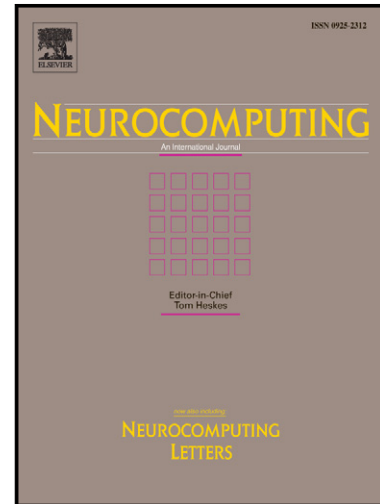


Evolving Spiking Neural Networks for Personalised Modelling, Classification and Prediction of Spatio-Temporal Patterns with a Case Study on Stroke

Nikola Kasabov, Valery Feigin, Zeng-Guang Hou, Yixiong Chen, Linda Liang, Rita Krishnamurthi, Muhaini Othman, Priya Parmar



www.elsevier.com/locate/neucom

PII: S0925-2312(14)00109-X
DOI: <http://dx.doi.org/10.1016/j.neucom.2013.09.049>
Reference: NEUCOM13889

To appear in: *Neurocomputing*

Received date: 17 April 2013
Revised date: 16 August 2013
Accepted date: 16 September 2013

Cite this article as: Nikola Kasabov, Valery Feigin, Zeng-Guang Hou, Yixiong Chen, Linda Liang, Rita Krishnamurthi, Muhaini Othman, Priya Parmar, Evolving Spiking Neural Networks for Personalised Modelling, Classification and Prediction of Spatio-Temporal Patterns with a Case Study on Stroke, *Neurocomputing*, <http://dx.doi.org/10.1016/j.neucom.2013.09.049>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting galley proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Evolving Spiking Neural Networks for Personalised Modelling, Classification and Prediction of Spatio-Temporal Patterns with a Case Study on Stroke

Nikola Kasabov*, Valery Feigin+, Zeng-Guang Hou^, Yixiong Chen^, Linda Liang*, Rita Krishnamurthi+, Muhaini Othman*, Priya Parmar+

* Knowledge Engineering & Discovery Research Institute (KEDRI), Auckland University of Technology

+ Institute for Stroke and Applied Neuroscience (NISAN), Auckland University of Technology

^ China Academy of Sciences Institute for Automation (CASIA), Beijing

Abstract

The paper presents a novel method and system for personalised (individualised) modelling of spatio/spectro-temporal data (SSTD) and prediction of events. A novel evolving spiking neural network reservoir system (eSNNr) is proposed for the purpose. The system consists of: spike-time encoding module of continuous value input information into spike trains; a recurrent 3D SNNr; eSNN as an evolving output classifier. Such system is generated for every new individual, using existing data of similar individuals. Subject to proper training and parameter optimisation, the system is capable of accurate spatio-temporal pattern recognition (STPR) and of early prediction of individual events. The method and the system are generic, applicable to various SSTD and classification and prediction problems. As a case study, the method is applied for early prediction of occurrence of stroke on an individual basis. Preliminary experiments demonstrated a significant improvement in accuracy and time of event prediction when using the proposed method when compared with standard machine learning methods, such as MLR, SVM, MLP. Future development and applications are discussed.

Keywords: Personalised modelling; Spatio-temporal pattern recognition; Spiking neural networks; Evolving connectionist systems; Stroke occurrence prediction.

1. Personalised Modelling

Personalised modelling (PM) is concerned with the creation of an individual model from data to better estimate an unknown outcome for an individual [1, 2]. This is in contrast to global modelling, where a model is created to cover the whole problem space. Personalised medicine and many other areas of science and technology rely now on efficient PM methods. Classical methods, such as kNN, wkNN, wwNN and other have a limited success on complex problems [1,2,13].

In [1] a personalised modelling method and system were proposed as graphically shown in Fig.1a. For every new input vector \mathbf{x} , for which an output needs to be estimated, specific input variables/features (V_x), their weightings (W_x), the number K_x and the neighbourhood of samples (D_x), as well as the model M_x and its parameters (P_x) are optimised together as a common chromosome used in a genetic algorithm (GA) (Fig.1b). This method was explored on different applications in [2,29].

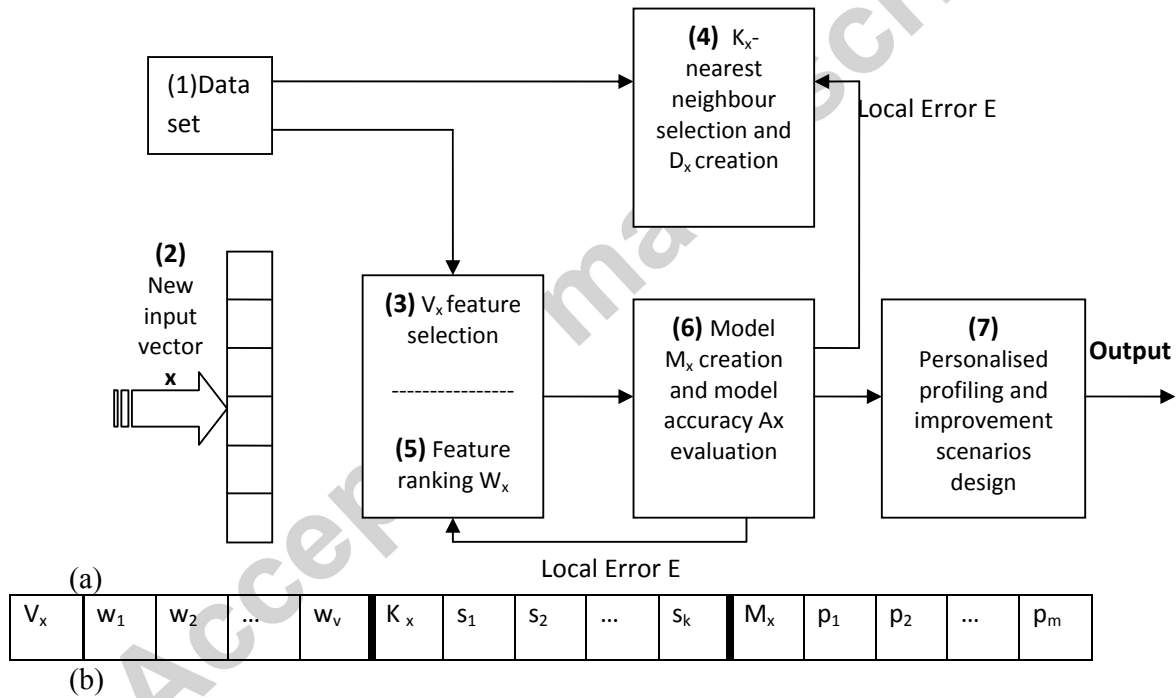


Fig.1a,b: A functional block diagram of a personalised modelling method that optimises each individual model and creates an individual profile (from [1,2]).

This paper extends the method from [1,2] to apply on spatio/spectro-temporal data (SSTD) for classification and for an early prediction of outputs that are results of spatio-temporal patterns from a stream of SSTD. The classification models are proposed to be based on spiking neural networks (SNN), suitable to learn and classify SSTD, rather than on traditional machine learning classifiers that classify static, vector-based data as it is in [1,2].

