Full course DLU, 2023, Lecture Content

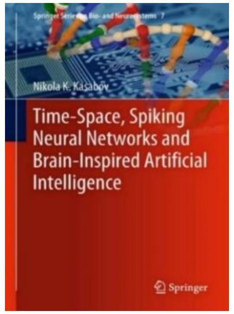
Advanced Artificial Intelligence Technologies and Applications

Lecture 16. AI in finance and economics

1. AI and the evolution of its principles. Evolving processes in Time and Space (Ch1, 3-19)
2. From Data and Information to Knowledge. Fuzzy logic. (Ch1,19-33 + extra reading)
3. Artificial neural networks - fundamentals. (Ch2, 39-48). Computational modelling with NN. Tut1: NeuCom.
4. Deep neural networks (Ch.2, 48-50 + extra reading).
5. Evolving connectionist systems (ECOS) (Ch2, 52-78). Tutorial 2: ECOS in NeuCom.
6. Deep learning and deep knowledge representation in the human brain (Ch3)
7. Spiking neural networks (Ch4). Evolving spiking neural networks (Ch5)
8. Brain-inspired SNN. NeuCube. (Ch.6). Tutorial 3: NeuCube software (IA)
9. From von Neuman Machines to Neuromorphic Platforms (Ch20 , 22)
10. Other neurocomputers: Transformers.
11. Evolutionary and quantum inspired computation (Ch.7)
12. AI applications for brain data: EEG, fMRI (Ch.8-11)
13. Brain-computer interfaces (BCI) (Ch.14)
14. AI applications for audio-visual information (Ch.12,13). AI for language modelling.
15. AI in bioinformatics and neuroinformatics (Ch15,16, 17,18)
16. AI in finance and economics (Ch19)
17. AI applications for multisensory environmental data (Ch19). Revision of the course.

iabouhassan@tu-sofia.bg (Doct. Iman AbouHassan) nkasabov@aut.ac.nz (Prof. N.Kasabov)

Course book: N.Kasabov, Time-Space, Spiking Neural Networks and Brain-Inspired Artificial Intelligence Springer, 2019, https://www.springer.com/gp/book/9783662577134



Additional materials: https://www.knowledgeengineering.ai/china

ZOOM link for all lectures: https://us05web.zoom.us/j/4658730662?pwd=eFN0eHRCN3o4K0FaZ0lqQmN1UUgydz09

AI in finance and economics

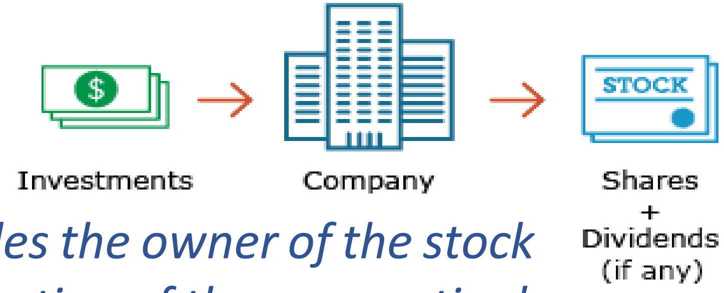
NeuCube

iabouhassan@tu-sofia.bg; nkasabov@aut.ac.nz; https://www.knowledgeengineering.ai/china

AI in finance and economics

stockX

a security that represents the ownership of a fraction of a corporation.

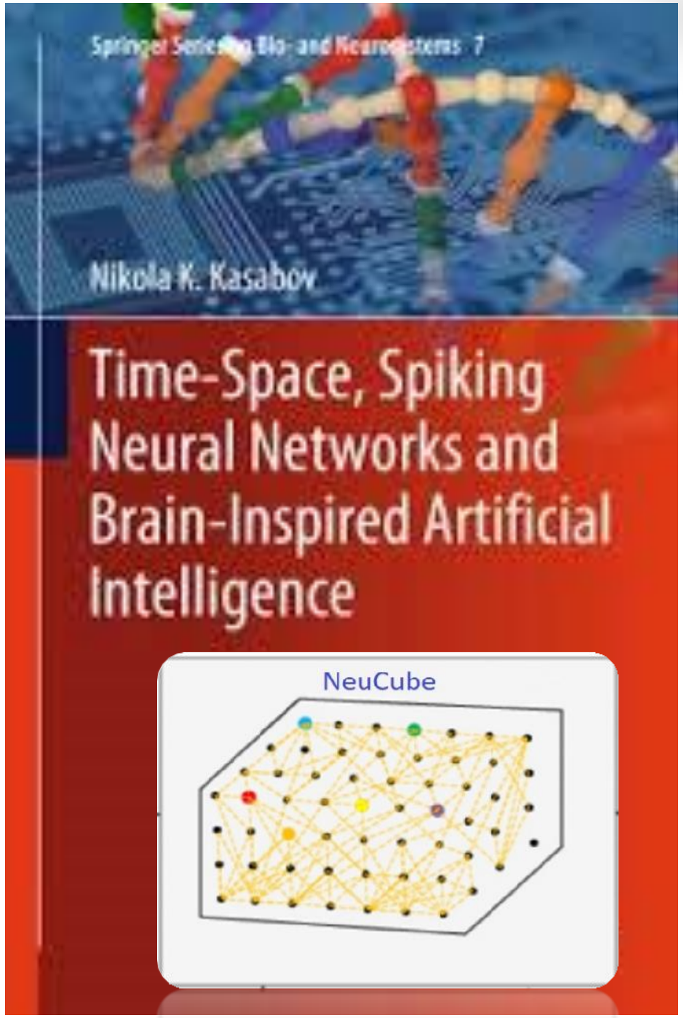


This entitles the owner of the stock to a proportion of the corporation's assets and profits equal to how much stock they own.

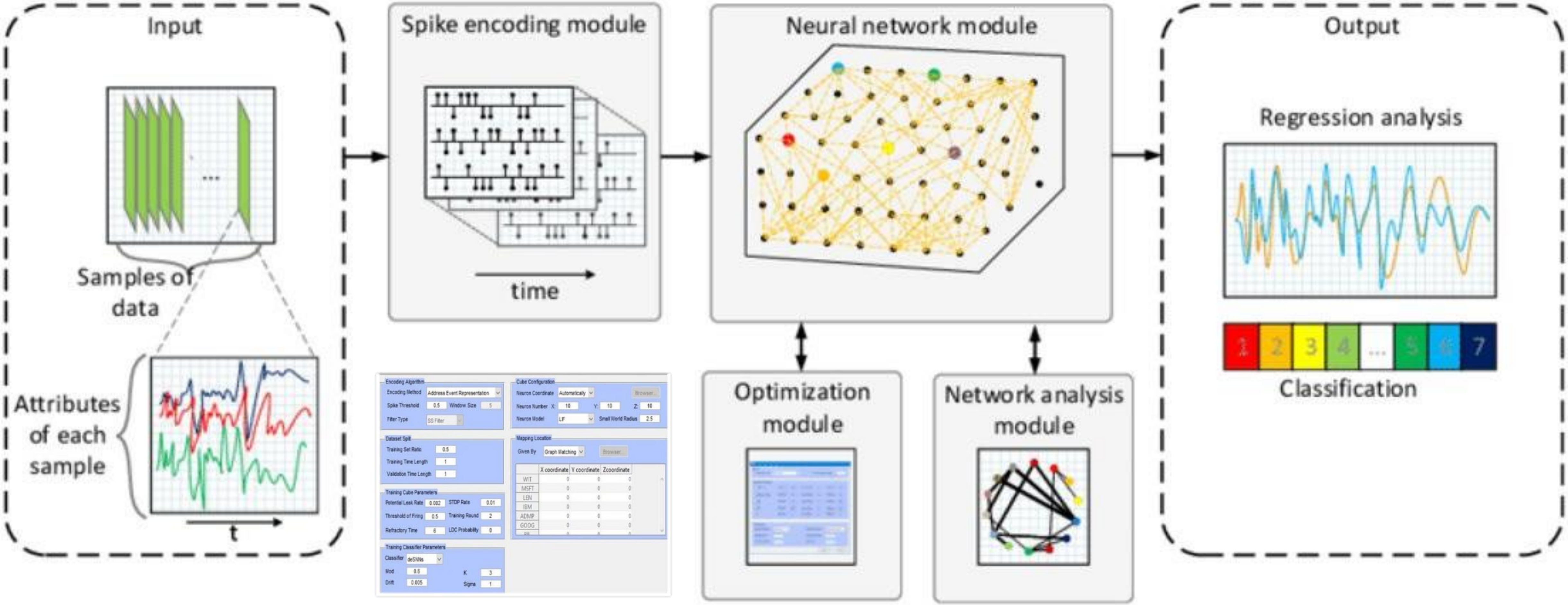
In April 2021, Ford stock fell by 10.4% despite the company's solid core business and impressive quarterly results that exceeded Wall Street expectations (www.cnb

The NeuCube Brain Inspired Spiking Neural Network is a Generic Spatio-temporal Data Machine that allows:

- Mapping temporal variables,
- Learning their temporal interaction,
- Capturing informative patterns,
- Visualizing temporal data relationships,
- Improving prediction accuracy,
- Allowing incremental and evolving learning abilities,
- Outperforming other traditional statistical and machine learning techniques.



