

EQUIPMENT ANOMALY DETECTION METHOD, COMPUTER READABLE STORAGE
MEDIUM, CHIP, AND DEVICE

Technical Field

[0001] The present disclosure relates to the field of computer technology, and more particularly, to neural network based equipment anomaly detection method, and device.

Background Art

[0002] Industrial and residential machines typically use motors to drive operational mechanisms and functions. Motors and mechanical parts may age and wear. When deformation of such components aggregate to a certain level may cause a machine to fail suddenly and sometimes unexpected accidents and loss.

Technical Problem

[0003] End users and consumers of industrial and residential machines have difficulties to obtain sufficient data to recognize anomalies, such as malfunctioning, faults, or failure modes of the machines. Insufficient data may render such anomalies unpredictable.

Technical Solution

[0004] A first aspect of the present disclosure is to propose an equipment anomaly detection device. The equipment anomaly detection device includes an cascade of neural networks and a detection module. The cascade of neural networks is configured for receiving and learning N nominal sensor event streams from N event stream input connections in a learning state to form a learned state of the cascade of neural networks. The cascade of neural networks in the learned state receives N monitored sensor event streams from the N event stream input connections as input of the cascade of neural networks in an operation state, and outputs N monitored reconstructed event streams as output of the cascade of neural networks based on the N monitored sensor event streams. The N monitored reconstructed event streams represent a reconstructed version of the N monitored sensor event streams. The detection module is configured for calculating difference between the N monitored sensor event streams and the N monitored reconstructed event streams. The detection module sends a detection trigger when the difference between N monitored sensor event streams and N monitored reconstructed event streams exceeds a threshold difference value.

[0005] A second aspect of the present disclosure is to propose an equipment anomaly detection method. The method includes using an cascade of neural networks to receive and learn N nominal sensor event streams from N event stream input connections in a learning state to form a learned state of the cascade of neural networks, wherein the cascade of neural networks in the learned state receives N monitored sensor event streams from the N event stream input connections as input of the cascade of neural networks in an operation state, and outputs N monitored reconstructed event

streams as output of the cascade of neural networks based on the N monitored sensor event streams, and the N monitored reconstructed event streams represent a reconstructed version of the N monitored sensor event streams; and calculating difference between the N monitored sensor event streams and the N monitored reconstructed event streams, wherein the detection module sends a detection trigger when the difference between N monitored sensor event streams and N monitored reconstructed event streams exceeds a threshold difference value.

[0006] The disclosed method may be implemented in a chip. The chip may include a processor, configured to call and run a computer program stored in a memory, to cause a device in which the chip is installed to execute the disclosed method.

[0007] The disclosed method may be programmed as computer executable instructions stored in non-transitory computer readable medium. The non-transitory computer readable medium, when loaded to a computer, directs a processor of the computer to execute the disclosed method.

[0008] The non-transitory computer readable medium may comprise at least one from a group consisting of: a hard disk, a CD-ROM, an optical storage device, a magnetic storage device, a Read Only Memory, a Programmable Read Only Memory, an Erasable Programmable Read Only Memory, EPROM, an Electrically Erasable Programmable Read Only Memory and a flash memory.

[0009] The disclosed method may be programmed as computer program product, that causes a computer to execute the disclosed method.

[0010] The disclosed method may be programmed as computer program, that causes a computer to execute the disclosed method.

Advantageous Effects

[0011] The equipment anomaly detection system of the disclosure uses unsupervised or self-supervised neural networks to learn nominal conditions of a target system, such as a device or a machine. The trained neural networks can reproduce sensory signals of the target system as a neural-network-reconstructed version of the sensory signals in the nominal conditions of a target system. The equipment anomaly detection system may analyze and predict operation conditions of the target system exceed a certain level based on the neural-network-reconstructed version. For example, when signal difference between the sensory signals and the neural-network-reconstructed version exceed a certain level, the equipment anomaly detection system may issue an alert signal to reflect abnormal operation conditions of the target system, which may be a period ahead of an actual onset of system failure of the target system. Thus, personnel may have more time to address anomalies of the target system and prevent substantial loss.

Description of Drawings

[0012] In order to more clearly illustrate the embodiments of the present disclosure or related art, the following figures will be described in the embodiments are briefly introduced. It is obvious that the drawings are merely some embodiments of the present disclosure, a person having ordinary skill in this field can obtain other figures according to these figures without paying the premise.

[0013] FIG. 1 is a schematic diagram showing an equipment anomaly detection system according to an embodiment of the present disclosure.

[0014] FIG. 2 is a schematic diagram showing a signal conversion module according to an embodiment of the present disclosure.

[0015] FIG. 3 is a schematic diagram showing a signal conversion unit according to an embodiment of the present disclosure.

[0016] FIG. 3 is a schematic diagram showing signal waveforms during signal processing in the signal conversion unit according to an embodiment of the present disclosure.

[0017] FIG. 5 is a schematic diagram showing connection from a input event stream to a embedding network and a detection module in the equipment anomaly detection system according to an embodiment of the present disclosure.

[0018] FIG. 6 is a schematic diagram showing an embedding network realizing a winner-takes-all (WTA) algorithm according to an embodiment of the present disclosure.

[0019] FIG. 7 is a schematic diagram showing an embedding network realizing a soft-winner-takes-all (sWTA) algorithm according to another embodiment of the present disclosure.

[0020] FIG. 8 is a schematic diagram showing a decoding network according to an embodiment of the present disclosure.

[0021] FIG. 9 is a schematic diagram showing a detection module according to an embodiment of the present disclosure.

[0022] FIG. 10 is a schematic diagram showing operations of the equipment anomaly detection system according to an embodiment of the present disclosure.

[0023] FIG. 11 is a schematic diagram showing the equipment anomaly detection system in a learning state.