Spatio- and spectro- temporal data (SSTD), that are characterised by a strong temporal component, are the most common types of data collected in many domain areas. However, there is lack of efficient methods for modelling such data. In particular, there is a need for STPR that can facilitate new discoveries from complex SSTD and produce more accurate and earlier prediction of events. This is especially relevant for an early prediction of serious personal events such as stroke and heart failure, where it would be desirable to predict these events accurately and at the earliest possible time. This is also crucial for brain data processing.

This paper suggests a new method and system to address this challenge. For the proposed method a novel computational architecture is introduced built of spiking neural networks (SNN). This can be used to predict early events in many domains, such as engineering (surveillance for crime prediction and prevention; cyber security), bioinformatics (gene expression and protein folding), neuroinformatics (EEG, fMRI patterns) and its application for BCI and neurorehabilitation, ecology (establishment of species), environment (global warming process), medicine (patients risk of disease or recovery over time), economics (financial time series, macroeconomics).

2. Evolving Spiking Neural Networks for Personalised Modelling, Classification and Prediction of Spatio-Temporal Patterns

To address the problem of PM with SSTD, here we propose a method and a system based on SNN. SNN have already proved that they are superior in learning and capturing spatio-temporal patterns from SSTD [3-19] (see also: <http://ncs.ethz.ch/projects/evospike>). SNN use temporal encoding of data as an internal mechanism to learn temporal relationships between input variables related to a spatio-temporal pattern that needs to be learned, classified and predicted.

The proposed here method for PM based on SNN consists of the following steps:

1. For a new individual vector \mathbf{x} , described by a set of static variables S_x and by dynamic, temporal variables T_x , with unknown output (event) O_x in future times, select the closest individuals to \mathbf{x} according to the static variables from a data repository of individuals with known output and known time of event occurrence with the use of Euclidean or other form of distance measure (e.g. [1,2]).
2. Create an evolving SNN personalised model, denoted as PMeSNNr, to predict the chances of event occurrence for \mathbf{x} using the dynamic data sets of the neighbouring individuals selected in (1) for training the model. After that the temporal data T_x of the individual \mathbf{x} is applied to recall the model and to predict the outcome for \mathbf{x} .
3. Estimate the time in the future when there is a chance for an event to happen to the individual \mathbf{x} , the accuracy of this prediction and the likelihood of the event to happen.

4. Optimise iteratively the features, the neighbourhood and the parameters of the PMeSNNr using the personalised optimisation procedure from [1], or an alternative one, to achieve a maximum accuracy for an earliest possible time of prediction.

The method for the creation of a PMeSNNr model consists of the following steps realised as functional modules in the system shown in fig.2:

- a. Encode SSDT into spike sequences: continuous value input information is encoded into trains of spikes;
- b. Construct and train a recurrent 3D SNN reservoir, SNNr, to learn the spike sequences that represent individual input patterns;
- c. Construct and train an evolving SNN classifier to learn to classify different trajectories of the SNNr activities that represent different input patterns from SSTD that belong to different classes;
- d. Optimise the model – optimise input features, the neighbourhood and the parameters of the MPeSNNr model through iterative applications of steps (a) to (c) above for different parameter values until maximum accuracy is achieved at an earliest time of prediction.
- e. Save the optimised model.
- f. Recall the model for the input data \mathbf{x} .

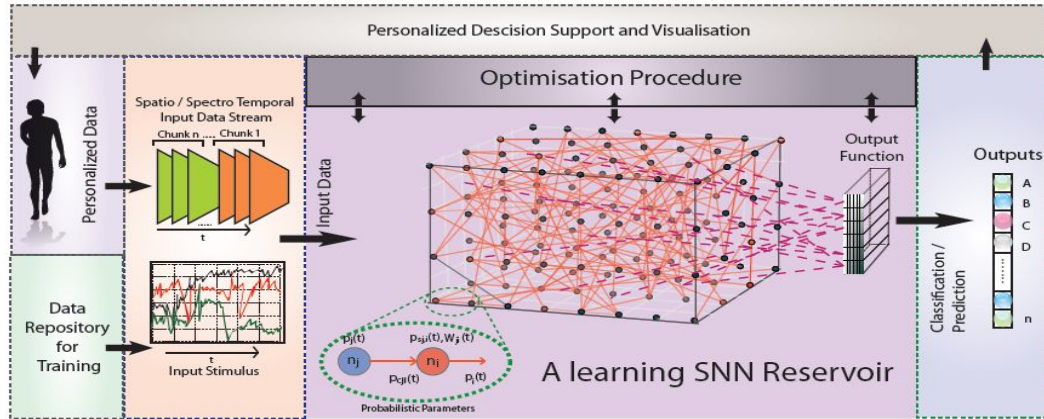


Figure 2. A schematic diagram of the PMeSNNr architecture

The above modules from (a) to (e) are described further in this section.

2.1. Input data encoding module

Possible methods that can be used to encode input SSTD into spike sequences are: Population Rank Coding; Address Event Representation (AER); Ben's Spike Algorithm. Fig.3a shows an example of encoding a single value of input data (shown as a thick vertical line) into a spike train of 5 spikes emitted by 5 neurons based on their receptive fields (Gaussian functions). Fig 3b illustrates the AER coding.

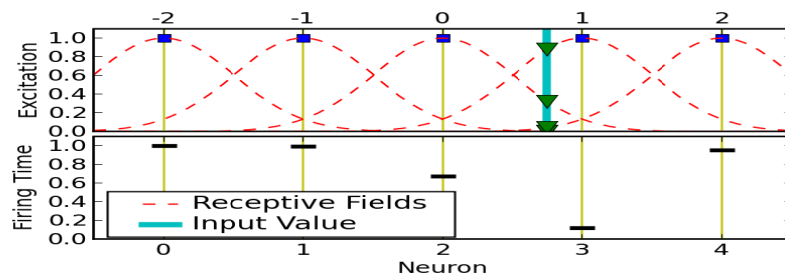


Figure 3a: Population rank-order coding

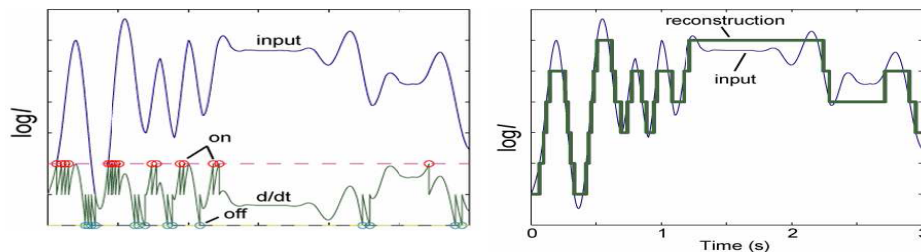


Fig.3b: Address Event Representation (AER) encoding of continuous time series data into spike trains and consecutive recovery of the signal.

These series of input spike trains are entered into spatially located neurons from the SNNr depending on the problem in hand, e.g. : entering input spike sequences into randomly selected neurons from the SSNNr; entering brain data sequences to spatially located neurons that map brain areas where data is collected. Spike trains are continuously fed into the SNNr in their temporal order.