NeuCube Architecture



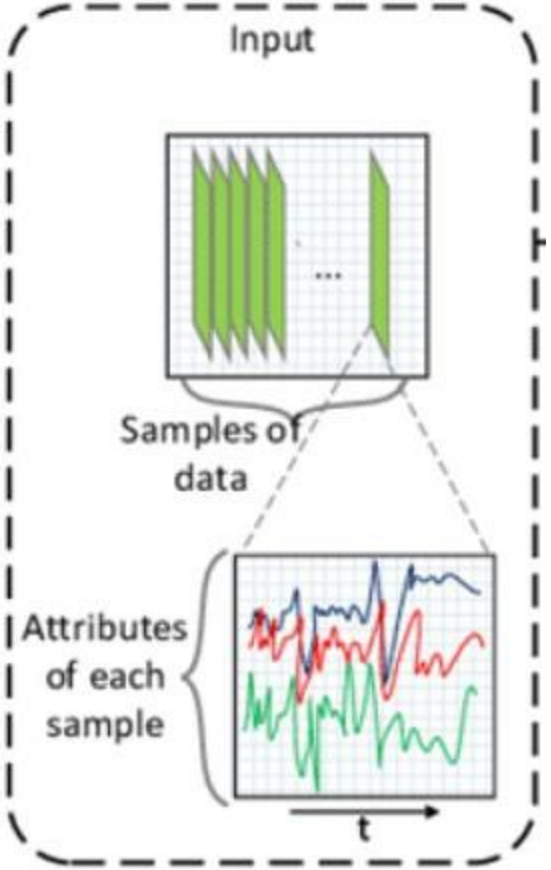
Data Selection

Stock market

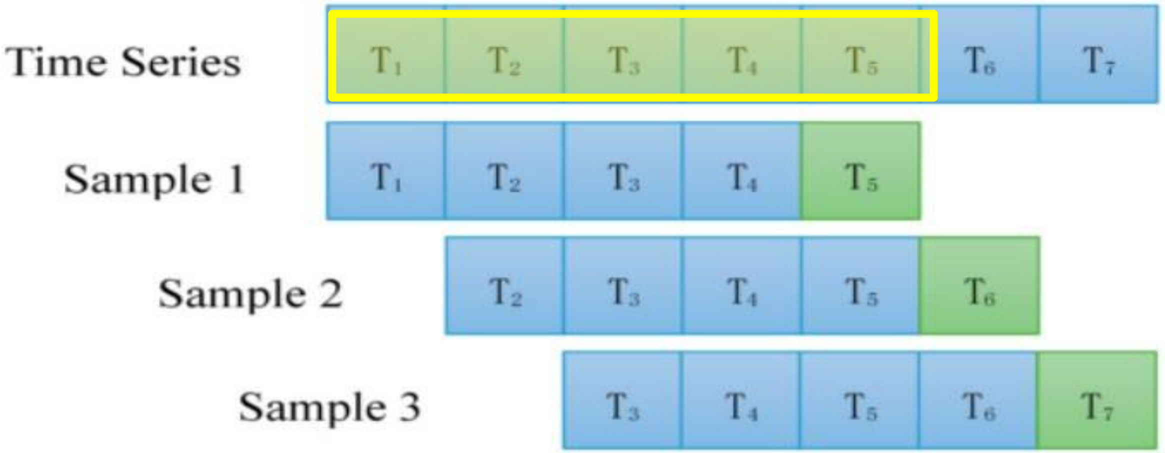
Data Description

- Stock indices (spatial features): Apple Inc., Google, Intel Corp, Microsoft, Yahoo, and **NASDAQ**.
- Original dataset (temporal features): 150 daily observations for 6 variables.
- Sample generation: 50 Samples, each of which contained 100 timed sequences of daily closing prices.
- New dataset = 30,000 data point (5,000 observations for 6 variables)
- The target values representing the closing price of ***NASDAQ*** at the next day are arranged in a column in the target file.

Data Preprocessing

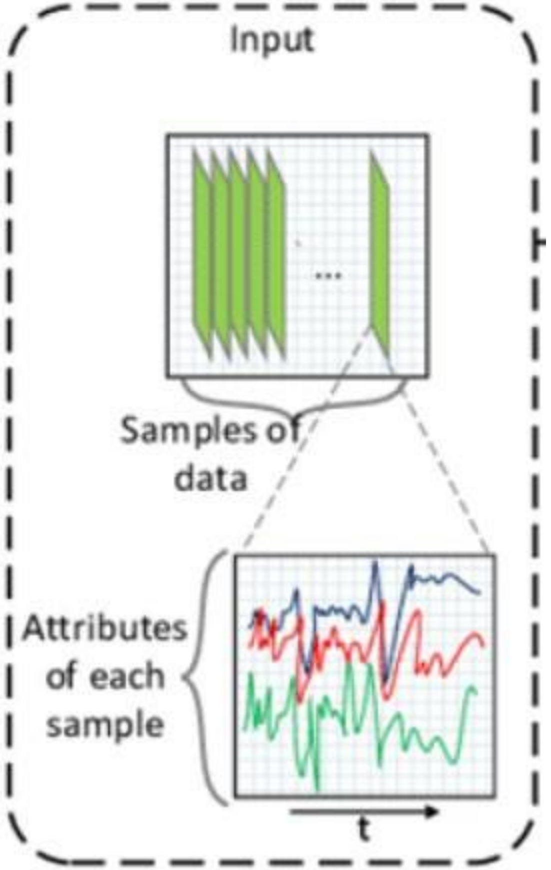


Sliding Window

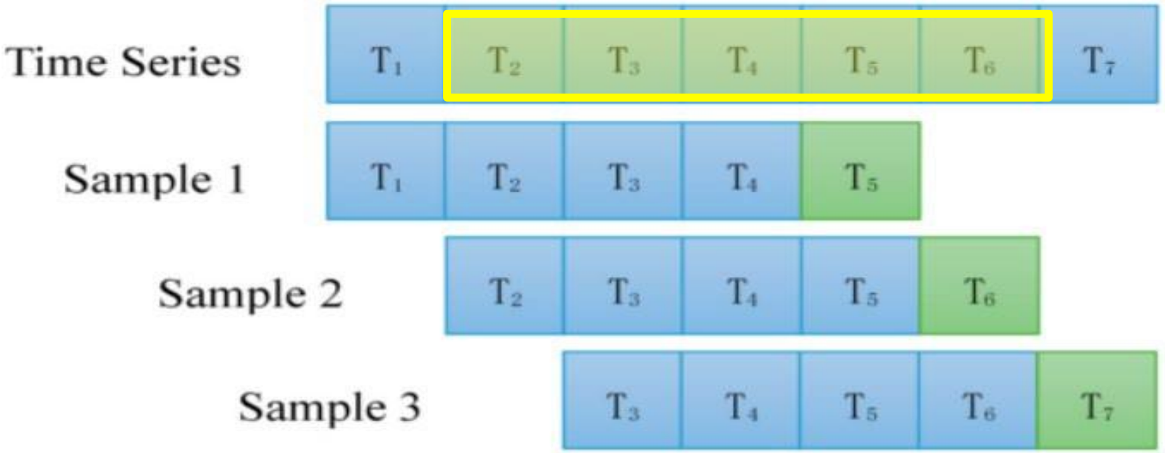


- The original dataset is converted into 50 sample files using NeuCube architecture.
- Each sample is organized as a matrix, with temporal features (rows) represented by 100 ordered days; and the spatial features (columns) represented by 6 input stocks.
- A sliding window approach segments the original dataset into equal sized samples with a sliding step of one day. Historical data are used to feed, learn, and test model.

Data Preprocessing

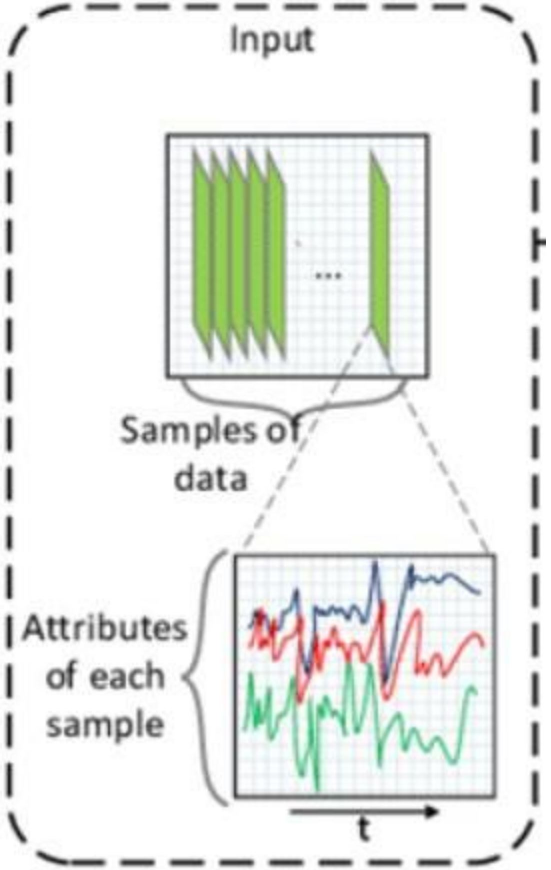


Sliding Window

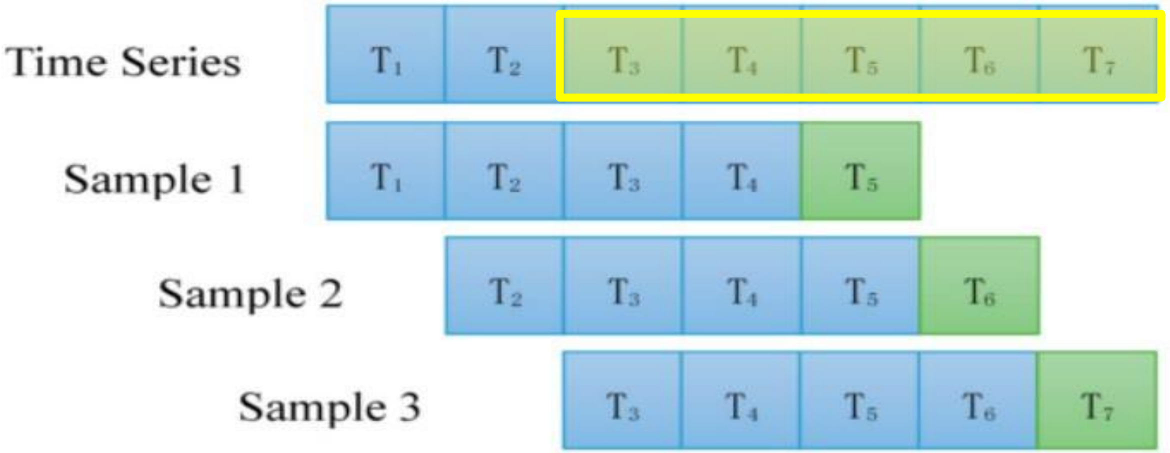


- The original dataset is converted into 50 sample files using NeuCube architecture.
- Each sample is organized as a matrix, with temporal features (rows) represented by 100 ordered days; and the spatial features (columns) represented by 6 input stocks.
- A sliding window approach segments the original dataset into equal sized samples with a sliding step of one day. Historical data are used to feed, learn, and test model.

Data Preprocessing



Sliding Window

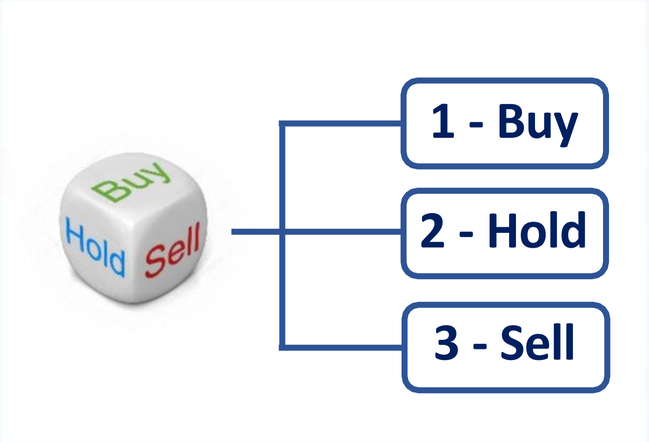


- The original dataset is converted into 50 sample files using NeuCube architecture.
- Each sample is organized as a matrix, with temporal features (rows) represented by 100 ordered days; and the spatial features (columns) represented by 6 input stocks.
- A sliding window approach segments the original dataset into equal sized samples with a sliding step of one day. Historical data are used to feed, learn, and test model.