[0024] FIG. 12 is a schematic diagram showing the equipment anomaly detection system in an operation state.

[0025] FIG. 13 is a schematic diagram showing a Tetzlaff learning rule and a Jonke learning rule.

[0026] FIG. 14 is a schematic diagram showing software modules of an equipment anomaly detection system according to an embodiment of the present disclosure.

DETAILED DESCRIPTION OF EMBODIMENTS

[0027] Embodiments of the disclosure are described in detail with reference to the accompanying drawings as follows. It can be appreciated that the embodiments described herein are merely a portion not all of embodiments of the present invention. Based on the embodiments of the present invention, all other embodiments obtained by one of ordinary skill in the art without performing inventive work are within the claims the present invention.

[0028] The disclosure presents an equipment anomaly detection method and device for detecting anomaly in temporal signals of a monitored target, such as system, a device, a facility or a plant.

[0029] The equipment anomaly detection device in a learning state automatically adapts and learns a nominal state of the monitored target from nominal sensor signals and identifies deviations away from the nominal state as anomalies. The disclosed device uses online learning in an encoder-decoder network, to learn nominal input sensor signals in a learning state, and learns to reproduce network reconstructed version of the nominal input sensor signals. The device in an operation state measures difference between current input sensor signals and network reconstructed version of the current input sensor signals to detect and signal an anomaly associated with the monitored target and the current input sensor signals.

[0030] When the network reconstructed version matches the current input sensor signals, the equipment anomaly detection device determines the monitored target to be in the nominal state. When the difference between the current input sensor signals and network reconstructed version is small, such as not exceeding a threshold, no anomaly signal is produced by the equipment anomaly detection device.

[0031] When the monitored target moves into an anomalous state, the network cannot reproduce the input sensor signals accurately. The equipment anomaly detection device may determine a trend of the monitored target where the difference between the current input sensor signals and network reconstructed version becomes greater, such as exceeding the threshold. Once this difference crosses a threshold magnitude, the equipment anomaly detection device raises and sends an anomaly signal.

[0032] System overview:

[0033] With reference to FIG. 1, an embodiment of the disclosure provides an equipment anomaly detection system 100. A monitored target system 901 may include one or more entities, such as one or more industrial plants, motors, devices, facilities, or systems. The monitored target system 901 emits vibration signals 902 indicative of the monitored target's condition.

[0034] A sensor system may include sensors 111-1 to 111-N for detecting the monitored target system 901. N is a positive integer, which may be two or more. For example, the sensors in the sensor system may be attached or fixed to the monitored target system 901. Each of the sensor

111-1 to 111-N detects the monitored target and generates a stream of sensor signals representing vibration signals of the monitored target. The vibration signals 902 are detected by the sensor system 110 as sensor signals 112, and converted to trains of asynchronous input event streams 190 by a signal conversion module 120. The sensor signals 112 may be transmitted through wirelines or wireless connections. An embedding network 130 generates and outputs embedded encoding streams 139 to a decoding network 140 based on the input event streams 190. The embedding network 130 and the decoding network 140 form a cascade of neural networks and learn to reproduce reconstructed event streams 149 as a network reconstructed version of the input event streams 190. The detection module 150 compares the reproduced event streams 149 against the input event streams 190, and computes difference between the reproduced event streams 149 and the input event streams 190. The difference below a threshold represents no anomaly is detected, and the detection module 150 accordingly does not raise a trigger signal 180. If the difference is above a threshold, which represents anomaly is detected, then the detection module 150 accordingly raises a trigger signal 180.

[0035] An external system 200 may send signal 210 to the equipment anomaly detection system 100, to reset, configure and control the monitored target system 901 and the system 100. The external system 200 receives the trigger signal 180 and information about detected condition 170 associated with the anomaly detected. The trigger signal 180 and detected condition 170 may be transmitted through wirelines or wireless connections.

[0036] A learning configuration and control module 160 switches the system 100 between a learning state and a learned state.

[0037] An equipment anomaly detection device 101 includes the embedding network 130, the decoding network 140, the detection module 150, and the learning configuration and control module 160. In an embodiment, the device 101 may be packaged as an integrated circuit (IC), such as a system on a chip (SOC). In an alternative embodiment, the equipment anomaly detection device 101 may include the signal conversion module 120.

[0038] In an embodiment, the embedding network 130, the decoding network 140, the detection module 150, and the learning configuration and control module 160 may be implemented as computer executable instructions which when executed by a processor, performs the functions in the disclosure.

[0039] Signal conversion module function:

[0040] With cross reference to FIG. 1 and FIG. 2, the sensors 111-1 to 111-N send multiple sensor signals 112-1, 112-2, ... and 112-N to signal conversion units 122-1, 122-2, ... and 122-N in the signal conversion module 120. Each of the input sensor signals 112-1, 112-2, ... and 112-N is

converted by one of signal conversion units 122-1, 122-2, ... and 122-N into one of sensor event streams 190-1, 190-2, ... and 190-N. Specifically, the input sensor signals 112 include sensor signals 112-1, 112-2, ... and 112-N. The input event streams 190 include sensor event streams 190-1, 190-2, ... and 190-N. The signal conversion unit 122-1 generates the sensor event stream 190-1 from input sensor signal 112-1. Similarly, the signal conversion unit 122-2 generates the sensor event stream 190-2 from input sensor signal 112-2. The signal conversion unit 122-N generates the sensor event stream 190-N from input sensor signal 112-N.

[0041] Each of the signal conversion unit 122-1 to 122-N in the signal conversion module 120 performs the following functions, and an example of an input signal passing through a signal conversion unit is shown in the following.