2.2. The SNNr module

After selecting the neighbourhood for an individual \mathbf{x} based on static data, a *learning phase* takes place to build a personalised model $M_{\mathbf{x}}$ as a PMeSNNr. All temporal input data of all neighbours of \mathbf{x} are propagated one by one in the SNNr and learned as whole patterns in it. After a whole input pattern is entered (and learned) in the SNNr, the ‘liquid’ state of the SNNr is measured and an output classifier is trained to recognise this liquid state in a predefined output class for this input pattern. The SNNr has a 3D structure connecting leaky-integrate and fire model (LIFM) spiking neurons with recurrent connections [3-10]. A LIFM of a neuron is illustrated in figure 4.

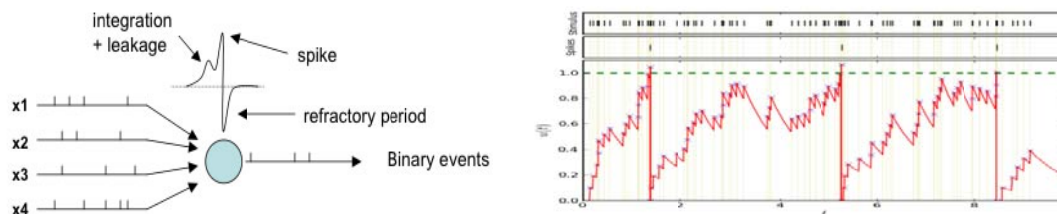


Figure 4. The most popular spiking neuron model is the leaky integrate and fire model

In the SNNr we use spike-time dependent plasticity learning rule (STDP) for the neurons to learn spatio-temporal relationships from the input data and to adapt their connections in order to generate specific trajectories when a particular input pattern is entered [6].

On figure 2 a special class of the LIFM is shown – the probabilistic neuron model [19] that has probability parameters attached to the connections, the synapses and the output of the spiking neuron. Before training, the SNNr connectivity is initialised as small world connections, where closets neurons are connected with a higher probability.

Once spike trains are entered into the SNNr, it acts as a larger dimensional space. The SNNr accumulates temporal information of all input spike trains and transforms them into high-dimensional intermediate ('liquid') states /trajectories that can be measured over time. The recurrent reservoir generates unique accumulated neuron spike time responses for different classes of input spike trains.

As an illustration, figs.5a,b,c show the spiking activity (a) and connectivity of a SNNr before training (b) and after training - (c) on illustrative SSTD. It can be seen that as a result of training new connections have been created that represent *spatio-temporal interaction* between input variables captured in the SNNr from the data. The connectivity can be dynamically visualised for every new pattern submitted.

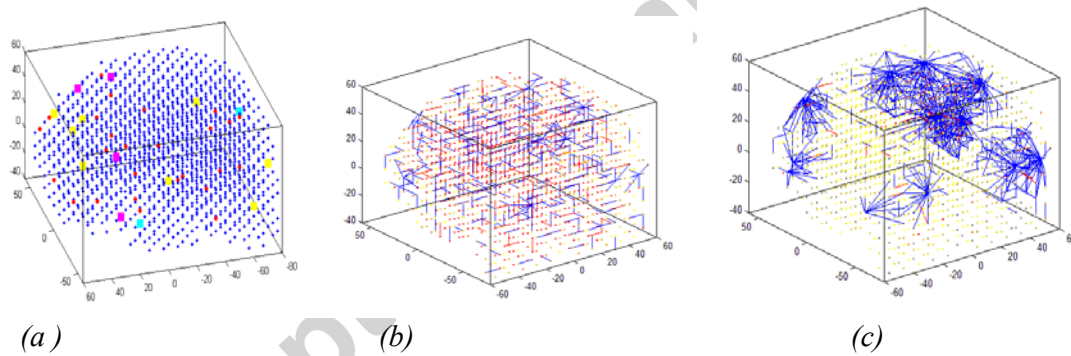


Fig.5a,b,c Illustrative visualisation of connectivity and spiking activity of a SNNr: (a) spiking activity – active neurons are represented in red colour and large size; (b) connectivity before training – small world connections – positive connections are represented in blue and negative – in red; (c) connectivity after training.

2.3. Evolving output classification module

All neurons in the SNNr are connected to an evolving SNN classifier (eSNN) [13-15]. eSNN uses LIFM of a spiking neuron and rank order (RO) learning to achieve fast, one-pass learning with the emphasis of the first incoming spike on a synapse as suggested in [11] and used in [12]. One of the originality of the proposed PMeSNNr method is that it utilises the ability of the eSNN to learn to recognise complex spatio-temporal patterns generated in the SNNr before the whole input data pattern is entered.

Different types of eSNN can be used, including:

- Simple eSNN [13,14,16];
- Dynamic eSNN (deSNN), as introduced in [15], where RO learning is used for

initialisation of a synaptic weight based on the first incoming spike on this synapse, but than this weight is modified based on following spikes using spike time dependent plasticity (STDP) learning rule;

- Spike pattern association neurons (SPAN) as classifiers, where as a reaction to a recognised input pattern, a precise time sequence of spikes is generated at the neuronal output [17,18].

The RO learning rule allows in principle for an eSNN to learn complex spatio-temporal patterns from data streams and then to recognise *early* an incoming pattern (therefore not necessarily ‘waiting’ for the whole pattern to be presented). This paper uses this property of the LIFM and eSNN for building accurate personalised models to predict *early* onset of an event for an individual, based on historical SSTD of many individuals for which the event had happened and on current, temporal data for the new individual.

An eSNN classifier is schematically shown in fig.6. The RO learning is based on the assumption that most important information of an input pattern is contained in earlier arriving spikes [13,14]. It establishes a priority of inputs based on the order of the spike arrival on the input synapses for a particular pattern. RO learning makes use of the information contained in the order of the input spikes. This method has two main advantages when used in eSNN: (1) fast learning (as the order of the first incoming spikes is often sufficient information for recognising a pattern; only one pass propagation of the input pattern may be sufficient for the model to learn it); (2) asynchronous, data-driven processing.

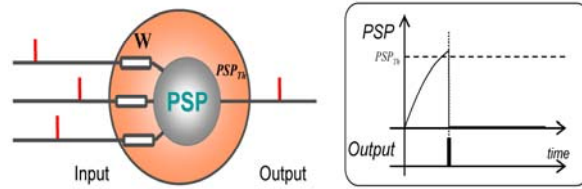


Figure 6: Integrate-and-fire neuron with RO learning

eSNN utilize the principles of evolving connectionist systems [13,20]. An eSNN evolves its structure and functionality in an on-line manner, from incoming information. For every new input data vector, a new output neuron is dynamically allocated and connected to the eSNN input neurons (these are the neurons of the SNNr). The connections are established using the RO rule for the output neuron to recognise this input vector or a similar one as a positive example. The weight vectors of the output neurons represent centres of clusters in the problem space and can be represented as fuzzy rules [21].

In some system implementations, neurons with similar weight vectors are merged based on Euclidean distance between them. That makes it possible to achieve a very fast learning (only one pass may be sufficient), both in a supervised and in an unsupervised mode [13,14].