Output variable: the target

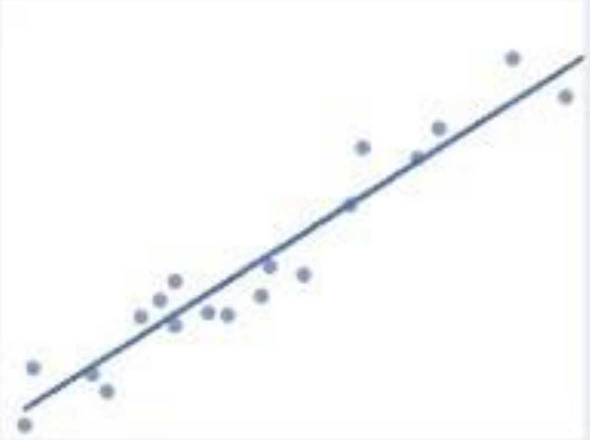
Classification vs Regression

Investment options for NASDAQ

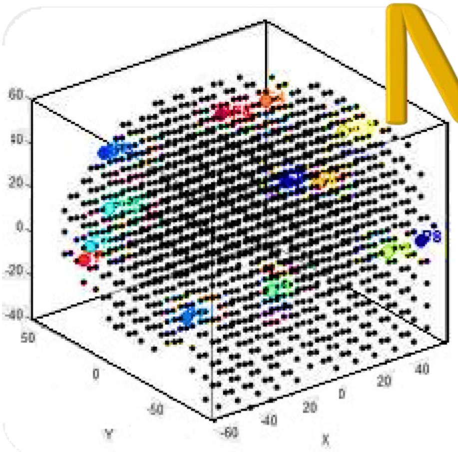


Y = 1 or 2 or 3

Next day's closing price of *NASDAQ*



$Y(t) = f\{Y(t-1), X_i(t)\}$

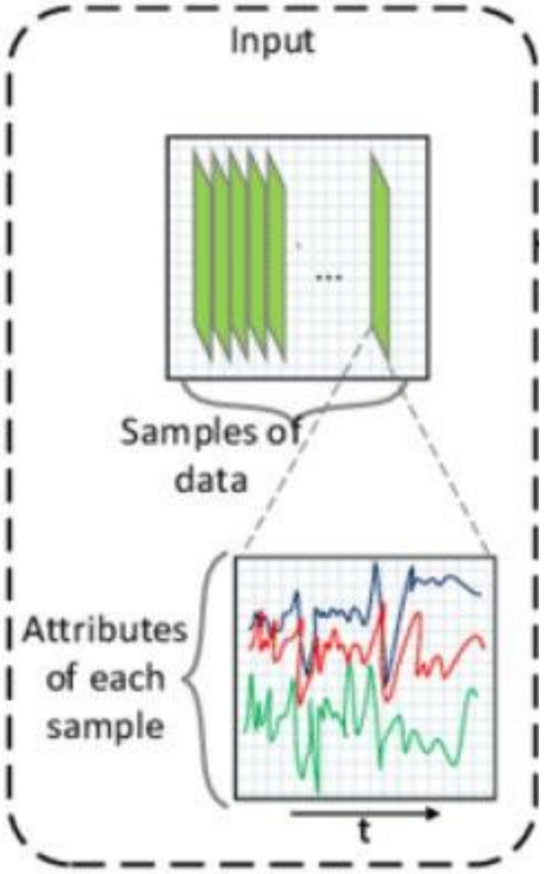


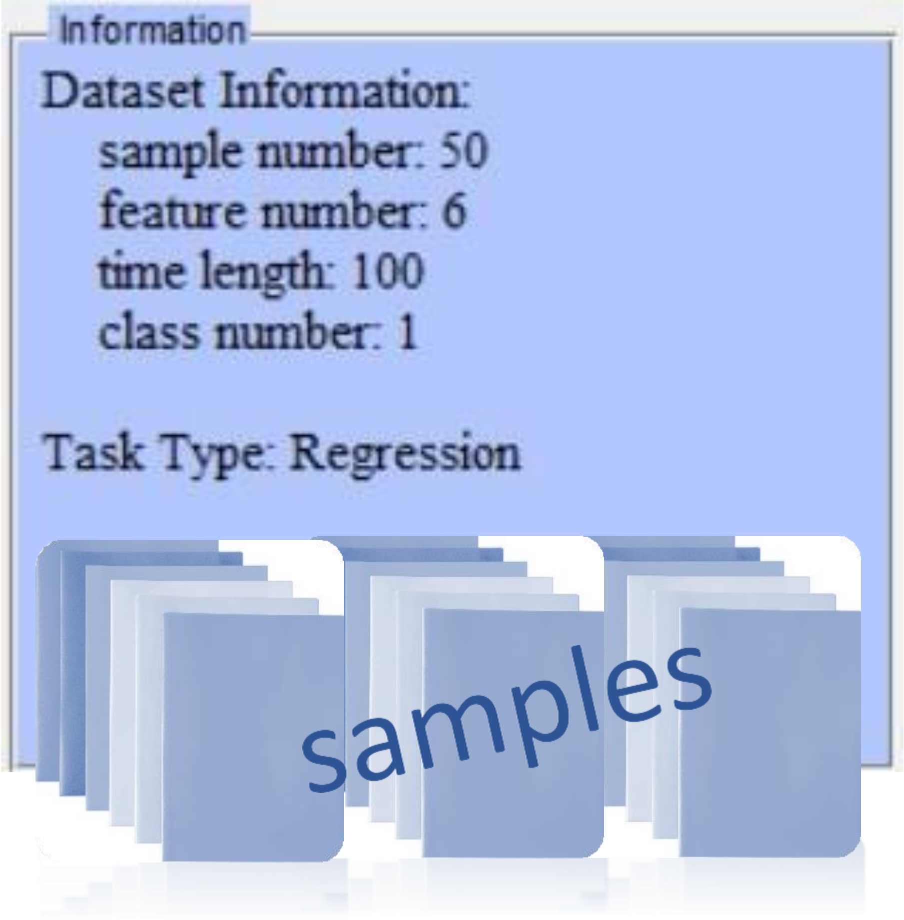
NeuCube

Regression Model

iabouhassan@tu-sofia.bg; nkasabov@aut.ac.nz; https://www.knowledgeengineering.ai/china

Regression Model



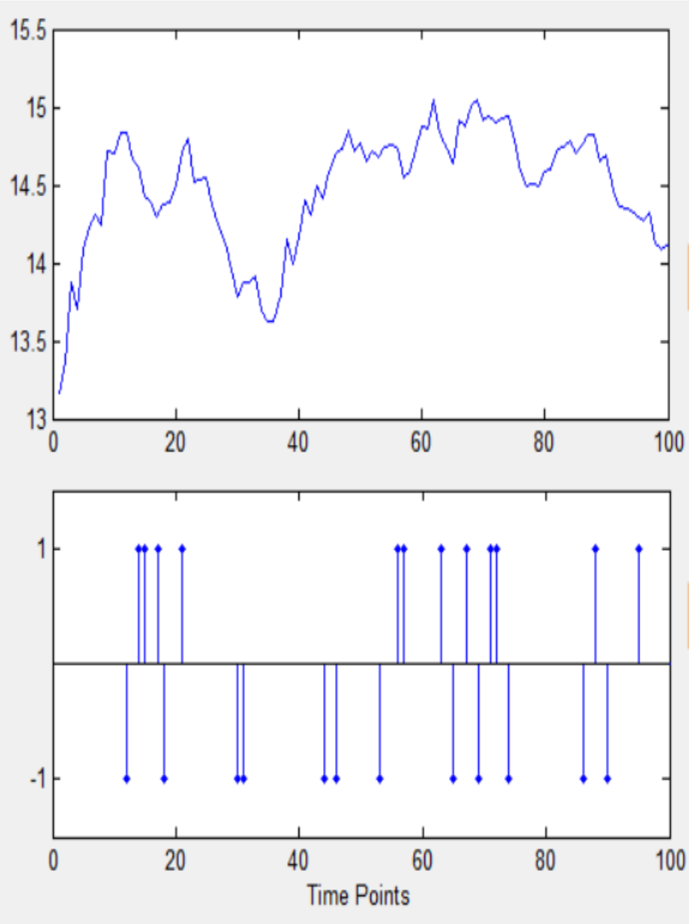


AAPL	GOOGL	INTC	MSFT	YHOO	QQQX
373.62	579.04	20.79	25.68	11.74	13.16
377.37	577.52	20.85	25.94	12.00	13.36
392.57	601.17	21.81	26.92	13.02	13.88
388.91	592.40	21.72	26.80	12.76	13.70
396.75	606.77	22.24	27.27	13.10	14.08
390.48	603.69	22.33	27.40	13.10	14.23
391.82	610.94	22.55	27.72	13.50	14.31
392.59	607.22	22.53	27.33	13.59	14.24
403.41	622.52	22.90	28.08	13.94	14.72
398.50	618.98	23.03	27.91	13.69	14.70
393.30	618.23	23.13	27.53	13.98	14.83
387.29	606.99	22.81	27.10	13.59	14.84
386.90	595.35	22.99	27.06	13.48	14.67
376.85	602.55	23.06	27.54	14.59	14.61
⋮					
358.02	538.26	22.48	26.63	14.91	14.37
353.75	534.01	22.45	26.54	14.86	14.39
354.00	527.28	22.85	26.63	15.05	14.49
359.71	531.99	23.09	26.92	15.61	14.72

Regression Model

Data Encoding

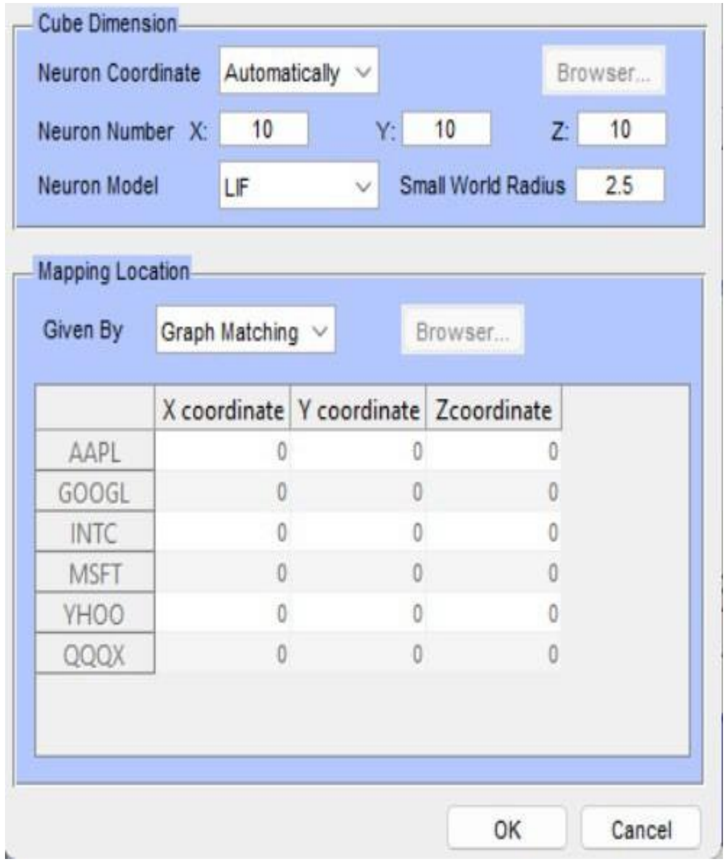
- **TR = 0.5** a threshold encoding method based on thresholding the difference between two consecutive values of same input variable over time.
- **Split ratio = 50|50** Training/incremental learning and testing.
- The real input data is transformed from continuous values to discrete sequences of spikes.
- Generating positive spikes that encode increased values at a next time point; and negative spikes for decreased values.