[0042] With cross-reference to FIG. 3, FIG. 4 and FIG. 5, a sensor in the sensor system 110 sends an input signal 1220 to a signal conversion unit 122. The signal conversion unit 122 may be an embodiment of each of the signal conversion units 122-1, 122-2, ... and 122-N. The input signal 1220 represents one of the sensor signals 112-1, 112-2, ... and 112-N, and event stream 1225 represents one of sensor event streams 190-1, 190-2, ... and 190-N.

[0043] A bandpass filter 1221 with a chosen center frequency and bandwidth is used to select frequency components from the input signal 1220 and outputs the selected frequency components as filtered input signal 1221a. The input signal 1220 is sent to several bandpass filters, such that the input signal 1220 is separated into several frequency components.

[0044] The bandpass filtered signal 1221a is rectified and smoothened with a low-pass filter 1223. For example, the low-pass filter 1223 includes a rectifier 1223-1 and a signal smoothing unit 1223-2. The signal smoothing unit 1223-2 may be made from one or more capacitors. The rectifier 1223-1 generates a rectified signal 1223a from the filtered input signal 1221a. The signal smoothing unit 1223-2 generates a smooth rectified signal 1223b from the rectified signal 1223a.

[0045] The smooth rectified signal 1223b is encoded by pulse-density modulation (PDM) encoder 1224 using pulse-density modulation. For example, the PDM encoder 1224 may be a sigma-delta encoder. The PDM encoder 1224 generates the encoded smooth rectified signal 1223b as an event stream 1225. Each pulse in the event stream 1225 is transmitted as an event.

[0046] Network architecture and function:

[0047] The signal conversion module 120 emits a number of input event streams 190 which encode the input sensor signals as a number of event-coded signals in the selected frequency bands as described above.

[0048] The embedding network 130 learns to represent the high-dimensional input signals, such as N input event streams, in a lower-dimensional representation, such as M-dimensional

representation, where M is a positive integer, and $M < N$. The embedding network 130 facilitates particular pattern of input event streams 190 to be reliably mapped to a particular pattern of embedded encoding streams 139 at any given moment in time, which is based on an assumption that the event streams 190 being monitored exist in a small number of sub-states during the normal operation state of the monitored target. These sub-states represent a nominal condition of the monitored signals and the monitored target. The number of the possible normal sub-states can be chosen by changing the size of the M in the embedding network 130.

[0049] Embedding network:

[0050] The disclosure presents at least two embodiment of the embedding network 130.

[0051] In one embodiment, the embedding network 130 includes a number of embedding neurons to implement an online-learning algorithm. In one embodiment, the embedding network 130 is a spiking neural network including a plurality of spiking neurons to implement an online-learning algorithm. An example of the proposed spiking neural network is basically an auto-encoder composed of an input layer, a hidden layer and an output layer. The main differences with a classical auto-encoder are the following:

[0052] The embedding network 130 are trained through unsupervised learning. Training data are a small subset of the full data set, that supposedly do not contain any anomalies. The embodiment of the disclosure allows to learn an alternative representation of normal input signals by a encoder, that is the embedding network 130, and to accurately reconstruct normal input signals from such representation by a decoder, that is the decoding network 140. The basic principle behind is that the auto-encoder never learns to reconstruct abnormal signals. Thus, the reconstruction loss is small during normal conditions, and this loss is large in presence of abnormal signals since the auto encoder never learned to reconstruct abnormal signals. Such training procedure allows to detect anomalies without defining what an anomaly is supposed to look like.

[0053] The output layer is twice the dimension of the input layer, whereas an output layer and an input layer in a classic auto-encoder has equal dimensions. The reason is that the reconstruction loss is the absolute difference between inputs and outputs. A single spiking neuron cannot perform the absolute value operation, but a pair of spiking neurons can.

[0054] With reference to FIG. 6, an input connections with online learning 131 in the embedding network 130 combines the input event streams 190-1 to 190-N with a set of weights W_I to provide signals to a set of embedding neurons 135, with $y(t) = W_I \cdot x(t)$, where $y(t)$ is the activity of the embedding neurons 135 and $x(t)$ is the activity of the input connections 131. That is, activity $y(t)$ represents output of the embedding network, activity $x(t)$ represents input of the embedding network. The activity $y(t)$ of the embedding neurons 135 becomes the embedded

encoding streams 139 including embedded encoding streams 139-1 to 139-M.

[0055] The learning configuration and control module 160 uses an online learning rule to set the values in W_I . An example of the online learning rule is disclosed by Tetzlaff C., Kolodziejcki, C., Timme, M., Tsodyks, M., Wörgötter, F. in an article entitled “Synaptic scaling enables dynamically distinct short-and long-term memory formation” in BMC neuroscience 14(1), P415 in 2013. For example, the learning configuration and control module 160 applies a rule according to Tetzlaff which updates the weights W_I such that,

$$\Delta w_{ij} = \eta \left(\int_{\Delta t} x_j(t) y_i(t) dt + \kappa \left(\int_{\Delta t} y_{tgt} - y_i(t) dt \right) w_{ij}^2 \right); \quad (1)$$

Δw_{ij} represents adjustment to weight w_{ij} in W_I ;

η represents a learning rate;

Δt represents relative timing between timing of a postsynaptic spike and timing of a presynaptic spike;

t represents time;

dt indicates the integral with respect to time;

$x_j(t)$ represents presynaptic traces in the input of the embedding network 130;

$y_i(t)$ represents postsynaptic traces in the input of the embedding network 130;

κ represents a post synaptic learning rate;

y_{tgt} represents a target postsynaptic firing rate;

w_{ij} represents a weight applied to $x_j(t)$ for $y_i(t) = w_{ij} \cdot x_j(t)$.