During a *learning phase* of the classifier, for each M-dimensional input pattern P_i a new output neuron i is created and its connection weights $w_{j,i}$ ($j=1,2,...,M$) to the input neurons are calculated based on the order of the incoming spikes on the corresponding synapses using the RO learning rule :

$$w_{j,i} = \alpha \cdot \text{mod}^{\text{order}(j,i)} \quad (1)$$

where: α is a learning parameter (as a partial case, it is equal to 1); mod is a modulation factor, that defines how important the order of the first spike is; $w_{j,i}$ is the synaptic weight between a pre-synaptic neuron j and the postsynaptic neuron i ; $order(j,i)$ represents the order (the rank) of the *first* spike at synapse j,i ranked among all spikes arriving from all synapses to the neuron i ; $order(j,i)$ has a value 0 for the first spike to neuron i and increases according to the input spike order at other synapses.

While the input training pattern (example) is presented (all input spikes on different synapses, encoding the input vector are presented within a time window of T time units), the spiking threshold Th_i of the neuron i is defined to make this neuron spike when this or a similar pattern (example) is presented again in the recall mode. The threshold is calculated as a fraction (C) of the total PSP_i (denoted as PSP_{imax}) accumulated during the presentation of the whole input pattern:

$$PSP_{imax} = \sum_{(j)} mod^{order(j,i)} \quad (2)$$

$$Th_i = C \cdot PSP_{imax} \quad (3)$$

If the C parameter has a value of 1 it means that during a recall phase the neuron would need the whole pattern to be propagated in order for the neuron to recognise this pattern and to emit an output spike. A value of 0 will make this neuron spike immediately (and definitely wrongly) after the first input spike is entered into the neuron. But *modifying* C from a value of 1 down to 0 (not reaching it) will make it possible for the neuron, once trained on the whole input pattern, to emit a spike if only part of it is presented during recall. And it has to be a correct classification which requires a careful optimisation of the C value in order to achieve correct and earliest spiking activity of the output neurons. This is a key feature of the proposed method that extends the previous art from [1,2].

Here, optimising C , means how much early an output class will be recognised after the start of a spatio-temporal data pattern entry. And the value for C will be different for different classes and different individuals, i.e. it will be individualised/personalised.

If the weight vector of the evolved and trained new output neuron is similar to the one of an already trained neuron (in a supervised learning mode for classification this is a neuron from the same class), i.e. their Euclidean distance is less than a similarity threshold Sim , the new neuron will be merged with the most similar one, averaging the connection weights and the threshold of the two neurons [13,14]. Otherwise, the new neuron will be added to the set of output neurons (or the corresponding class pool of neurons when a supervised learning for classification is performed). The similarity between the newly created neuron and a training neuron is computed with the use of the Euclidean distance between weight matrices of the two neurons. The merged neuron has weighted average weights and thresholds of the merging neurons. While an individual output neuron represents a single input pattern, merged neurons represent clusters of patterns or prototypes in a transformed spatial – RO space. These clusters can be represented as fuzzy rules [21].

The eSNN learning is evolving, adaptive, incremental, theoretically – ‘lifelong’, so that the system can learn new patterns through creating new output neurons, connecting them to the input neurons, and possibly merging the most similar ones.

During the *recall phase*, when a new input vector is presented to the eSNN, the spiking pattern is submitted to all created output neurons during the learning phase. An output spike is generated by neuron i at a time l if the $PSP_i(l)$ becomes higher than its threshold Th_i . After the first neuron spikes, indicating a recognised output class, the PSP of all neurons are set to initial value (e.g. 0) to prepare the system for the next pattern for recall or learning.

The postsynaptic potential $PSP_i(l)$ of a neuron i at time l is calculated as:

$$PSP_i(l) = \sum_{t=0,1,2,\dots,l} \sum_{(j)} e_j(t) \cdot \text{mod}^{\text{order}(j,i)} \quad (4)$$

where: $e_j(t)=1$ if there is a *first* spike at time t on synapse j , otherwise it is 0; $\text{order}(j,i)$ is the rank order of the first spike at synapse j among all input spikes to neuron i for this recall pattern.

The recall procedure can be performed using different recall algorithms implying different methods of comparing input patterns for recall with already learned patterns in the output neurons:

- (a) Spikes of the new input pattern are propagated as they arrive to all trained output neurons and the first one that spikes (its PSP is greater than its threshold) defines the output. The assumption is that the neuron that best matches the input pattern will spike earlier based purely on the PSP (membrane potential). This method is called eSNNm.
- (b) The second method implies a creation of a new output neuron for each recall pattern, in the same way as the output neurons were created during the learning phase, and then – comparing the connection weight vector of the new one to the already existing neurons using Euclidean distance. The closest output neuron in terms of synaptic connection weights is the ‘winner’. This method uses the principle of transductive reasoning and nearest neighbour classification in the connection weight space. It compares spatially distributed synaptic weight vectors of a new neuron that captures a new input pattern and existing ones. This method is called eSNNs.

Recently, the eSNN was extended to a dynamic eSNN (deSNN) model with the two modifications – deSNNs and deSNNm [15]. In addition to the initialisation of the connection weights based on the first arriving spikes at a synaptic input, the deSNN model adjusts dynamically these connections based on following spikes on the same synaptic inputs using a version of the STDP learning rule (spike-time dependent plasticity) – SDSP (spike-dependent synaptic plasticity). A small drift of a synaptic weight is used to increase the weight if there is a spike, or decrease it if there is no spike, at each of time moments of simulation.

Both deSNNm and deSNNs can be used for early event prediction in the following ways:

- deSNNm, through the C parameter, but that does not reflect linearly into the time scale, e.g. if $C=0.75$ that means that it takes approximately 75% of the input pattern to classify it, but that does not mean that correct classification will occur 25% earlier in time than the time of the whole pattern presentation, as most of the information is stored in the connection weights as a result of the first spikes at the beginning of the input pattern.
- deSNNs, through either learning with the use only of certain percentage of the patterns in terms of time (e.g. 75%), or through applying a similarity parameter, which means how similar the newly created output neuron should be to the already existing ones in order to classify this neuron (input pattern) into the class of the most similar one.

The main advantage of the eSNN, when compared with other supervised or unsupervised learning and classification SNN models, is that it is computationally inexpensive and boosts the importance of the order in which input spikes arrive, thus making the eSNN suitable for on-line learning and early prediction of temporal events.

2.4. The optimisation module

For each personalised model, to select an optimal subset of features (weather variables), a number of neighbouring sample (K) and parameters of the classifier (for the eSNN they are Mod , Sim and C), different values for all these parameters are tested in their combination using a genetic algorithm, whose chromosome is represented in Figure 1b [1,2]. Different

modifications of genetic algorithms can be used, e.g. Gravitational Search Algorithm [30] as experimented in [29], quantum inspired genetic algorithms [31], etc.

2.5. Recall of the personalised model for the input data x

During a *recall phase* the stream of temporal data, which is related to the individual x , is continuously entered into the SNNr and the SNNr spiking activity sequences are continuously fed into the output classifier until the classifier recognises a certain output class (event) – an output neuron, allocated for a particular class, spikes first. The ability of the eSNN classifiers [14-16], once trained on whole input patterns, to recognise similar input patterns in a recall phase when only initial spikes appear on its synapses, makes eSNN very suitable for the task of early onset prediction of a personal event.

2.6. Implementation of the PMeSNNr

A current implementation of PMeSNNr is based on the NeuCube architecture [34]. NeuCube is designed to map brain data into spatially allocated neurons in the SNNr so that the neurons map the spatial location of the sources of the brain data (e.g. EEG channels). The NeuCube architecture can be used for both brain data and other SSTD depending on the mapping of the SSDT as trains of input spikes into the spiking neurons from the SNNr.