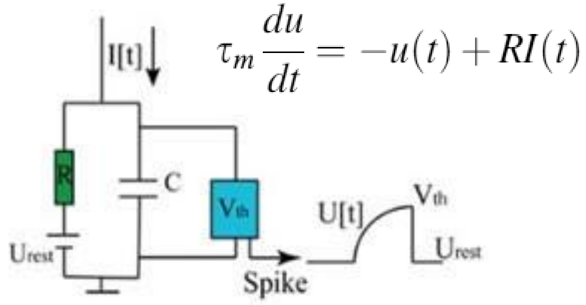


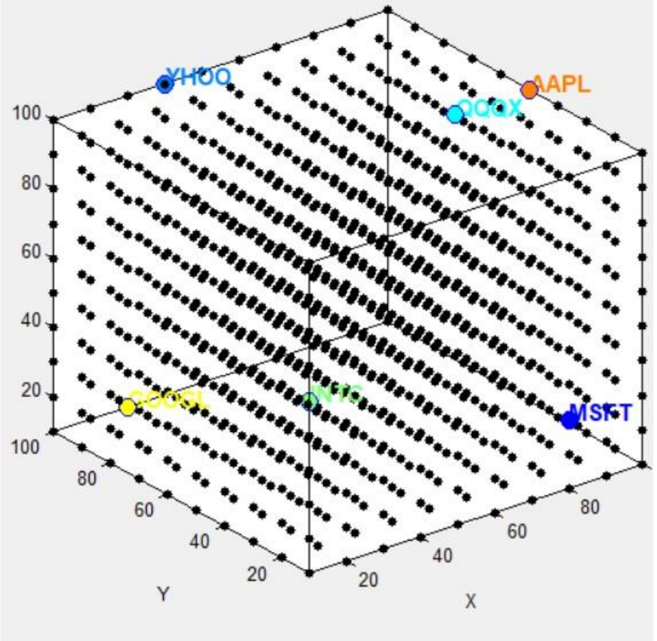
Regression Model

- **1000** neurons in the 3D-cube.
- **SWR = 2.5** is the small world connectivity to initialize the connections in the SNN reservoir so that closer neurons are more likely to be connected.
- Leaky Integrate and Fire model of spiking neuron: a simple RC circuit, with current (I), membrane potential (u), and membrane time constant
- A graph matching algorithm is adopted to assign the coordinates of the neurons since no spatial ordering for financial datasets.

Cube Initialization



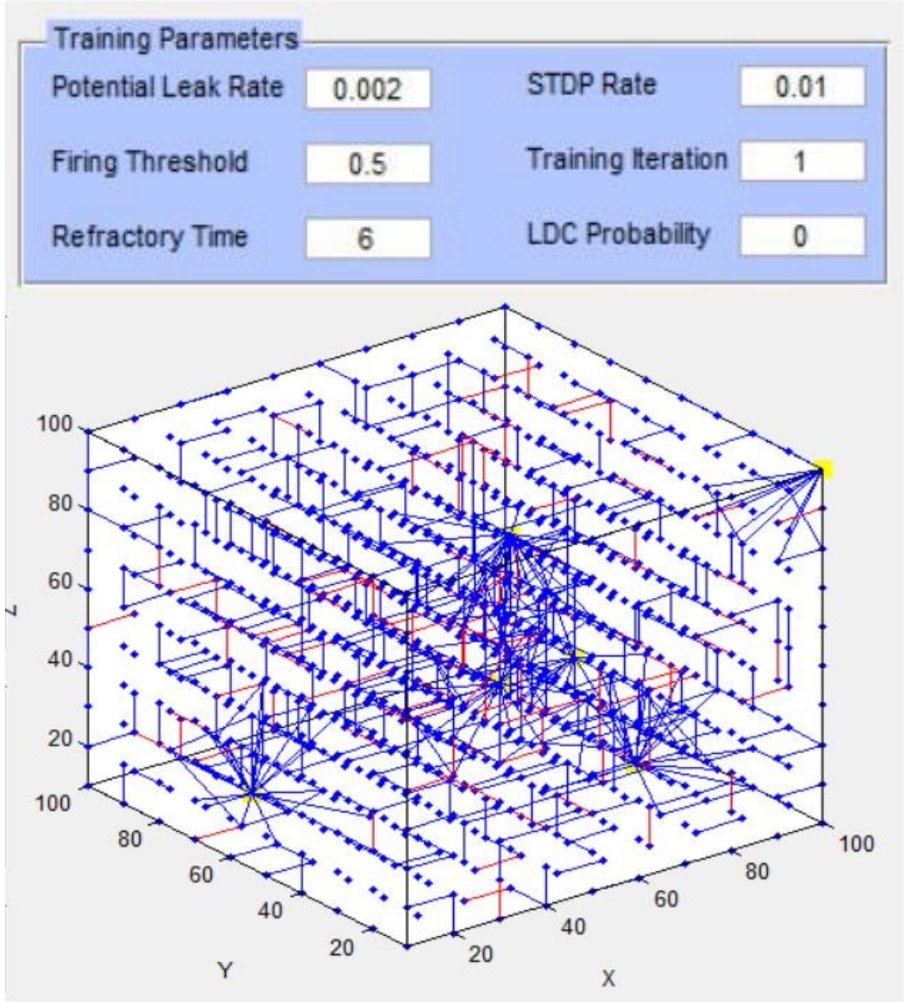




Regression Model

Unsupervised Learning

- **Potential leak rate = 0.002** is the leak in the membrane potential of a neuron between spikes, when the neuron does not fire.
- **Firing threshold = 0.5** is the threshold membrane potential beyond which the LIF neuron fires a spike.
- **Refractory time = 6** is the absolute time in units to reset membrane potential after a neuron emits spike and during which it will not fire.
- **Spike-timing-dependent synaptic plasticity (STDP) learning rate = 0.01** defines how much the weights of connected neurons should change when the neurons spike one after another within a small time window.
- **Training iteration = 1** is the number of times the NeuCube is trained.
- **LDC probability** is the probability of creating long distance connection.



Regression Model

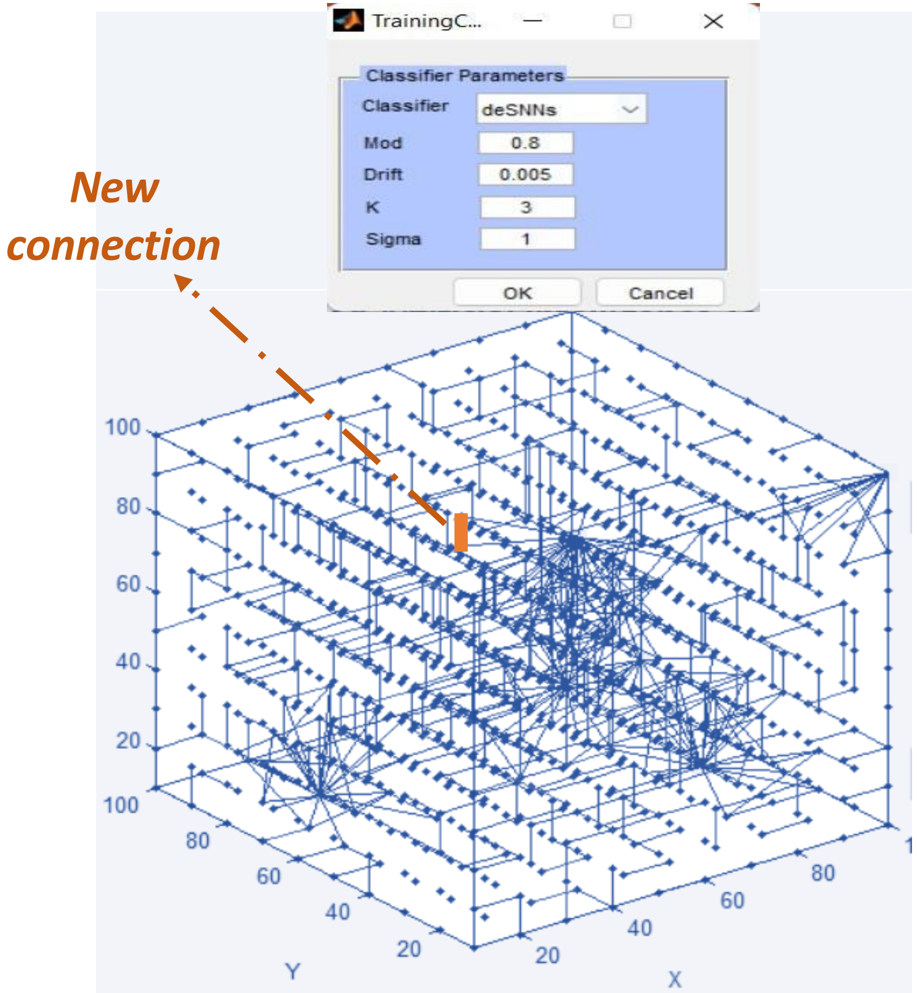
Supervised Learning

- The output regression module is trained using the dynamic evolving Spiking Neural Networks (**deSNN**), a computationally efficient model that:
  - gives a high priority to the first spike arriving at the output neuron.
  - **Rank Order** learning rule for weight initialization based on the first spikes;

$$w_{j,i} = \alpha \,.\, mod^{order(j,i)}$$

- Further learning and adjusting the connections from input spikes at a synapse following the first spike through a drift.

$$\Delta w_{j,i}(t) = e_j(t) \,.\, D$$



Regression Model

Output Layer visualization

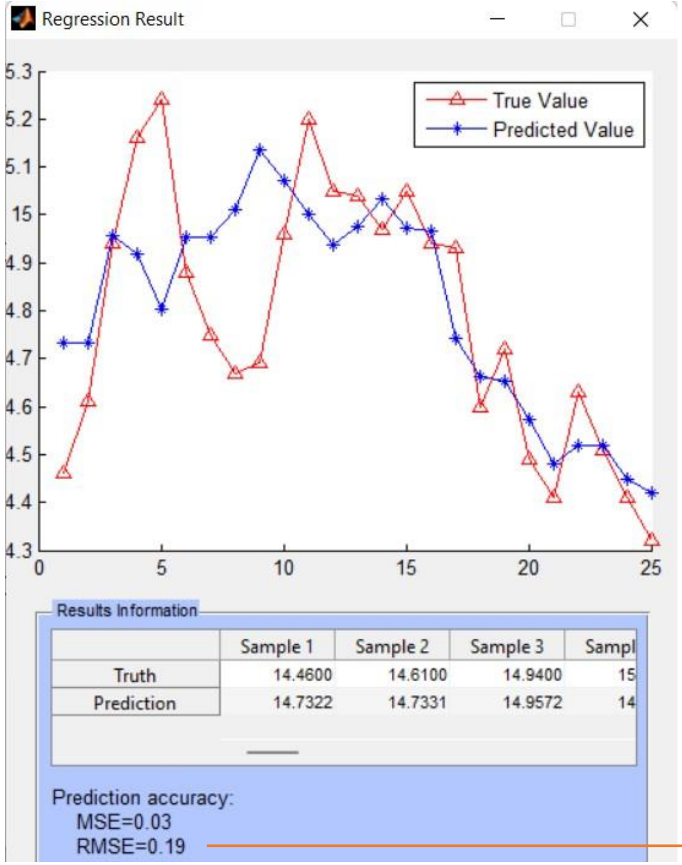
- **True label**: displays the real input data of each sample by using a different color for each association. The samples are ordered by their number from bottom to top.
- **Predicted label**: displays the predicted data of each sample from the test/validation data set in the same way as for the true labels.
- **deSNN potential**: displays the membrane potential of the output neuron per sample. A brighter neuron signifies higher potential.