[0056] In this embodiment, non-learned inhibitory connections between all embedding neurons, once a single embedding neuron begins to be active, suppress activity of all other neurons. This encourages a particular pattern of input events to be associated with the activity of a particular embedding neuron. That is, the embedding network 130 utilizes lateral inhibitory connections 136a between embedding neurons 135 of the embedding network 130 to perform first type of winner-takes-all algorithm, such as a typical winner-takes-all (WTA) algorithm. The inhibitory connections 136a between neurons contain inhibitory connections 136-2 to 136-M.

[0057] Jonke, Z., Legenstein, R., Habenschuss, S., and Maass, W. disclose a learning rule in an article entitled “Feedback inhibition shapes emergent computational properties of cortical microcircuit motifs” in Journal of Neuroscience 37(35), 8511-8523 in 2017.

[0058] With reference to FIG. 7, in another embodiment, the learning configuration and control module 160 uses a similar approach but adopt a learning rule according to Jonke to adapt the weights W_I . Under this rule, the learning configuration and control module 160 updates the weights W_I such that:

$$\Delta w_{ij} = \begin{cases} \eta e^{-w_{ij}+1} x_j(t) & \text{if } \Delta t > 0, \\ -\eta y_i(t) & \text{if } \Delta t < 0, \end{cases} \quad (2)$$

Δw_{ij} represents adjustment to weight w_{ij} in W_I

η represents a learning rate;

e represents Euler's number;

Δt represents relative timing between timing of a postsynaptic spike and timing of a presynaptic spike;

t represents time;

$x_j(t)$ represents the activity of the pre-synaptic neuron j ;

$y_i(t)$ represents the activity of the post-synaptic neuron i ;

w_{ij} represents a weight applied to $x_j(t)$ for $y_i(t) = w_{ij} \cdot x_j(t)$.

[0059] In another embodiment, the embedding network 130 uses a population of inhibitory neurons 136b in the place of direct inhibitory connections between the embedding neurons 135. The inhibitory neurons receive input from all embedding neurons 135, and send inhibitory outputs to all embedding neurons 135. This network design permits several embedding neurons to be simultaneously active, and increases a number of embedding patterns that can be learned by the embedding network for a given number of embedding neurons. That is, the embedding network 130 utilizes a population of inhibitory connections between embedding neurons 135 of the embedding network 130 to perform second type of winner-takes-all algorithm, such as a soft-winner-takes-all (sWTA) algorithm.

[0060] During the learning state of the system 100, the learning of the embedding network 130 is enabled. During the operation state of the system 100, the learning of the embedding network 130 is disabled.

[0061] Decoding network:

[0062] With reference to FIG. 8, the decoding network 140 is responsible for predicting the input signals based on the embedded encoding streams 139. The decoding network 140 filters the M nominal embedded encoding streams to obtain a filtered version $\hat{y}(t)$ of the M nominal embedded encoding streams, filters the N nominal sensor event streams to obtain a filtered version $\hat{x}(t)$ of the N nominal sensor event streams, and generates N nominal reconstructed event streams $\hat{z}(t)$ as output of the decoding network based on weighted combination of the filtered version $\hat{y}(t)$ of the M nominal embedded encoding streams, where $\hat{z}(t) = W_o \cdot \hat{y}(t)$.

[0063] Specifically, the embedded encoding streams 139 including embedded encoding streams 139-1 to 139-M are filtered with low-pass exponential filters 141-1 to 141-M in a low-pass exponential filter unit 141, and the input event streams 190 including the input event streams 190-

1 to 190-N are filtered with low-pass exponential filters 142-1 to 142-N in a low-pass exponential filter unit 142. The filtered embedded encoding streams are referred to as $\hat{y}(t)$, and the filtered input event streams are referred to as $\hat{x}(t)$.

[0064] The decoding network 140 comprises neurons 145 with weights W_O for self-supervised learning. The decoding network 140 uses the neurons 145 to generate reconstructed event streams 149 from the filtered embedded encoding streams 139. The reconstructed event streams 149 including reconstructed event streams 149-1 to 149-N are represented by $\hat{z}(t)$. The reconstructed event streams $\hat{z}(t)$ are generated by a weighted combination of the filtered embedded encoding streams $\hat{y}(t)$, such that $\hat{z}(t) = W_O \cdot \hat{y}(t)$. The decoding network in the learning state performs a self-supervised learning process to adjust the weight W_O to reduce an error E , where

$$\text{[0065]} \quad E = \|W_O \cdot \hat{y}(t) - \hat{x}(t)\|^2 + \lambda \|W_O\|^2; \quad (3)$$

λ represents a regularization parameter for the ridge regression.

[0066] For example, a self-supervised learning process adjusts the weights W_O to minimize the error E using a ridge-regression process or any other optimization approach. During the learning state of the system 100, the learning of the decoding network 140 is enabled. During the operation state of the system 100, the learning of the decoding network 140 is disabled.

[0067] Detection module:

[0068] With reference to FIG. 9, the detector module 150 compares the reconstructed event streams 149 against the input event streams 190 after being filtered through the exponential low-pass filters 152. The exponential low-pass filters 152 includes exponential low-pass filters 152-1 to 152-N. The detector module 150 comprises a subtraction calculator 155, a summation calculator 156, and a comparator 157. The subtraction calculator 155 includes a plurality of subtractors 155-1 to 155-N. Each of the subtractors 155-1 to 155-N computes the absolute difference $|\hat{z}_i - \hat{x}_i|$. The variable $\hat{z}_i$ represents one of the reconstructed event streams 149-1 to 149-N. The variable $\hat{x}_i$ represents a filtered version of one of input event streams 190-1 to 190-N output by one of the exponential low-pass filters 152. The results of each of the subtractors 155-1 to 155-N are combined in the summation unit 156 to generate a total difference, and the total difference is compared against a threshold by the comparator 157. Output of the comparator 157 indicates whether an anomaly is present or not.