3. Personalised Stroke Occurrence Prediction Based on the PMeSNNr

3.1. Problem and data specification

The stroke occurrence prediction problem can be formulated as follows: Given historical data, both static and temporal, of persons who had stroke in the past, can we predict accurately and early enough what the risk of stroke occurrence for a new person will be in a near future time? This and other stroke related problems have been described and studied in the past with limited success [22,23,24,25,26].

In this paper we have used a personalised data set of persons who had stroke in the past. The experiment uses weather data and the data of stroke occurrence obtained from the results of an international collaborative study carried out under the auspices of six population regions: Auckland (NZ), Perth and Melbourne (Australia), Oxfordshire (UK), Dijon (France), Norrbotten and Vasterbotten (Northern Sweden). The study areas are grouped in the Southern Hemisphere (Auckland, Perth, and Melbourne) and Northern Hemisphere (Oxfordshire, Dijon, Norrbotten and Vasterbotten counties). The data has a static component and dynamic component.

The complete dataset consists of 11,453 samples (all with first occurrence of stroke). Each sample is described by 9 features/variables (4 static patient clinical features and 5 temporal weather features) along with the geographic region. Patient static clinical features (categorical data) are: age, gender, history of hypertension, smoking status, geographical region. Weather (temporal) features (continuous daily data) are: Temperature (Degree Celcius); Humidity (Percentage); Atmospheric Pressure (kPA); Wind Speed (Knots) and Wind Chill (Degrees Celsius). All of these weather parameters were measured over a 60-day period preceding data of stroke (including the day of stroke occurrence as the last day).

Case-crossover design is used in our experiment. It is a longitudinal study which represents a special situation in which no control group exists for separate comparison. In effect, each subject serves as both 'patient' and 'control' at different time intervals, precluding covariate

imbalance. This design has been widely applied in many medical and health studies. Mukamal and his colleagues [27] used this approach to compare measures of weather and ambient air pollution on the day of stroke presentation and on other days (as control) for each patient. Vlak [28] applied case-crossover design to identify trigger factors for rupture of intracranial aneurysms.

Similar conditions exist in our case study dataset where the case-crossover design is applied to the data as all subjects are “stroke” patients with the absence of healthy subjects (normal/control group). The period spanning 29 days pre-stroke occurrence until the stroke event (30 days of time window) is considered as the “stroke” group. This is considered as the critical time window potentially contributing to risk of stroke. For the same participants, period spanning another 30 days of time window from day 30-59 pre-stroke occurrence as the “normal/control” group, due to the assumption that weather parameters 60 to 30 days prior had no influence on the actual stroke event. The separate “normal/control” time should represent the expected distribution of exposure for follow-up time that does not result in a an event. Fig.7. shows the wind speed variable over the 60 days before stroke for several patients of age 60 and 68 belonging to the selected subset. It can be seen that during the 30 days before stroke the changes of the wind speed are different from those during 60 days before and this may help to learn and predict a stroke event.

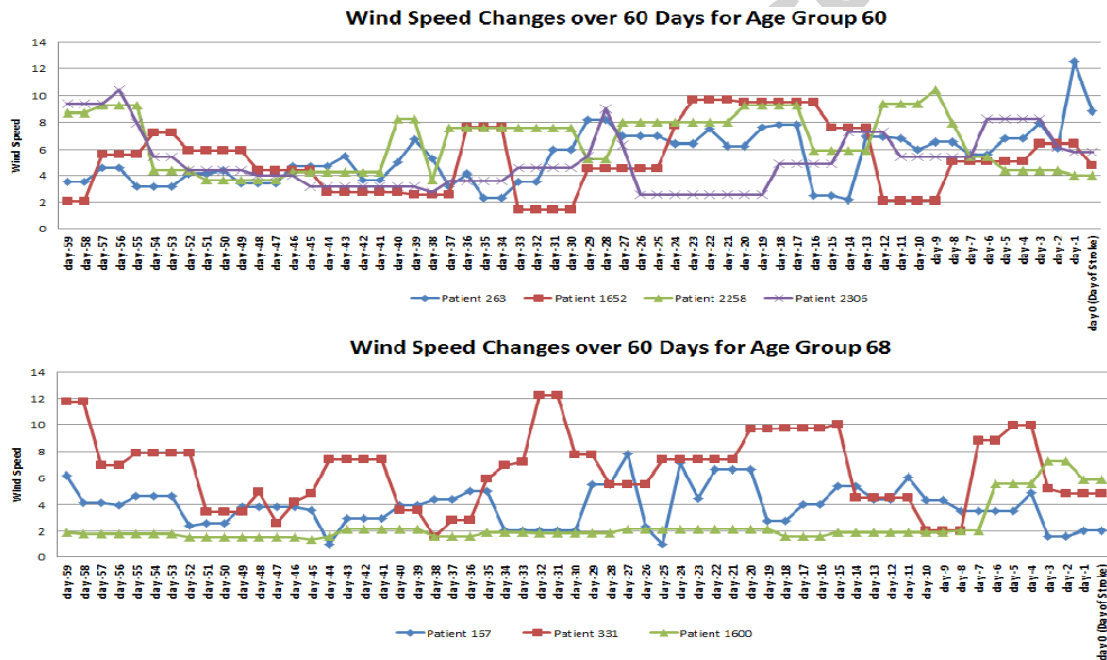


Fig.7. The wind speed variable over the 60 days before stroke for several patients of age 60 and 68 belonging to the selected subset of patients. It can be seen that during the 30 days before stroke the changes of the wind speed are more significant that may help the model to learn and predict the stroke event.

3.2. Experimental settings and results

To illustrate the proposed PMeSNNr method and system and to compare it with other methods, here we have performed personalised modelling in the following way:

- For a new individual, select the nearest neighbour individuals from the data repository using static data (geographic region, age, gender, history of hypertension, smoking status).

- Create and *optimise* a PMeSNNr for this individual using the temporal weather data of the nearest individuals as ‘stroke’ and ‘control’ samples as explained above.
- Recall continuously the model on the individual climate data with a sliding window of 1 day. If an output neuron, that is trained in the eSNN classifier to recognise the class ‘stroke’ is activated (fires a spike), this indicates a risk for a stroke for this person and the number days ahead of the most probable day to happen can be calculated using the optimised value of the parameter C for this person and this data.

As a case study we have tested personalised models for individuals from the following neighbourhood: Auckland region; autumn season; age between 60 and 69; experience of hypertension; current smokers. The number of the individuals in this cluster is 20. Since a case-crossover design is applied, 40 samples have been used, 20 in the “normal/control” group and 20 in the “stroke” group derived from the 20 individuals data. Here we demonstrate that the PMeSNNr framework provides much higher accuracy than conventional machine learning methods. In addition, the PMeSNNr will provide early event prediction.