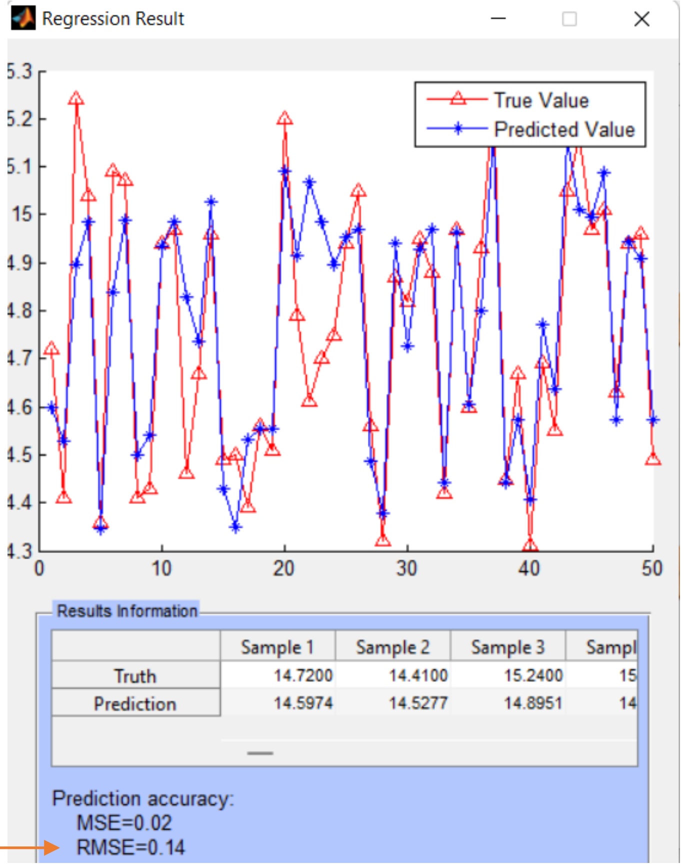
Regression Model

Verifying Regressor

Optimization

- This stage validates the model's accuracy.
- The graph depicts the difference between the real and predicted values of the validation samples.
- Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) are helpful measurements of model performance and forecast accuracy.
- Optimization is used to minimize error and improve forecast accuracy.

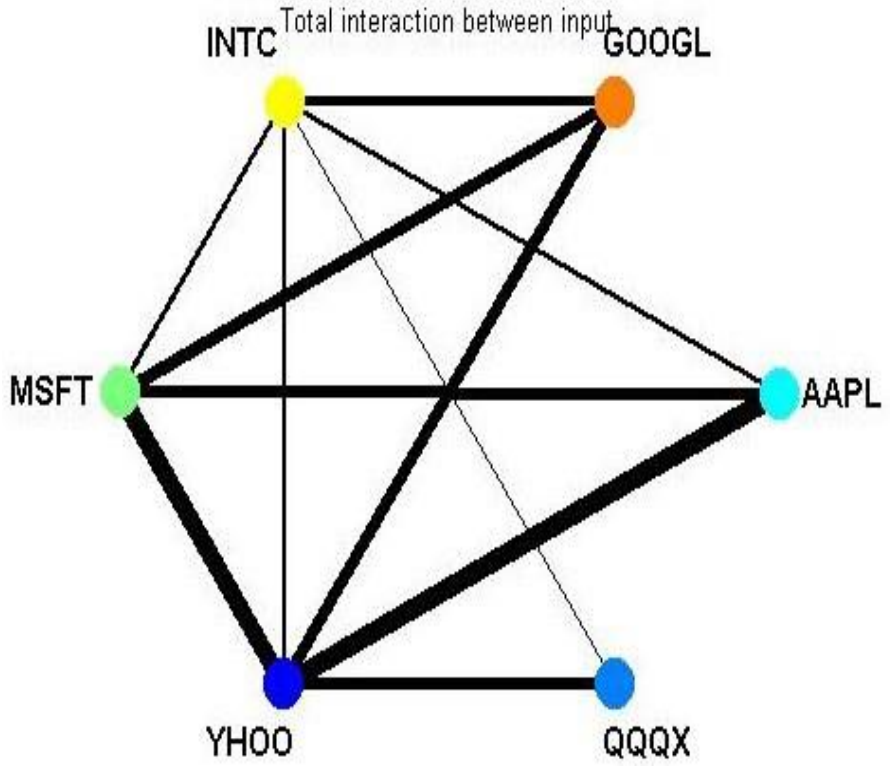


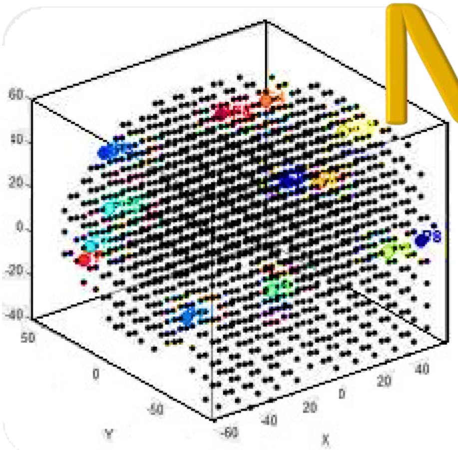


Regression Model

Feature Interconnection

Total interaction between the input neuron clusters based on the connection weight analysis. Thicker lines indicate more interaction.