[0069] Activity of the subtraction calculator 155, a summation calculator 156, and the input event streams 190 are collected as detected condition information 170 which can be requested by the external system 200. Connections between the modules 120, 130, 140, 150, and 160 may be implemented as wirelines, buses, cables, or fibers.

[0070] The external system 200 may include an on-site server located with the monitored target

system 901 or a remote cloud based server located away from the monitored target system 901. Alternatively, the external system 200 may further include a portable device, such as a mobile phone, or a combination of the portable device and a server.

[0071] Neural networks in the disclosure may be implemented by a field programmable gate array (FPGA), an application-specific integrated circuit (ASIC), a tensor processing unit (TPU), a graphics processing unit (GPU), or a multi-core central processing unit (CPU).

[0072] System function:

[0073] With reference to FIG. 10, a process for adapting and using the equipment anomaly detection system 100 to monitor signals is detailed in the following.

[0074] In step S102, the system 100 is installed and switched to the learning state to receive a set of vibration signals 902 from the monitored target system 901 when the monitored target system 901 operates in a nominal operating state in step S103.

[0075] In step S104, the system 100 is reset, such that the embedding network 130 has random weights W_I , the decoding network 140 has random weights W_O , and learning of the embedding network 130 and the decoding network 140 is switched on. The detection trigger is disabled.

[0076] In step S105, the monitored target is allowed to operate in the known nominal operating state which may include a plurality of nominal operating sub-states. In step S106, the system 100 in the learning state adapts and learns this nominal state by adjusting weights W_I and weights W_O according to the learning rules. This adaptation also known as the learning process of the system 100 is complete when the detection module indicates no anomaly over all normal operation sub-states of the monitored target. The system 100 can be configured to control the duration of this adaptation process.

[0077] In step S108, when the learning of the system 100 completes, and the system 100 transits from the learning state to a learned state, where each of the embedding network 130 and the decoding network 140 also transits from a learning state to a learned state. The weights W_I of the embedding network 130 and the weights W_O of the decoding network 140 are updated during the learning state and fixed in the learned state. The learned state of the embedding network 130 is referred to as a trained embedding network state of the embedding network 130. The learned state of the decoding network 140 is referred to as a trained decoding network state of the decoding network 140.

[0078] In step S109, the monitored target system 901 works in a operating state under monitoring by the system 100.

[0079] In step S110, the system 100 in the operation state operates continuously with the embedding network 130 and the decoding network 140 in the learned state as described above.

The system 100 determines whether an anomaly of the monitored target is detected. The detection module 150 is configured to raise a detection trigger 180 when detecting an anomaly for a desired period, such as one minute.

[0080] In step S112, when detecting an anomaly of the monitored target system 901, the detection module 150 sends the trigger signal 180 to the external system 200, which may for example wake the external system 200. The external system 200 may sent control signals 210 to the system 100 to request detailed information 170 about the anomaly. The detected condition information 170 may then be communicated to the external system 200. The trigger signal 180, the control signals 210, and the detected condition information 170 may be transmitted through wirelines or wireless connections.

[0081] Examples of system operations in the learning state:

[0082] With reference to FIG. 11 and FIG. 12, the learning configuration and control module 160 switches the system 100 between a learning state and a learned state. Examples of the vibration signals 902, input sensor signals 112, input event streams 190, embedded encoding streams 139, and reconstructed event streams 149 are shown in a following Table 1.

Table 1

	Learning state	Learned state
Vibration signals 902	Nominal vibration signals 902a	Monitored vibration signals 902b
Input sensor signals 112	N nominal sensor signal streams 112a	N monitored sensor signal streams 112b
Input event streams 190	N nominal sensor event streams 190a	N monitored sensor event streams 190b
Embedded encoding streams 139	M nominal embedded encoding streams 139a	M monitored embedded encoding streams 139b
Reconstructed event streams 149	N nominal reconstructed event streams 149a	N monitored reconstructed event streams 149b

[0083] In the learning state, the system 100 respectively processes nominal vibration signals 902a, N nominal sensor signal streams 112a, N nominal sensor event streams 190a, M nominal embedded encoding streams 139a, and N nominal reconstructed event streams 149a as vibration signals 902, input sensor signals 112, input event streams 190, embedded encoding streams 139, and reconstructed event streams 149. In the learned state, the system 100 respectively processes monitored vibration signals 902b, N monitored sensor signal streams 112b, N monitored sensor event streams 190b, M monitored embedded encoding streams 139b, and N monitored

reconstructed event streams 149b as vibration signals 902, input sensor signals 112, input event streams 190, embedded encoding streams 139, and reconstructed event streams 149.

[0084] With reference to FIG. 11, the sensor system 110 generates N nominal sensor signal streams 112a from nominal vibration signals 902a. The signal conversion module 120 receives, filters, rectifies, smoothens and analog to digital converts the N nominal sensor signal streams 112a into N nominal sensor event streams 190a.

[0085] In the learning state of the equipment anomaly detection system 100, the embedding network 130 and the decoding network 140 form a cascade of neural networks for receiving and learning the N nominal sensor event streams 190a from N event stream input connections to form a learned state of the cascade of neural networks.

[0086] The embedding network 130 is a spiking neural network for receiving the N nominal sensor event streams 190a from the N event stream input connections as input of the embedding network 130 in the learning state, and outputting M nominal embedded encoding streams 139a as output of the embedding network 130 based on the N nominal sensor event streams 190a. The N is an integer greater than the integer M. The embedding network 130 learns to generate the output of the embedding network 130 from the input of the embedding network 130 and to form a trained embedding network state of the embedding network 130 using spike-timing dependent plasticity (STDP). The embedding network 130 may learn from the N nominal sensor event streams 190a by updating the weights W_I according to a learning rule associated with the formula (1) or (2).

