and cognitive data as numbers	Early	Feature interaction network	SNN
[6]	2	1	Time and space	Early	Feature interaction network.	SNN
[9,14]	3	1	Personalised vector data of numbers; Time and space	Preliminary selection	Reveals feature interaction over time	SNN
[23,24]	1	1	Numerical vector-based data	No integration	Extracted fuzzy rules	Fuzzy neural networks
[25]	3	1	Time, space and direction: fMRI + DTI data	Early	Feature interaction network	SNN
This paper—SAIN	Multiple, practically unlimited	Multiple, multicriteria metrics	Multiple, practically unlimited	Early	Visualisation and interpretation of personalised model	Statistical: search and inference using wwkNN

7. Conclusions

This paper presents a new search and inference method, called SAIN, for multi-modal data integration and personalised model creation based on these multi-modal data. The model not only evaluates the outcome for a person more accurately than traditional machine learning methods using a single modality of data, but it also explains the proposed solution in terms of probability and visual explanation.

In its current form, this paper is more directed towards revealing a new methodology and algorithms than real full-scale medical applications. However, we have illustrated the methods using hypothetical and real-case health and medical datasets. Further utilisation of the proposed framework is currently being developed for large-scale biomedical data.

The proposed new method offers new functionality and features for personalised search and model creation in multimodal data, some of which are listed below:

- The method is suitable for multimodal data searches in heterogeneous datasets, e.g., numbers, text, images, sound, categorical data.
- It is suitable for personalised model creation to classify or predict specific outcomes based on multimodal and heterogeneous data.
- It uses a similarity measure based on multicriteria metrics. In this way, inaccurate measurement of similarity on a large number of heterogeneous variables is avoided.
- Its search is fast even on large datasets and includes advanced personalised searches with multiple parameters and features.
- It facilitates multiple solutions with corresponding probabilities.
- It is suitable for unsupervised clustering in multimodal heterogeneous data.

In conclusion, integrating all possible data modalities for a single subject to predict/classify the object's state in relation to existing ones is an open problem in data science. While the creation of personalised models based on a single modality data [23] and clustering of single modality data into a single cluster [26] have been successfully developed, the theory, framework, and algorithms proposed in this paper are the first to integrate all data modalities for a single subject together into a single vector-based representation and to make an inference based on it. For the first time, time series, such as EEG and ECG data, are included in this unified representation, after suitable encoding. In this respect, spike encoding of time series is used, integrating statistical and brain-inspired information representation. The human brain integrates sensory data modalities into its spatio-temporal structure, and brain-inspired models using spike information representation have already been developed for learning [6,11,27] and for explanation of the learned patterns [28]. However, brain-inspired computers are still in their early stage of development [11], and even if they are developed, they may not be able to integrate all possible modalities of data into one brain-inspired mode. This paper offers a solution to the problem of multi-modal personalised data integration and inference, with six novel features as follows: (1) it includes all possible modalities of data; (2) it can be implemented on any conventional computer platforms; (3) it takes into account the differences across modalities of data through offering different distance measures; (4) it offers a new way of ranking existing multimodal objects in order of similarity to a new multimodal object and uses that for building multiple neighbourhood clusters; (5) it offers a probability-based inference with the use of the different similarity clusters; (6) it explains the inferred results, both in terms of probabilities and visual representation. The proposed original method here is planned to be applied to large-scale multimodal data for biomedical and health applications in the future.

The proposed method is implemented as a computer system and applied to several case studies to illustrate its advantages and applicability. The SAIN method described in Section 4 was implemented as a software system, as seen in Data Availability Statement.

Author Contributions: N.K. designed the overall framework of SAIN, wrote the initial draft of the paper and took part in the experiments and the paper revision; C.S.C. introduced the mathematical description of the distance measures for different data modalities, took part in the preparation and the revision of the paper; A.H. developed the software implementation of SAIN in Python and ran the experiments, also took part in the paper preparation and its revision; P.G. provided cardio data for experiments and took part in the analysis of results. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data has been obtained from the UCI Cleveland from https://archive.ics.uci.edu/dataset/45/heart+disease (accessed on: 20 February 2024); the EEG data is available from https://github.com/KEDRI-AUT/NeuCube-Py/tree/master/example_data (accessed on 20 February 2024). Access to the software is available upon request.