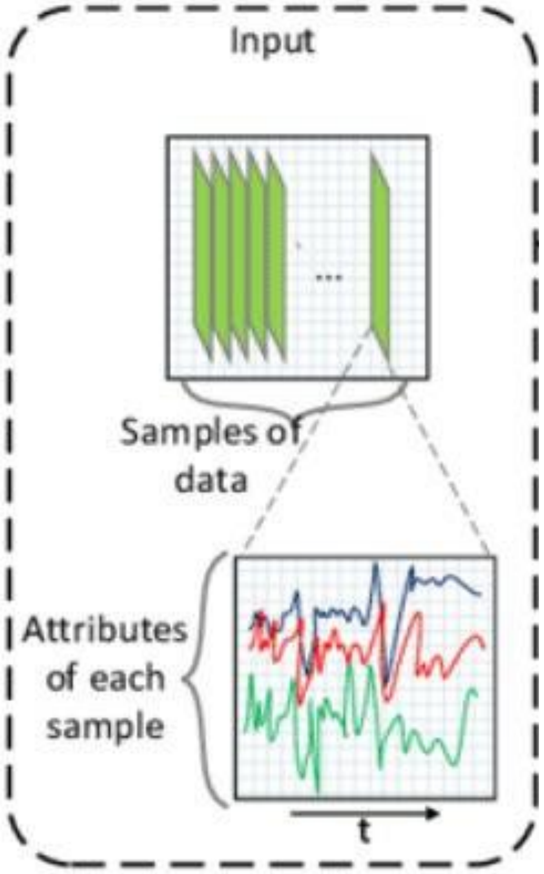


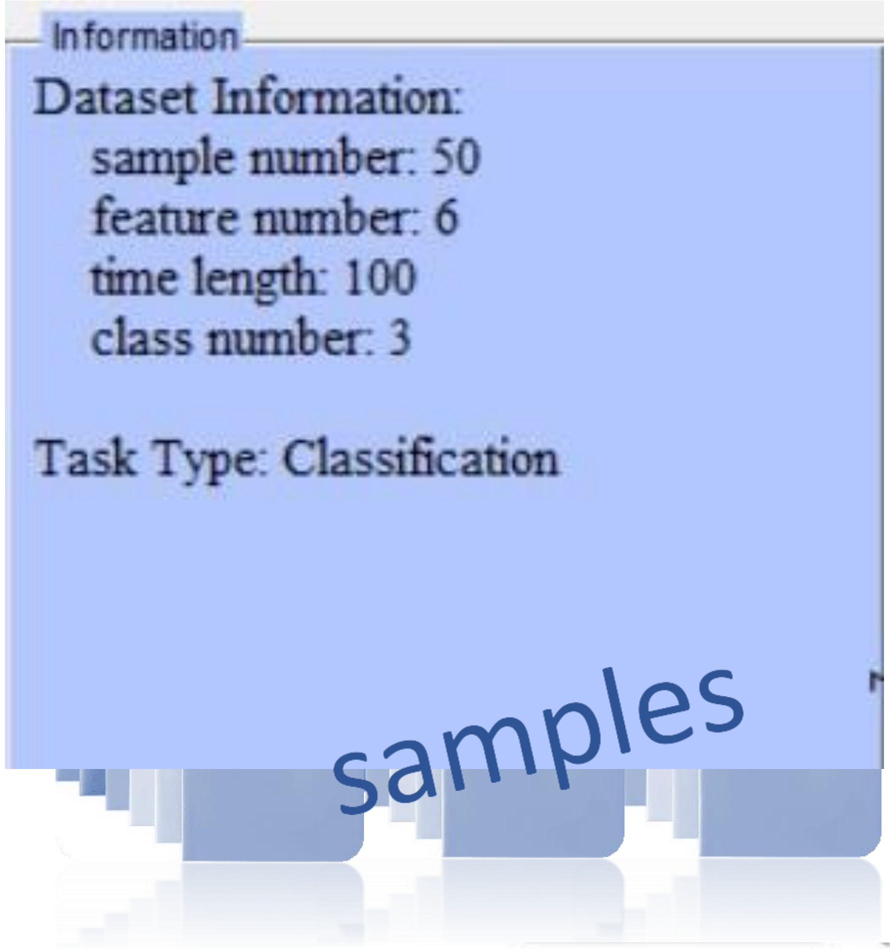


NeuCube

Classification Model

Classification Model



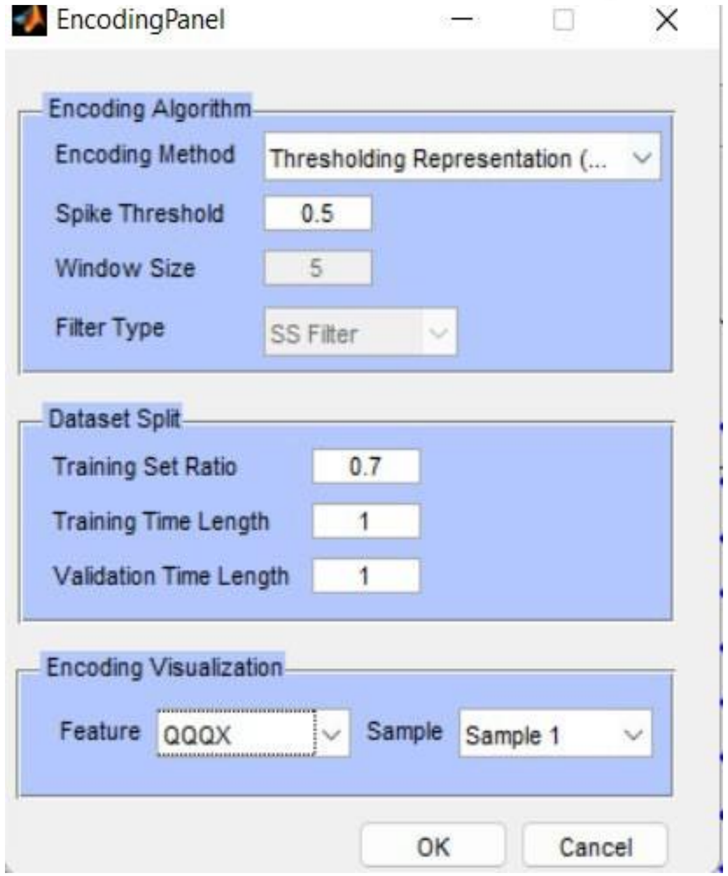


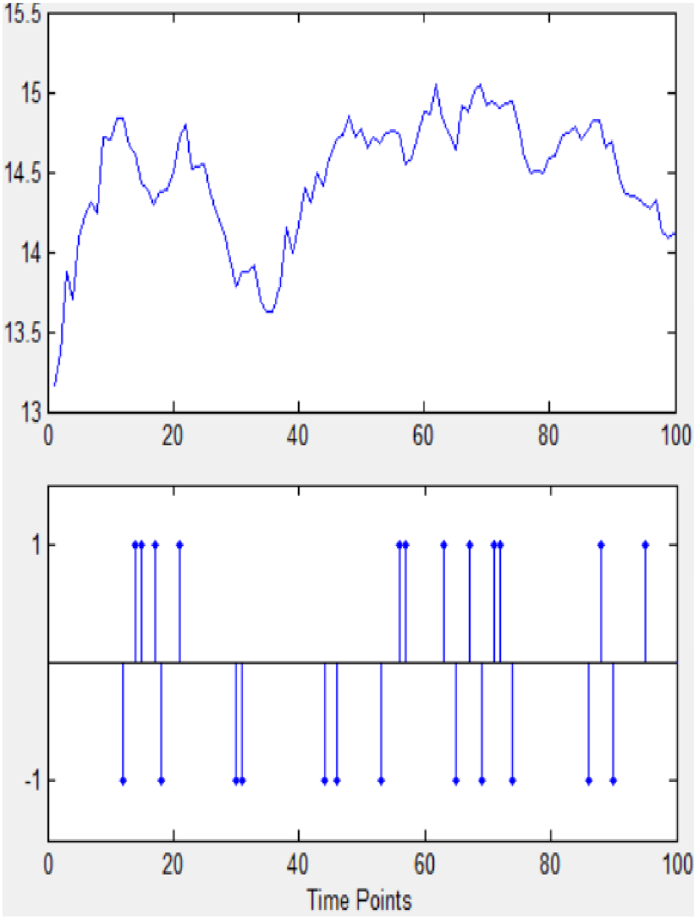
AAPL	GOOGL	INTC	MSFT	YHOO	QQQX	QQQX (class)
373.62	579.04	20.79	25.68	11.74	13.16	2
377.37	577.52	20.85	25.94	12.00	13.36	3
392.57	601.17	21.81	26.92	13.02	13.88	3
388.91	592.40	21.72	26.80	12.76	13.70	3
396.75	606.77	22.24	27.27	13.10	14.08	3
390.48	603.69	22.33	27.40	13.10	14.23	3
391.82	610.94	22.55	27.72	13.50	14.31	3
392.59	607.22	22.53	27.33	13.59	14.24	3
403.41	622.52	22.90	28.08	13.94	14.72	3
398.50	618.98	23.03	27.91	13.69	14.70	3
393.30	618.23	23.13	27.53	13.98	14.83	2
387.29	606.99	22.81	27.10	13.59	14.84	2
386.90	595.35	22.99	27.06	13.48	14.67	2
376.85	602.55	23.06	27.54	14.59	14.61	2
⋮						
358.02	538.26	22.48	26.63	14.91	14.37	3
353.75	534.01	22.45	26.54	14.86	14.39	3
354.00	527.28	22.85	26.63	15.05	14.49	3
359.71	531.99	23.09	26.92	15.61	14.72	3

Classification Model

Data Encoding

- **TR = 0.5** a threshold encoding method based on thresholding the difference between two consecutive values of same input variable over time.
- **Split ratio = 70|30** Training/incremental learning and testing.
- The real input data is transformed from continuous values to discrete sequences of spikes.
- Generating positive spikes that encode increased values at a next time point; and negative spikes for decreased values.

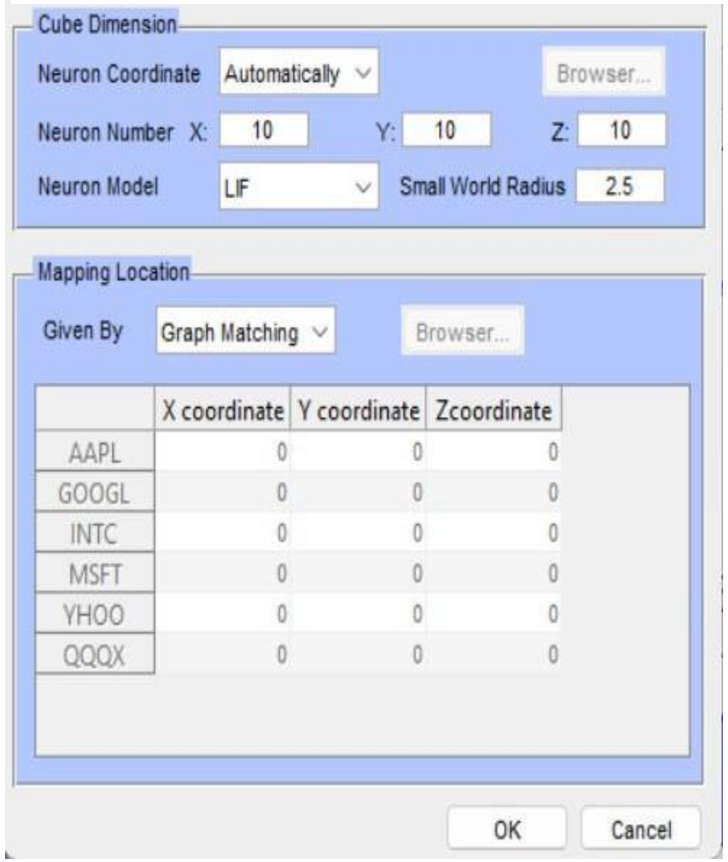


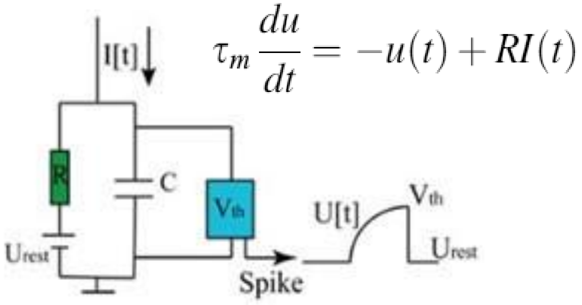


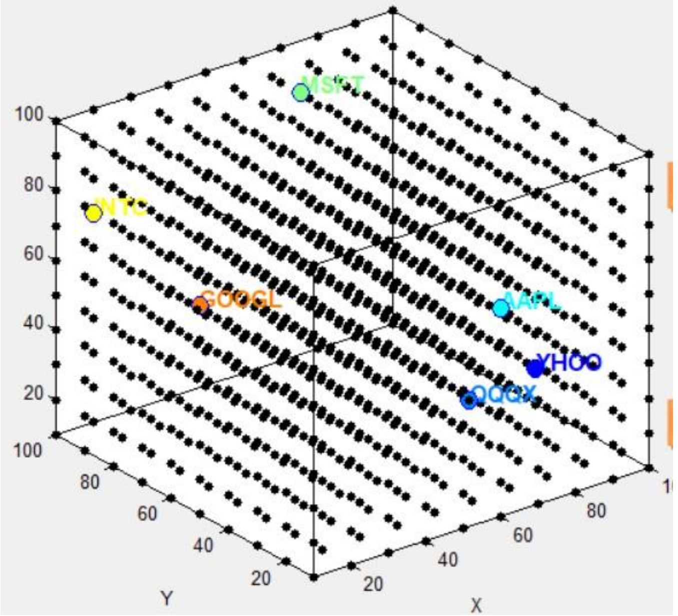
Classification Model

- **1000** neurons in the 3D-cube.
- **SWR = 2.5** is the small world connectivity to initialize the connections in the SNN reservoir so that closer neurons are more likely to be connected.
- Leaky Integrate and Fire model of spiking neuron: a simple RC circuit, with current (I), membrane potential (u), and membrane time constant
- A graph matching algorithm is adopted to assign the coordinates of the neurons since no spatial ordering for financial datasets.