Experimental results

Here we use the NeuCube architecture [34] with an AER method for encoding the input data. The SNNr consists of 1471 LIFM spiking neurons. We apply STDP learning and a deSNN classifier (see fig.8). The following parameter values were selected for optimal classification accuracy:

- threshold for the AER 0.1 (applied on normalised input weather data in the range [0,1]); it means that when the value of input variable increases above 0.1 at consecutive intervals of time, there will be a positive spike generated and entered into the SNNr; if there is a decrease larger than 0.1, there will be a ‘negative’ spike generated, i.e. a spike sent to an inhibitory neuron from the SNNr; and if there is no change of the value of the input variable or a change less than 0.1 there is no spike generated.
- SNNr has 1471 neurons.
- Small World Connectivity (SWC) used to initialise the connections in the SNNr, with a radius of initial connections of 0.15. The initial connections are generated probabilistically, so that closer neurons are more likely to be connected. The excitatory connections (with small positive weights) are 80% versus the 20% of inhibitory connections (small negative weights).
- threshold of the LIFM neurons in the SNNr is 0.5;
- the leak parameter of the LIFM neurons is 0.002;
- STDP learning rate is 0.01;
- Mod parameter of the deSNNs classifier is 0.04 and the drift is 0.25.

The obtained best accuracy of the designed and trained PMeSNNr with the described above parameter values is 94% (88% for the TP - stroke prediction, class 2; 100% for the TN – no stroke – class 1) using 60% of the data for training and 40 % for testing, randomly selected – fig.8.

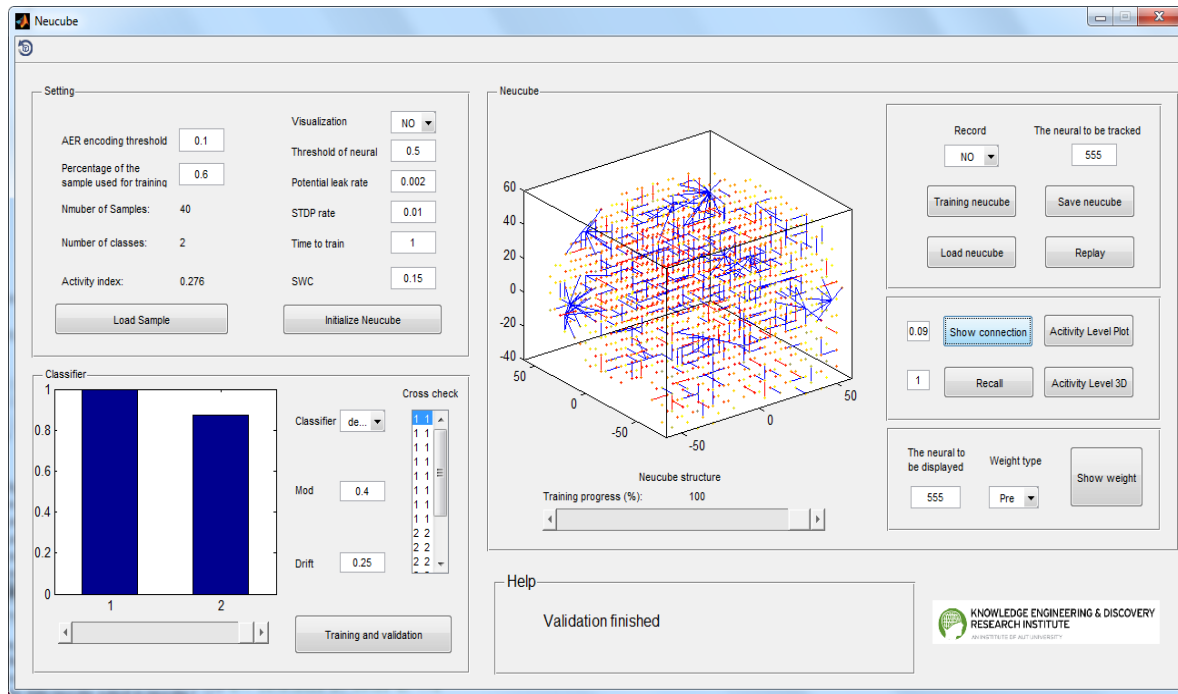


Fig.8. A snapshot from the NeuCube implementation of the PMeSNNr applied on selected nearest neighbourhood of 40 samples of stroke data (20 stroke cases and 20 control) from the case study problem.

Table 1 shows a comparative analysis on the same data set of 40 samples between classical machine learning methods using whole input patterns of 30 days (prediction of just one day ahead) versus an optimal PMeSNNr model that uses only part of the input data to predict stroke events. The PMeSNNr offers a much earlier and a much accurate prediction. The SVM method uses a Polynomial kernel of first degree and 10 variables (5 variables times 30 days). The MLP model uses 150 inputs, 20 hidden nodes and one output, with a learning rate of 0.01 and 500 iterations. The PMeSNNr model uses only one iteration for training, produces a significantly more accurate prediction results and much earlier (fig.8).

Table 1. Comparative analysis of classical machine learning methods using whole SSTD input patterns versus the proposed PMeSNNr model that uses only part of the input data to predict stroke much earlier and much accurately. The data set consists of 40 samples of 5 temporal weather variables. The samples are selected based on 4 static variables.

Method	Overall accuracy (%)	TP – stroke prediction (%)	TN – no stroke (%)
Multiple Linear regression (MLR)	67.50	65	70
SVM	72.5	65	80
MLP	87.5	85	90
PMeSNNr	94	88	100

3.3. Future experiments and impact

Other experiments of applying the proposed methodology for stroke prediction include integrating the presented above stroke data with solar eruption and geomagnetic data [32]. Future in depth analysis of the whole cohort of individual-participant stroke data and various environmental factors (weather characteristics, air pollution, geomagnetic activity) collected

over the last 25-30 years from the six participating centers (NZ, Australia, France, UK, Sweden) will allow more accurate individualised prognostic modelling stratified by various patient (e.g. gender, age groups, presence or absence of hypertension etc) or environmental characteristics (e.g. Northern Hemisphere vs Southern Hemisphere).

Applying the methodology on stroke occurrence data from different geographic regions can improve significantly the population health and reduce the extremely high cost for the society. In China, for example, 2mln people suffer of stroke occurrence every year, 20% of which die. Most of the rest never recover completely.

This paper will enable the creation of on-line available decision support systems for a personal risk evaluation of stroke occurrence, specific for different geographical regions, such as New Zealand, China, USA, Europe, subject to data available.

4. Conclusions and future work

The paper introduces a novel method and system for PM on SSTD based on SNN. The novelty of the proposed method and system are in the following:

- (1) It is the first personalised modelling method and system developed to deal with SSTD without defining in advance the 'time lags' of the time series data.
- (2) The same paradigm, spiking information processing, used in the brain efficiently for STPR tasks is used here to represent and to process SSTD in the form of spiking neural networks (SNN).
- (3) The system is *evolving*, as previously unknown classes could be added incrementally as a result of new SSTD patterns being learned and recognised, which is also a principle of brain cognitive development.
- (4) The system always retains a *transparent* spatio-temporal memory that can be mined and interpreted either in real time or retrospectively for new knowledge discovery.
- (5) The system is able to recognise and predict the outcome of a new SSTD pattern that is similar to previously learnt ones even before the new pattern is fully presented to its inputs.
- (6) For the first time a system is created for early stroke occurrence prediction.

Other applications of the proposed PMeSNNr method and system will include in the future:

- Ecological data modelling and event prediction [33];
- Cardio-vascular risk of occurrence prediction;
- Integrated fMRI-, EEG- and gene brain data modelling;
- Earthquake and other environmental events prediction;
- Financial market crash prediction;
- Machine failure prediction;
- Personal gambling addiction prediction.