Acknowledgments: We thank Elena Calude for her contributions to the mathematical model. We also thank the referees for their suggestions, which improved the paper.

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

References

1. Budhraja, S.; Singh, B.; Doborjeh, M.; Doborjeh, Z.; Tan, S.; Lai, E.; Goh, W.; Kasabov, N. Mosaic LSM: A Liquid State Machine Approach for Multimodal Longitudinal Data Analysis. In Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, QLD, Australia, 18–22 June 2023; IEEE: New York, NY, USA, 2023; pp. 1–8. [CrossRef]
2. AbouHassan, I.; Kasabov, N.K.; Jagtap, V.; Kulkarni, P. Spiking neural networks for predictive and explainable modelling of multimodal streaming data with a case study on financial time series and online news. *Sci. Rep.* **2023**, *13*, 18367. [CrossRef] [PubMed]
3. Rodrigues, F.; Markou, I.; Pereira, F.C. Combining time-series and textual data for taxi demand prediction in event areas: A deep learning approach. *Inf. Fusion* **2019**, *49*, 120–129. [CrossRef]
4. Li, J.; Liu, J.; Zhou, S.; Zhang, Q.; Kasabov, N.K. GeSeNet: A General Semantic-Guided Network With Couple Mask Ensemble for Medical Image Fusion. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 16248–16261. [CrossRef]
5. Doborjeh, Z.; Doborjeh, M.; Sumich, A.; Singh, B.; Merkin, A.; Budhraja, S.; Goh, W.; Lai, E.M.; Williams, M.; Tan, S.; et al. Investigation of social and cognitive predictors in non-transition ultra-high-risk'individuals for psychosis using spiking neural networks. *Schizophrenia* **2023**, *9*, 10. [CrossRef]
6. Kasabov, N.K. NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data. *Neural Netw.* **2014**, *52*, 62–76. [CrossRef]
7. Kasabov, N. Global, local and personalised modeling and pattern discovery in bioinformatics: An integrated approach. *Pattern Recognit. Lett.* **2007**, *28*, 673–685. [CrossRef]
8. Kasabov, N. Data Analysis and Predictive Systems and Related Methodologies. U.S. Patent 9,002,682 B2, 7 April 2015.
9. Doborjeh, M.; Doborjeh, Z.; Merkin, A.; Bahrami, H.; Sumich, A.; Krishnamurthi, R.; Medvedev, O.N.; Crook-Rumsey, M.; Morgan, C.; Kirk, I.; et al. Personalised predictive modelling with brain-inspired spiking neural networks of longitudinal MRI neuroimaging data and the case study of dementia. *Neural Netw.* **2021**, *144*, 522–539. [CrossRef]
10. Kasabov, N.K. *Evolving Connectionist Systems*, 2nd ed.; Springer: London, UK, 2007.
11. Kasabov, N.K. *Time-Space, Spiking Neural Networks and Brain-Inspired Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 750.
12. Santomauro, D.F.; Herrera, A.M.M.; Shadid, J.; Zheng, P.; Ashbaugh, C.; Pigott, D.M.; Abbafati, C.; Adolph, C.; Amlag, J.O.; Aravkin, A.Y.; et al. Global prevalence and burden of depressive and anxiety disorders in 204 countries and territories in 2020 due to the COVID-19 pandemic Global prevalence and burden of depressive and anxiety disorders in 204 countries and territories in 2020 due to the COVID-19 pandemic. *Lancet* **2021**, *398*, 1700–1712.
13. Swaddiwudhipong, N.; Whiteside, D.J.; Hezemans, F.H.; Street, D.; Rowe, J.B.; Rittman, T. Pre-diagnostic cognitive and functional impairment in multiple sporadic neurodegenerative diseases. *bioRxiv* **2022**. [CrossRef]
14. Kasabov, N.K.; Hou, Z.; Feigin, V.; Chen, Y. Improved Method and System for Predicting Outcomes Based on Spatio/Spectro-Temporal Data. 2015. Available online: https://patents.google.com/patent/WO2015030606A2/en (accessed on 20 December 2015).
15. Paprotny, D.; Morales-Nápoles, O.; Worm, D.T.; Ragno, E. BANSHEE—A MATLAB toolbox for non-parametric Bayesian networks. *SoftwareX* **2020**, *12*, 100588. [CrossRef]