Cube Initialization



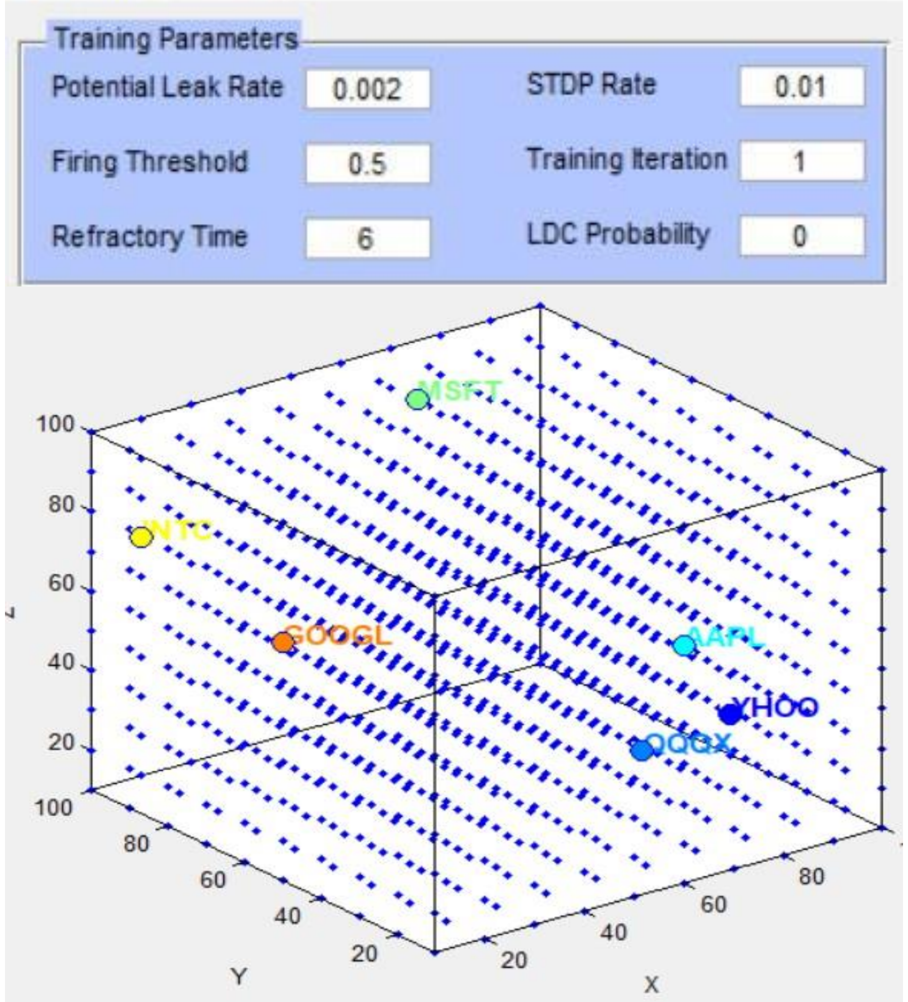




Classification Model

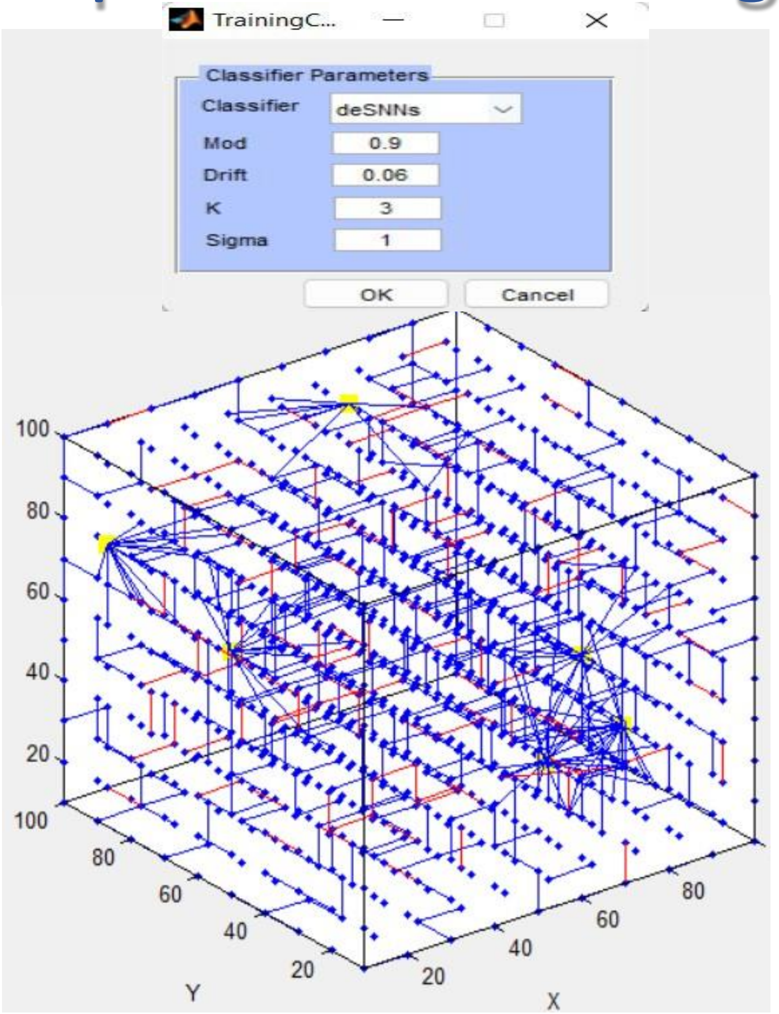
Unsupervised Learning

- **Potential leak rate = 0.002** is the leak in the membrane potential of a neuron between spikes, when the neuron does not fire.
- **Firing threshold = 0.5** is the threshold membrane potential beyond which the LIF neuron fires a spike.
- **Refractory time = 6** is the absolute time in units to reset membrane potential after a neuron emits spike and during which it will not fire.
- **Spike-timing-dependent synaptic plasticity (STDP) learning rate = 0.01** defines how much the weights of connected neurons should change when the neurons spike one after another within a small time window.
- **Training iteration = 1** is the number of times the NeuCube is trained.
- **LDC probability** is the probability of creating long distance connection.



Classification Model

Supervised Learning



- The output regression module is trained using the dynamic evolving Spiking Neural Networks (**deSNN**), a computationally efficient model that:
  - gives a high priority to the first spike arriving at the output neuron.
  - **Rank Order** learning rule for weight initialization based on the first spikes;

$$w_{j,i} = \alpha \,.\, mod^{order(j,i)}$$

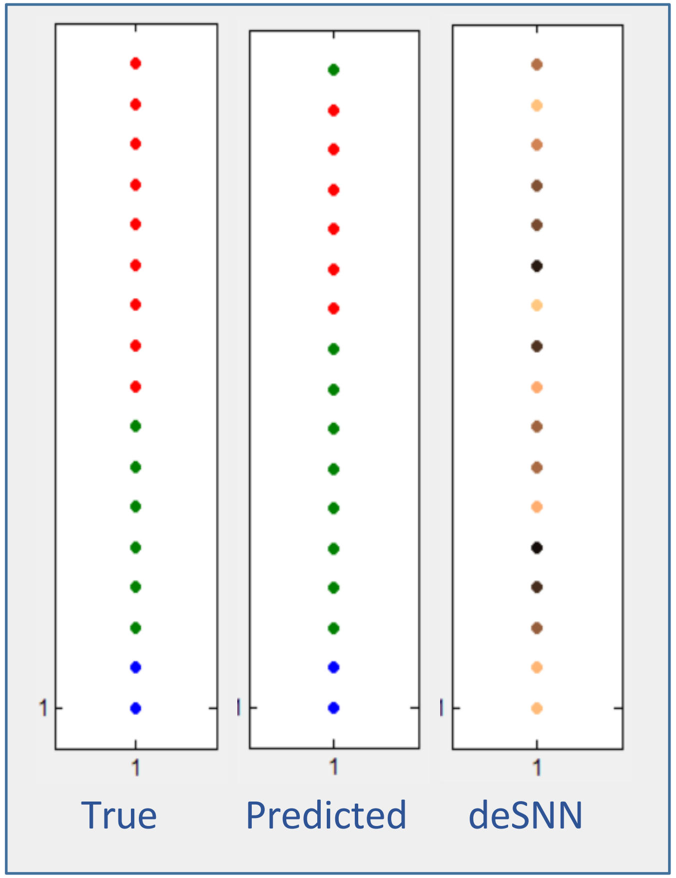
- Further learning and adjusting the connections from input spikes at a synapse following the first spike through a drift.

$$\Delta w_{j,i}(t) = e_j(t) \,.\, D$$

Classification Model

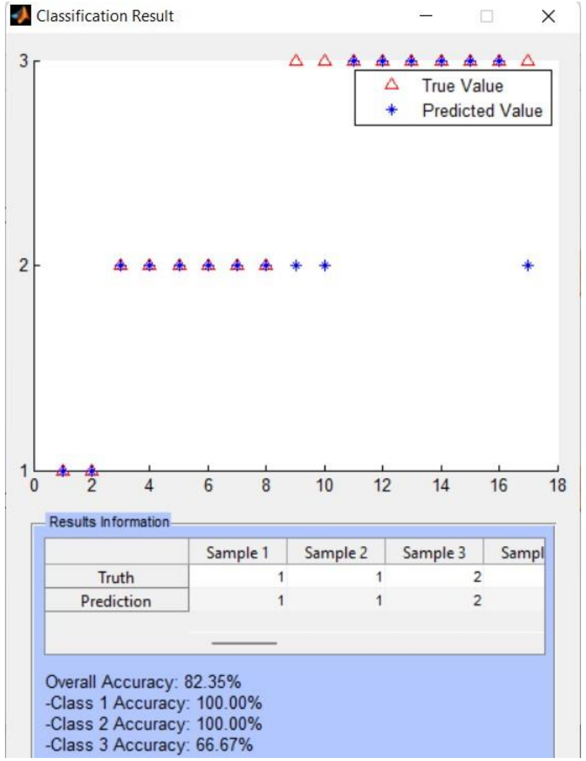
Output Layer visualization

- **True label**: displays the real input data of each sample by using a different color for each class. The samples are ordered by their number from bottom to top.
- **Predicted label**: displays the predicted data of each sample from the test/validation data set in the same way as for the true labels.
- **deSNN potential**: displays the membrane potential of the output neuron per sample. A brighter neuron signifies higher potential.



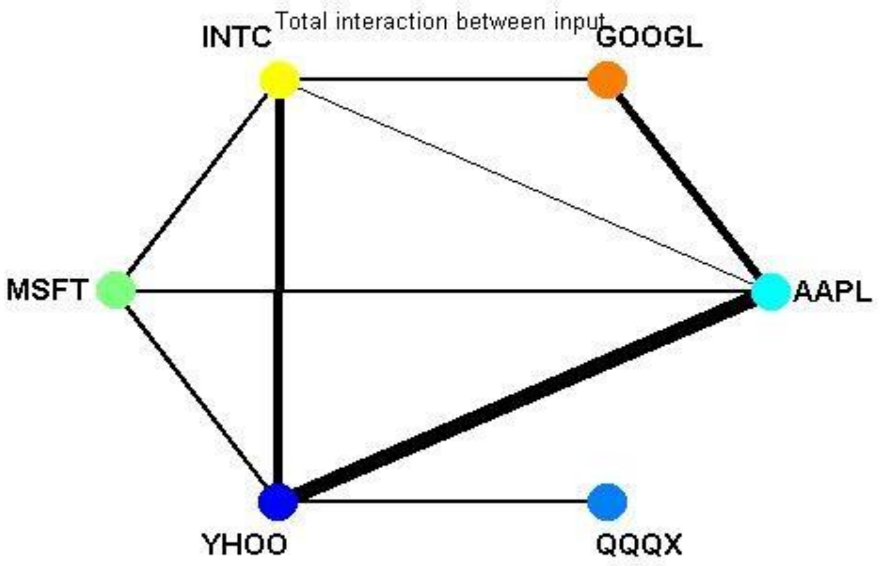
Classification Model

Verifying Classifier

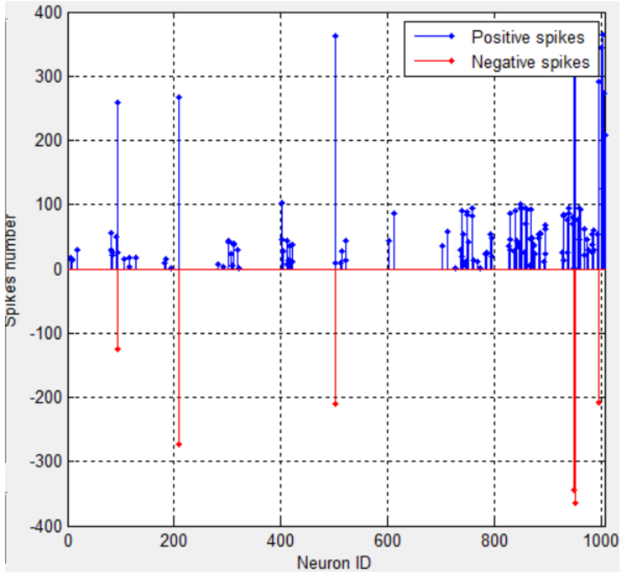


Feature Interconnection

Total interaction between the input neuron clusters based on the connection weight analysis. Thicker lines indicate more interaction.