There are several avenues to be followed for a future improvement of the model. Further experiments are needed and the model performance needs to be analysed in terms of both accuracy and time of event prediction in relation to different types of input SSTD and different parameter values. The model is indeed very sensitive to parameter values. The results may differ dramatically due to small changes of the parameters. Another problem that needs to be addressed in the future is the initialisation of the connectivity of the SNNr. Different connectivity may result in different accuracy and time of prediction. For an optimal performance, the model needs to be run several times and the best performed model saved for recall. These are new degrees of freedom in the learning model that need to be properly

utilised for better performance. The eSNNr are complex systems and they have to be treated and utilised as such. This is in contrast to the simple and much limited in terms of SSTD classical machine learning methods, such as SVM, MLR, MLP.

The complexity of the eSNNr systems require an optimisation of a model through multiple runs for different parameter values, always retaining the best performed model. This is a procedure that here is realised through a GA as explained in section 2, but a further improvement in terms of reducing the time for optimisation is needed.

The paper demonstrates that it is possible to design much more accurate, although more complex, models for personalised modelling on SSTD than the ones developed with the use of traditional machine learning techniques. The new models can be used for an early prediction of individual events, such as the chosen case study problem of stroke prediction.

Acknowledgement

This paper is supported by a grant from the NZ Ministry of Business, Innovation and Enterprise for Strategic Research Alliance with the China Academy of Sciences Institute for Automation (CASIA). It is also supported by internal grants of the Knowledge Engineering and Discovery Research Institute (KEDRI, <http://www.kedri.info>) and the NISAN Institute of the Auckland University of Technology and also by the CASIA. We acknowledge the help from Stefan Schliebs, Raphael Hu from KEDRI, and the assistance by the organisers of this special issue.

References

- [1] Kasabov, N. (2008) Data Analysis and Predictive Systems and Related Methodologies—Personalised Trait Modelling System, New Zealand Patent No. 572036, PCT/NZ2009/000222, NZ2009/000222-W16-79.
- [2] Kasabov, N., & Hu, Y. J. (2010). Integrated optimisation method for personalised modelling and case studies for medical decision support. *International Journal of Functional Informatics and Personalised Medicine*, 3(3), 236–256.
- [3] Hodgkin, A. L. and A. F. Huxley (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117: 500–544.
- [4] Gerstner, W. (1995) Time structure of the activity of neural network models, *Phys. Rev* 51: 738–758.
- [5] Maass, W. and A. M. Zador. Computing acreate nd learning with dynamic synapses. In *Pulsed neural networks*, pages 321–336. MIT Press, 1999.
- [6] Kistler, G., and W. Gerstner, *Spiking Neuron Models - Single Neurons, Populations, Plasticity*, Cambridge Univ. Press, 2002.
- [7] Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons? *IEEE TNN*, 15(5)1063–70.
- [8] Izhikevich, E.M.: Polychronization: Computation with Spikes. *Neural Computation* 18, 245–282 (2006)
- [9] Belatreche, A., Maguire, L. P., and McGinnity, M. Advances in Design and Application of Spiking Neural Networks. *Soft Comput.* 11, 3, 239–248, 2006
- [10] Brette R., et al, (2007). Simulation of networks of spiking neurons: a review of tools and strategies. *J. Comput. Neurosci.* 23, 349–398.
- [11] Thorpe, S. and J. Gautrais, “Rank order coding,” *Computational Neuroscience: Trends in research*, vol. 13, pp. 113–119, 1998.

- [12] Masquelier, T., R. Guyonneau and S. J. Thorpe (2008) Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains. *PLoS ONE* 3 (1) e1377.
- [13] Kasabov, N., (2007) *Evolving connectionist systems*, Springer, London (first edition 2002).
- [14] Wysoski, S., L. Benuskova, N. Kasabov, *Evolving spiking neural networks for audiovisual information processing*, *Neural Networks*, vol. 23, 7, pp 819-835, 2010.
- [15] Kasabov, N., Dhoble, K., Nuntalid, N. and Indiveri, G. *Dynamic Evolving Spiking Neural Networks for On-line Spatio- and Spectro-Temporal Pattern Recognition*, *Neural Networks*, Volume 41, May 2013, Pages 188–201
- [16] Schliebs, S. and N.Kasabov, *Evolving spiking neural networks: A Survey*, *Evolving Systems*, Springer, vol.4, 2, 87-98, 2013.
- [17] Mohemmed, A. and S.Schliebs and S.Matsuda and N. Kasabov, *SPAN: Spike Pattern Association Neuron for Learning Spatio-Temporal Sequences*, *International Journal of Neural Systems*, Vol. 22, No. 4 (2012) 1-16, 2012.
- [18] Mohemmed, A., S. Schliebs, S. Matsuda and N. Kasabov, *Training Spiking Neural Networks to Associate Spatio-temporal Input-Output Spike Patterns*, *Neurocomputing*, Volume 107, 1 May 2013, Pages 3–10.
- [19] Kasabov, N. *To spike or not to spike: A probabilistic spiking neuron model*, *Neural Networks*, 23(1), 16–19, 2010.
- [20] Watts, M. (2009) *A Decade of Kasabov's Evolving Connectionist Systems: A Review*, *IEEE Trans. SMC - Part C: Applications and Reviews*, vol.39, no.3, 253-269.
- [21] Soltic and S. Kasabov, N. (2010), "Knowledge extraction from evolving spiking neural networks with rank order population coding," *Int. J. Neural Systems*, 20:6, pp. 437-445.
- [22] Feigin, V. L., Barker-Collo, S., Parag, V., Lawes, C. M. M., Ratnasabapathy, Y. & Glen, E. (2010). *Auckland stroke outcomes study. Part 1: Gender, stroke types, ethnicity, and functional outcomes 5 years poststroke*. *Neurology*, 75 (18), 1597-1607.
- [23] Feigin, V. L., Carter, K., Hackett, M., Barber, P., McNaughton, H. & Dyal, L.(2006). *Ethnic disparities in incidence of stroke subtypes: Auckland regional community stroke study, 2002-2003*. *The Lancet Neurology*, 5 (2), 130-139.
- [24] Feigin, V. L., Lawes, C., Bennett, D. & Anderson, C. (2003). *Stroke epidemiology: A review of population based studies of incidence, prevalence, and case fatality in the late 20th century*. *The Lancet Neurology*, 2 (1), 43-53.
- [25] Feigin, V. L., Lawes, C., Bennett, D., Barker, S. & Varsha, P. (2009). *Worldwide stroke incidence and early case fatality reported in 56 population based studies: A systematic review*. *The Lancet Neurology*, 8 (4), 355-369.
- [26] Barker-Collo, S., Feigin, V.L., Parag, V., Lawes, C. M. M., & Senior, H. (2010). *Auckland Stroke Outcomes Study*. *Neurology*, 75(18), 1608-1616.
- [27] Mukamal, K. J., Wellenius, G. A., Suh, H. H. & Mittleman, M. A. (2009). *Weather and air pollution as triggers of severe headaches*. *Neurology*, 72, 922-927.
- [28] Vlak, M., Rinkel, G., Greebe, P., van der Bom, J. & Algra, A. (2011). *Trigger factors and their attributable risk for rupture of intracranial aneurysms: A case-crossover study*. *Stroke*, 42, 1878-1882.
- [29] Liang, Wen, Yingjie Hu, Nikola Kasabov (2013) *Evolving Personalized Modeling System for Integrated Feature, Neighborhood and Parameter Optimization utilizing Gravitational Search Algorithm*. *Evolving System*. Springer, Accepted, 2013.
- [30] Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). *GSA: A gravitational search algorithm*. *Information Sciences*, 179, 2232–2248.
- [31] Defoin-Platel, M., S.Schliebs, N.Kasabov, *Quantum-inspired Evolutionary Algorithm: A multi-model EDA*, *IEEE Trans. Evol. Computation*, vol.13, No.6, Dec.2009, 1218-1232