[0087] The embedding network 130 learns to generate the output of the embedding network 130 from the input of the embedding network and to form the trained embedding network state of the embedding network 130 using STDP with a symmetric learning window or STDP with an asymmetric learning window. FIG. 13 shows a Tetzlaff learning rule as an example of the symmetric learning window and a Jonke learning rule as an example of the asymmetric learning window.

[0088] The decoding network 140 is a neural network for receiving the M nominal embedded encoding streams 139a as input of the decoding network 140 in the learning state, and outputting N nominal reconstructed event streams 149a as output of the decoding network 140 based on the M nominal embedded encoding streams 139a. The decoding network 140 learns to generate the output of the decoding network 140 from the input of the decoding network 140 and to form a trained decoding network state of the decoding network 140 based on a rule to reduce difference between the N nominal reconstructed event streams 149a and the N nominal sensor event streams 190a. The decoding network 140 may learn from the M nominal embedded encoding streams 139a by updating the weights W_O according to a learning rule associated with the formula (3).

[0089] The decoding network 140 filters the M nominal embedded encoding streams 139a to obtain a filtered version $\hat{y}(t)$ of the M nominal embedded encoding streams 139a, filters the N nominal sensor event streams 190a to obtain a filtered version $\hat{x}(t)$ of the N nominal sensor event streams 190a, and generates N nominal reconstructed event streams $\hat{z}(t)$ as output of the decoding network 140 based on weighted combination of the filtered version $\hat{y}(t)$ of the M nominal embedded encoding streams 139a, where $\hat{z}(t) = W_o \cdot \hat{y}(t)$. The decoding network 140 in the learning state performs a self-supervised learning process to adjust the weight W_o to reduce an error $E = \|W_o \cdot \hat{y}(t) - \hat{x}(t)\|^2 + \lambda \|W_o\|^2$.

[0090] Examples of system operations in the operation state:

[0091] With reference to FIG. 12, the sensor system 110 generates N monitored sensor signal streams 112b from monitored vibration signals 902b. The signal conversion module 120 receives, filters, rectifies, smoothens and analog to digital converts the N monitored sensor signal streams 112b into the N monitored sensor event streams 190b.

[0092] In an operation state, the cascade of neural networks in the learned state receives N monitored sensor event streams 190b from the N event stream input connections as input of the cascade of neural networks, and outputs N monitored reconstructed event streams 149b as output of the cascade of neural networks based on the N monitored sensor event streams 190b. The N monitored reconstructed event streams 149b represent a reconstructed version of the N monitored sensor event streams 190b.

[0093] The embedding network 130 in the trained embedding network state outputs M monitored embedded encoding streams 139b based on the N monitored sensor event streams 190b.

[0094] The decoding network 140 in the trained decoding network state outputs the N monitored reconstructed event streams 149b based on the M monitored embedded encoding streams 139b.

[0095] The detection module 150 calculates difference between the N monitored sensor event streams 190b and the N monitored reconstructed event streams 149b, and sends the detection trigger 180 when the difference between N monitored sensor event streams 190b and N monitored reconstructed event streams 149b exceeds a threshold difference value.

[0096] The decoding network 140 in the trained decoding network state filters the M monitored embedded encoding streams 139b to obtain a filtered version $\hat{y}_m(t)$ of the M monitored embedded encoding streams 139b, filters the N monitored sensor event streams 190b to obtain a filtered version $\hat{x}_m(t)$ of the N monitored sensor event streams 190b, and generates N monitored reconstructed event streams $\hat{z}_m(t)$ as output of the decoding network 140 based on weighted combination of the filtered version $\hat{y}_m(t)$ of the M monitored embedded encoding streams 139b, and $\hat{z}_m(t) = W_o \cdot \hat{y}_m(t)$.

[0097] The detection module 150 filters the N monitored sensor event streams 190b to obtain a filtered version $\hat{x}_m(t)$ of the N monitored sensor event streams 190b. The detection module 150 comprises the subtraction calculator 155, the summation calculator 156, and the comparator 157.

[0098] The subtraction calculator 155 includes a plurality of subtractors. A subtractor selected from the plurality of subtractors calculates reconstruction induced difference between a monitored sensor event stream in filtered version $\hat{x}_m(t)$ of the N monitored sensor event streams 190b and a monitored reconstructed event stream in the N monitored reconstructed event streams $\hat{z}_m(t)$, such that the plurality of subtractors calculate reconstruction induced differences stream by stream between the filtered version $\hat{x}_m(t)$ of the N monitored sensor event streams 190b and the N monitored reconstructed event streams $\hat{z}_m(t)$.

[0099] The summation calculator 156 calculates summation of the differences between the filtered version $\hat{x}_m(t)$ and the N monitored reconstructed event streams $\hat{z}_m(t)$.

[00100] The comparator 157 compares the summation with a signal difference threshold, and outputs an alert signal 180 representing anomaly of the N monitored sensor event streams 190b when the summation exceeds the signal difference threshold.

[00101] With reference to FIG. 14, in an embodiment, the embedding network 130, the decoding network 140, the detection module 150, and the learning configuration and control module 160 may be implemented as computer executable instructions in an embedding network module 130a, the decoding network module 140a, the detection module 150a, and the learning configuration and control module 160a. The computer executable instructions in module 130a, 140a, 150a, and 160a, which when loaded from a storage device 105 to a main memory 104 and executed by a CPU 102, performs the functions of the embedding network 130, the decoding network 140, the detection module 150, and the learning configuration and control module 160 in the disclosure. The CPU 102 may use a neural processing unit (NPU) 103 to execute functions of the embedding network module 130a and the decoding network module 140a. The NPU 103 may be implemented by a field programmable gate array (FPGA), an application-specific integrated circuit (ASIC), a tensor processing unit (TPU), or a graphics processing unit (GPU).