16. Koot, P.; Mendoza-Lugo, M.A.; Paprotny, D.; Morales-Nápoles, O.; Ragno, E.; Worm, D.T. PyBanshee version (1.0): A Python implementation of the MATLAB toolbox BANSHEE for Non-Parametric Bayesian Networks with updated features. *SoftwareX* **2023**, *21*, 101279. [CrossRef]
17. Mendoza-Lugo, M.A.; Morales-Nápoles, O. Version 1.3-BANSHEE—A MATLAB toolbox for Non-Parametric Bayesian Networks. *SoftwareX* **2023**, *23*, 101479. [CrossRef]
18. Calude, C.; Calude, E. A metrical method for multicriteria decision making. *St. Cerc. Mat* **1982**, *34*, 223–234.
19. Calude, C. A simple non-uniform operation. *Bull. Eur. Assoc. Theor. Comput. Sci.* **1983**, *20*, 40–46.
20. Akhtarzada, A.; Calude, C.S.; Hosking, J. A Multi-Criteria Metric Algorithm for Recommender Systems. *Fundam. Informaticae* **2011**, *110*, 1–11. [CrossRef]
21. Kahramanli, H.; Allahverdi, N. Design of a hybrid system for the diabetes and heart diseases. *Expert Syst. Appl.* **2008**, *35*, 82–89. [CrossRef]
22. Gleeson, S.; Liao, Y.W.; Dugo, C.; Cave, A.; Zhou, L.; Ayar, Z.; Christiansen, J.; Scott, T.; Dawson, L.; Gavin, A.; et al. ECG-derived spatial QRS-T angle is associated with ICD implantation, mortality and heart failure admissions in patients with LV systolic dysfunction. *PLoS ONE* **2017**, *12*, e0171069. [CrossRef]
23. Song, Q.; Kasabov, N. TWNFI–a transductive neuro-fuzzy inference system with weighted data normalization for personalized modeling. *Neural Netw.* **2006**, *19*, 1591–1596. [CrossRef]
24. Song, Q.; Kasabov, N.K. NFI: A neuro-fuzzy inference method for transductive reasoning. *IEEE Trans. Fuzzy Syst.* **2005**, *13*, 799–808. [CrossRef]
25. Sengupta, N.; McNabb, C.B.; Kasabov, N.; Russell, B.R. Integrating space, time, and orientation in spiking neural networks: A case study on multimodal brain data modeling. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 5249–5263. [CrossRef]
26. Kasabov, N. NeuCube EvoSpike Architecture for Spatio-temporal Modelling and Pattern Recognition of Brain Signals. In *Artificial Neural Networks in Pattern Recognition*; Springer Berlin/Heidelberg, Germany, 2012; pp. 225–243. [CrossRef]
27. Kumarasinghe, K.; Kasabov, N.; Taylor, D. Deep learning and deep knowledge representation in Spiking Neural Networks for Brain-Computer Interfaces. *Neural Netw.* **2020**, *121*, 169–185. [CrossRef]
28. Futschik, M.; Kasabov, N. Fuzzy clustering of gene expression data. In Proceedings of the 2002 IEEE World Congress on Computational Intelligence, 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No.02CH37291), Honolulu, HI, USA, 12–17 May 2002; pp. 414–419. [CrossRef]