- [32] Feigin, V. L., Nikitin, Y. & Vinogradova, T. (1997). Solar and geomagnetic activities: Are there associations with stroke occurrence? *Cerebrovascular Diseases*, 7 (6), 345-348.
- [33] Schliebs, Michael Defoin Platel, Susan Worner and Nikola Kasabov, Integrated Feature and Parameter Optimization for Evolving Spiking Neural Networks: Exploring Heterogeneous Probabilistic Models, *Neural Networks*, 22, 623-632, 2009.
- [34] Kasabov, N. (2012) NeuCube EvoSpike Architecture for Spatio-Temporal Modelling and Pattern Recognition of Brain Signals, in: Mana, Schwenker and Trentin (Eds) *ANNPR*, Springer LNAI 7477, 2012, 225-243.







Prof. Nikola Kasabov, FIEEE, FRSNZ is the Director of the Knowledge Engineering and Discovery Research Institute (KEDRI) at the Auckland University of Technology. He is a Past President and Governor Board member of the International Neural Network Society (INNS) and of the Asia Pacific Neural Network Assembly (APNNA). He is a Distinguished Lecturer of the IEEE CIS. He is a Co-Editor-in-Chief of the Springer journal *Evolving Systems* and has served as Associate Editor of *Neural Networks*, *IEEE TrNN*, *IEEE TrFS*, *Information Science*, and other journals. Kasabov holds MSc and PhD from the TU Sofia, Bulgaria. His main research interests are in the areas of neural networks, intelligent information systems, soft computing, bioinformatics, neuroinformatics. He has published more than 510 publications. Among the awards Prof. Kasabov has received are: APNNA 'Outstanding Achievements Award', the INNS Gabor Award for 'Outstanding contributions to engineering applications of neural networks', the EU Marie Curie Fellowship, Distinguished Visiting Fellowship of the Royal Academy of Engineering, UK, the Bayer Science Innovation Award, the RSNZ Science and Technology Medal. He has supervised to completion 35 PhD students. More information of Prof. Kasabov can be found on the KEDRI web site: <http://www.kedri.aut.ac.nz>.




Professor Valery Feigin is Professor and Director of the National Institute for Stroke and Applied Neurosciences (NISAN), Faculty of Health and Environmental Sciences at the AUT University of Auckland. He graduated in medicine from the Novosibirsk Medical University, Russia, and undertook advanced training in neurology and clinical epidemiology in Mayo Clinic, Rochester, MN, USA and Erasmus University, Rotterdam, The Netherlands. Professor Feigin's prime research interest is in the epidemiology, prevention and management of stroke and traumatic brain injury. He has published over 350 research articles in leading international and local medical journals (including *The Lancet* and *The Lancet Neurology*), and eleven stroke handbooks and twelve book chapters. Professor Feigin is Editor-in-Chief of the journal *Neuroepidemiology* and a member of the Editorial Boards of 10 international medical journals. He currently leads the Stroke Experts Panel of the Global Burden of Disease Project that contributes data to the WHO regular reports distributed all over the world. He is also a Director of the Board of Directors of the World Stroke Organization and a member of the Advisory Working Group on Stroke for the WHO ICD-11 version.



Prof. Zeng-Guang Hou received the B.E. and M.E. degrees in electrical engineering from Yanshan University, Qinhuangdao, China, in 1991 and 1993, respectively, and the Ph.D. degree in electrical engineering from Beijing Institute of Technology, Beijing, China, in 1997. He is a Professor and Deputy Director of the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. From September 2003 to October 2004, he was a Visiting Professor with the Intelligent Systems Research Laboratory, College of

	<p>Engineering, University of Saskatchewan, Saskatoon, SK, Canada. His current research interests include robotics and intelligent control systems with applications to life and health science automation. Dr. Hou currently serves as an Editorial Board Member of the Neural Networks, International Journal of Intelligent Systems Technologies and Applications, Journal of Intelligent and Fuzzy Systems, and Acta Automatica Sinica. He was an Associate Editor of the IEEE Computational Intelligence Magazine and IEEE Transactions on Neural Networks, and the Editor of Computational Intelligence Society E-Letter.</p> <p>-----</p> <p>Yixiong Chen is a PhD student at the Key Laboratory for Intelligent Systems of the Chinese Academy of Sciences Institute of Automation, Beijing. His study is on Neurorehabilitation robotic systems, under the supervision of Prof. Hou.</p>
	<p>Wen Liang (Linda) received her Master's degree at AUT in 2009 with First Class (with Honors). She is a PhD student at AUT under the supervision of Prof. Nik Kasabov and Prof. Valery Feigin. Her research interests are in the areas of Personalized Modeling of Brain Data and Spiking Neural Networks. Linda was awarded the Vice-Chancellor's Doctoral Scholarship by the AUT University in 2011, and the Graduate Assistantship Award 2010 from the Faculty of Design and Creative Technologies. As part of her study, she published two book chapters in Springer Handbook of Bio-and Neuroinformatics (HBBNI), two conference papers at the ICONIP 2011 and 2008 conference, a poster at the NZBIO Conference 2010, a short paper at the ICNN 09 congress in Munich.</p>
	<p>Dr Rita V. Krishnamurthi is a Senior Research Fellow who leads the Cerebrovascular Diseases Research Programme at National Institute for Stroke and Applied Neurosciences (NISAN), Faculty of Health and Environmental Sciences, Auckland University of Technology. She is particularly interested in primary stroke prevention using behaviour change models and identification of ethnic specific stroke risk factors. She received her PhD degree from the University of Auckland where she also worked prior to joining AUT.</p>
	<p>Muhaini Othman received her Bachelor in Information Technology from Northern University of Malaysia and Master in Computer Science from University of Malaya in 1999 and 2006 respectively. She is currently a PhD student under the supervision of Prof Nikola Kasabov, Dr Russel Pears, Dr Dave Parry and Dr Rita Krishnamurti. Her research interests include Personalised Modelling, Spiking Neural Network and the representation of spatial and temporal data in ontology-based system for personalised decision support.</p>

	<p>Dr Priya Parmar is a lecturer and biostatistician for the National Institute for Stroke and Applied Neurosciences (NISAN), Faculty of Health and Environmental Sciences at AUT University. She completed a BSc specialising in Bioinformatics and MSc in Medical Statistics from the University of Auckland. She then worked at Queensland Institute for Medical Research before embarking on a PhD at the University of Western Australia (UWA). Priya's PhD utilised methods from genetic epidemiology, biostatistics and the Developmental Origins of Health and Disease (DOHaD) paradigm to identify genetic precursors underlying hypertension. She currently works on the BIONIC and Stroke and Weather projects at NISAN. She is involved in all statistical aspects of projects from study design for trials to conventional and non-conventional analyses to identify associations between disease outcomes and effects.</p>
-----------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------