[00102] A person having ordinary skill in the art understands that each of the modules, units, algorithm, and steps described and disclosed in the embodiments of the present disclosure are realized using electronic hardware or combinations of software for computers and electronic hardware. Whether the functions run in hardware or software depends on the condition of application and design requirement for a technical plan. A person having ordinary skill in the art can use different ways to realize the function for each specific application while such realizations should not go beyond the scope of the present disclosure. It is understood by a person having

ordinary skill in the art that he/she can refer to the working processes of the system, device, and unit in the above-mentioned embodiment since the working processes of the above-mentioned system, device, and unit are basically the same. For easy description and simplicity, these working processes will not be detailed.

[00103] It is understood that the disclosed system, device, and method in the embodiments of the present disclosure can be realized with other ways. The above-mentioned embodiments are exemplary only. The division of the units is merely based on logical functions while other divisions exist in realization. It is possible that a plurality of units or components are combined or integrated in another system. It is also possible that some characteristics are omitted or skipped. On the other hand, the displayed or discussed mutual coupling, direct coupling, or communicative coupling operate through some ports, devices, or units whether indirectly or communicatively by ways of electrical, mechanical, or other kinds of forms.

[00104] The units as separating components for explanation are or are not physically separated. The units for display are or are not physical units, that is, located in one place or distributed on a plurality of network units. Some or all of the units are used according to the purposes of the embodiments. Moreover, each of the functional units in each of the embodiments can be integrated in one processing unit, physically independent, or integrated in one processing unit with two or more than two units.

[00105] If the software function unit is realized and used and sold as a product, it can be stored in a readable storage medium in a computer. Based on this understanding, the technical plan proposed by the present disclosure can be essentially or partially realized as the form of a software product. Or, one part of the technical plan beneficial to the conventional technology can be realized as the form of a software product. The software product in the computer is stored in a storage medium, including a plurality of commands for a computational device (such as a personal computer, a server, or a network device) to run all or some of the steps disclosed by the embodiments of the present disclosure. The storage medium includes a USB disk, a mobile hard disk, a read-only memory (ROM), a random access memory (RAM), a floppy disk, or other kinds of media capable of storing program codes.

[00106] The equipment anomaly detection device of the disclosure uses unsupervised or self-supervised neural networks to learn nominal conditions of a target system, such as a device or a machine. The trained neural networks can reproduce sensory signals of the target system as a neural-network-reconstructed version of the sensory signals in the nominal conditions of a target system. The equipment anomaly detection device may analyze and predict operation conditions of the target system based on the neural-network-reconstructed version exceed a certain level. When

signal difference between the sensory signals and the neural-network-reconstructed version exceed a certain level, the equipment anomaly detection device may issue an alert signal to reflect abnormal operation conditions of the target system, which may be a period ahead of an actual onset of system failure of the target system.

[00107] While the present disclosure has been described in connection with what is considered the most practical and preferred embodiments, it is understood that the present disclosure is not limited to the disclosed embodiments but is intended to cover various arrangements made without departing from the scope of the broadest interpretation of the appended claims.

Claims

1. An equipment anomaly detection device comprising:

an cascade of neural networks for receiving and learning N nominal sensor event streams from N event stream input connections in a learning state to form a learned state of the cascade of neural networks, wherein the cascade of neural networks in the learned state receives N monitored sensor event streams from the N event stream input connections as input of the cascade of neural networks in an operation state, and outputs N monitored reconstructed event streams as output of the cascade of neural networks based on the N monitored sensor event streams, and the N monitored reconstructed event streams represent a reconstructed version of the N monitored sensor event streams; and

a detection module for calculating difference between the N monitored sensor event streams and the N monitored reconstructed event streams, wherein the detection module sends a detection trigger when the difference between N monitored sensor event streams and N monitored reconstructed event streams exceeds a threshold difference value.

2. The equipment anomaly detection device of claim 1, wherein the cascade of neural networks comprises:

an embedding network being a spiking neural network for receiving the N nominal sensor event streams from the N event stream input connections as input of the embedding network in the learning state, and outputting M nominal embedded encoding streams as output of the embedding network based on the N nominal sensor event streams, wherein the N is an integer greater than the integer M, and the embedding network learns to generate the output of the embedding network from the input of the embedding network and to form a trained embedding network state of the embedding network using spike-timing dependent plasticity (STDP); and

a decoding network being a neural network for receiving the M nominal embedded encoding streams as input of the decoding network in the learning state, and outputting N reconstructed event streams as output of the decoding network based on the M nominal embedded encoding streams, wherein the decoding network learns to generate the output of the decoding network from the input of the decoding network and to form a trained decoding network state of the decoding network based on a rule to reduce difference between the N reconstructed event streams and the N nominal sensor event streams.

3. The equipment anomaly detection device of claim 2, wherein:

the embedding network in the trained embedding network state outputs M monitored embedded encoding streams based on the N monitored sensor event streams; and

the decoding network in the trained decoding network state outputs the N monitored reconstructed

event streams based on the M monitored embedded encoding streams.

4. The equipment anomaly detection device of claim 3, wherein the embedding network learns to generate the output of the embedding network from the input of the embedding network and to form the trained embedding network state of the embedding network using STDP with a symmetric learning window.