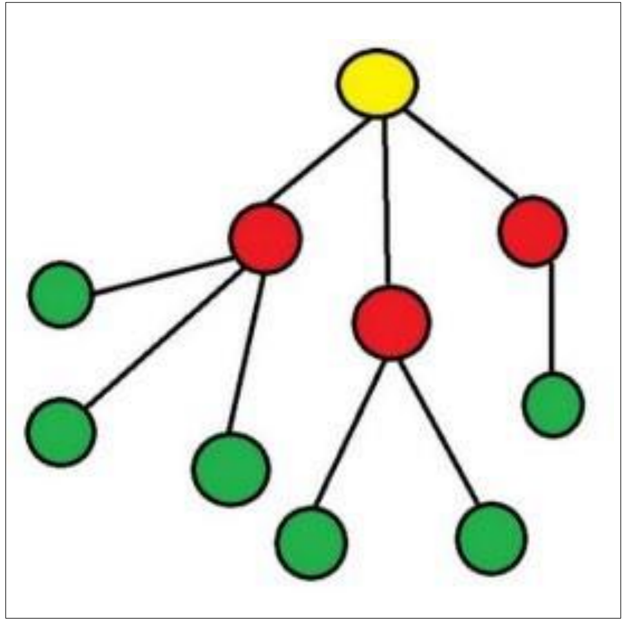


Analysis/visualization of NeuCube connectivity



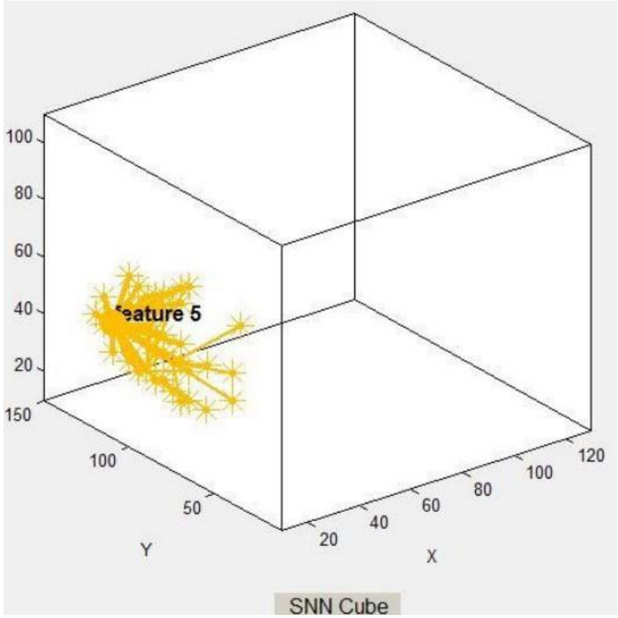
- ‘**Activation Level**’ shows the membrane potential or the spike activation level of the neurons.
- ‘**Spikes Emitted**’ shows a histogram of positive and negative spikes emitted by all neurons.
- ‘**Neuron Weight**’ visualizes the connection weights of all neurons connected to a specific neuron ID.
- ‘**Spike Raster**’ generates the raster plot of spike activity for a specific sample. It shows the response of the spiking neurons to changes of a neuronal parameter.
- ‘**Spike Activity Playback**’ allows to dynamically visualize the spike dynamics over time.

Analysis/visualization of NeuCube network



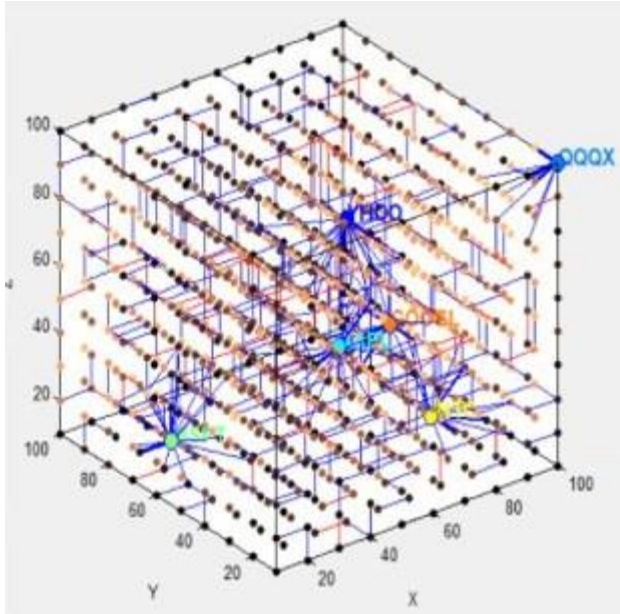
- **Information route analysis:**
  - analyzing the information propagation route of the spikes.
  - This analysis is based on the concept of a rooted tree structure. A rooted tree is a directed tree having a single root node (neuron). A neuron's 'parent' is a neuron which is one step higher in hierarchy and lying on the same branch. Different methods of analysis are available:
    - ***Max spike gradient***: shows a tree rooted by input neuron, where a child neuron is connected to its parent if it receives spike from them.
    - ***Spreading level***: shows a tree from the input neuron to its neighborhood which reflects the spreading of the spikes. The 'level number' parameter defines the neighborhood of spread. For example, setting this parameter to 2 will show the spike distribution from the input neuron to two layers of neighboring connected neurons.

Analysis/visualization of NeuCube network



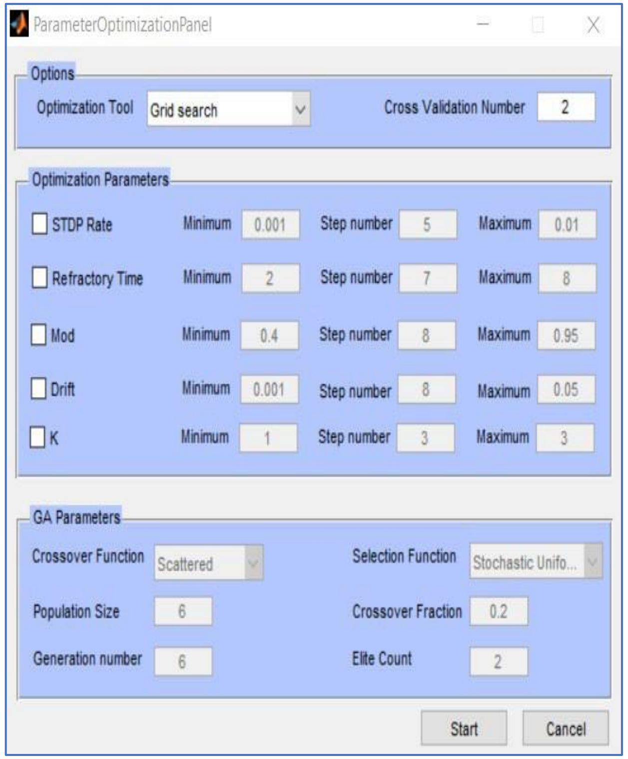
- **Information amount:**
  - shows a tree rooted by the input neuron where a child neuron is chosen to be part of the tree only if it receives a minimum percentage of spikes from its parent neuron.
  - The percentage is specified as decimal value (0.1 means a minimum of 10% spikes).

Output Layer visualization



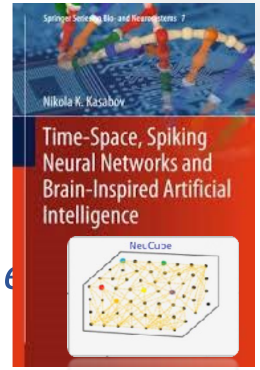
- **Connection strength**: visualize the **strength of connections** between the neurons for every output neuron (sample). By clicking on one of the neurons in the output layer, it shows the connection strength of the neurons in the cube for that particular output neuron. Brighter neurons are more strongly connected than darker neurons.
- **First spike order**: visualize the **spiking order** of the neurons for each output neuron (sample). By clicking on one of the neurons in the output layer, it shows the firing order of the neurons in the cube for that particular output neuron. Brighter neurons fire earlier than darker neurons.

Optimization procedure



- **Cross validation**: a function that is wrapped around the unsupervised and supervised learning. At every fold the cube is initialized, trained unsupervised, and trained supervised with different combinations of data. The fold number parameter defines the number of iterations of training and validation cycles.
- **Parameter optimization:** can be used to search for an optimal set of hyper-parameters that minimizes the test error of the model. The computational time for parameter optimization depends on the number of parameters to be optimized and the size of the NeuCube model.
  - **Exhaustive grid search**: using a grid-based combination of parameters.
  - **Genetic algorithm (GA):** This is a nature inspired algorithm that employs the workings of genetic recombination in beings as they happen in nature.

References:



- Kasabov N. (2019): *Time-Space, Spiking Neural Networks and Brain-Inspired Artificial Intelligence*; Springer.
- Tu, E., N. Kasabov, J. Yang, *Mapping Temporal Variables into the NeuCube Spiking Neural Network Architecture for Improved Pattern Recognition and Predictive Modelling*, IEEE Trans. on Neural Networks and Learning Systems, 28 (6), 1305-1317,, 2017 DOI: 10.1109/TNNLS.2016.2536742, 2017.
- Kasabov N. (2014): *NeuCube: A Spiking Neural Network Architecture for Mapping, Learning and Understanding of Spatio-Temporal Brain Data; Elsevier, Neural Networks*, Vol. 52, pp. 62-76, doi:10.1016/j.neunet.2014. 01.006.
- Kasabov N., Dhoble K., Nuntalid N., Indiveri G. (2013): *Dynamic Evolving Spiking Neural Networks for Online Spatio-and spectro-temporal Pattern Recognition*; Elsevier, Neural Networks, Vol. 41, pp. 188-201.
- NeuCube Development environnement: https://kedri.aut.ac.nz/neucube
- Join the Club: https://www.knowledgeengineering.ai/efunn-denfis-neucube-club

iabouhassan@tu-sofia.bg; nkasabov@aut.ac.nz; https://www.knowledgeengineering.ai/china