5. The equipment anomaly detection device of claim 4, wherein the embedding network comprises embedding neurons with a set of weights W_I , activity $y(t)$ represents output of the embedding network, activity $x(t)$ represents input of the embedding network, $y(t) = W_I \cdot x(t)$, Δw_{ij} represents adjustment to weight w_{ij} in W_I ,

$$\Delta w_{ij} = \eta \left(\int_{\Delta t} x_j(t) y_i(t) dt + \kappa \left(\int_{\Delta t} y_{tgt} - y_i(t) dt \right) w_{ij}^2 \right);$$

η represents a learning rate;

Δt represents relative timing between timing of a postsynaptic spike and timing of a presynaptic spike;

t represents time;

dt indicates an integral with respect to time;

$x_j(t)$ represents the pre-synaptic activity of neuron j ;

$y_i(t)$ represents the post-synaptic activity of neuron i ;

κ represents a post synaptic learning rate;

y_{tgt} represents a target activity for a post-synaptic neuron;

w_{ij} represents a weight applied to $x_j(t)$ for $y_i(t) = w_{ij} \cdot x_j(t)$.

6. The equipment anomaly detection device of claim 3, wherein the embedding network learns to generate the output of the embedding network from the input of the embedding network and to form the trained embedding network state of the embedding network using STDP with an asymmetric learning window.

7. The equipment anomaly detection device of claim 6, wherein the embedding network comprises embedding neurons with a set of weights W_I , activity $y(t)$ represents output of the embedding network, activity $x(t)$ represents input of the embedding network, $y(t) = W_I \cdot x(t)$, Δw_{ij} represents adjustment to weight w_{ij} in W_I ,

$$\Delta w_{ij} = \begin{cases} \eta e^{-w_{ij}+1} x_j(t) & \text{if } \Delta t > 0 \\ -\eta y_i(t) & \text{if } \Delta t < 0 \end{cases}$$

η represents a learning rate;

e represents Euler's number;

Δt represents relative timing between timing of a postsynaptic spike and timing of a presynaptic

spike;

t represents time;

$x_j(t)$ represents the activity of pre-synaptic neuron j ;

$y_i(t)$ represents the activity of post-synaptic neuron i ;

w_{ij} represents a weight applied to $x_j(t)$ for $y_i(t) = w_{ij} \cdot x_j(t)$.

8. The equipment anomaly detection device of claim 3, wherein the embedding network utilizes lateral inhibitory connections between embedding neurons of the embedding network to perform first type of winner-takes-all algorithm.

9. The equipment anomaly detection device of claim 3, wherein the embedding network utilizes a population of inhibitory connections between embedding neurons of the embedding network to perform second type of winner-takes-all algorithm.

10. The equipment anomaly detection device of claim 3, wherein the decoding network filters the M nominal embedded encoding streams to obtain a filtered version $\hat{y}(t)$ of the M nominal embedded encoding streams, filters the N nominal sensor event streams to obtain a filtered version $\hat{x}(t)$ of the N nominal sensor event streams, and generates N nominal reconstructed event streams $\hat{z}(t)$ as output of the decoding network based on weighted combination of the filtered version $\hat{y}(t)$ of the M nominal embedded encoding streams, $\hat{z}(t) = W_o \cdot \hat{y}(t)$, and the decoding network in the learning state performs a self-supervised learning process to adjust the weight W_o to reduce an error $E = \|W_o \cdot \hat{y}(t) - \hat{x}(t)\|^2 + \lambda \|W_o\|^2$;

λ represents a regularization parameter for ridge regression.

11. The equipment anomaly detection device of claim 10, wherein the decoding network in the learning state uses a ridge-regression process to minimize the error E .

12. The equipment anomaly detection device of claim 10, wherein the decoding network utilizes exponential low-pass filters to filter the M nominal embedded encoding streams and the N nominal sensor event streams.

13. The equipment anomaly detection device of claim 10, wherein the decoding network in the trained decoding network state filters the M monitored embedded encoding streams to obtain a filtered version $\hat{y}_m(t)$ of the M monitored embedded encoding streams, filters the N monitored sensor event streams to obtain a filtered version $\hat{x}_m(t)$ of the N monitored sensor event streams, and generates N monitored reconstructed event streams $\hat{z}_m(t)$ as output of the decoding network based on weighted combination of the filtered version $\hat{y}_m(t)$ of the M monitored embedded encoding streams, and $\hat{z}_m(t) = W_o \cdot \hat{y}_m(t)$;

the detection module filters the N monitored sensor event streams to obtain a filtered version

$\hat{x}_m(t)$ of the N monitored sensor event streams, and the detection module comprises:

a subtraction calculator comprising a plurality of subtractors, wherein a subtractor selected from the plurality of subtractors calculates reconstruction induced difference between a monitored sensor event stream in filtered version $\hat{x}_m(t)$ of the N monitored sensor event streams and a monitored reconstructed event stream in the N monitored reconstructed event streams $\hat{z}_m(t)$, such that the plurality of subtractors calculate reconstruction induced differences stream by stream between the filtered version $\hat{x}_m(t)$ of the N monitored sensor event streams and the N monitored reconstructed event streams $\hat{z}_m(t)$;

a summation calculator for calculating summation of the differences between the filtered version $\hat{x}_m(t)$ and the N monitored reconstructed event streams $\hat{z}_m(t)$; and

a comparator for comparing the summation with a signal difference threshold, and outputting an alert signal representing anomaly of the N monitored sensor event streams when the summation exceeds the signal difference threshold.

14. The equipment anomaly detection device of claim 3 further comprising:

a signal conversion module for receiving, filtering, rectifying, and analog to digital converting N nominal sensor signal streams into the N nominal sensor event streams, and N monitored sensor signal streams into the N monitored sensor event streams.

15. An equipment anomaly detection method comprising:

using an cascade of neural networks to receive and learn N nominal sensor event streams from N event stream input connections in a learning state to form a learned state of the cascade of neural networks;

using the cascade of neural networks in the learned state to receive N monitored sensor event streams from the N event stream input connections as input of the cascade of neural networks in an operation state, and output N monitored reconstructed event streams as output of the cascade of neural networks based on the N monitored sensor event streams, wherein the N monitored reconstructed event streams represent a reconstructed version of the N monitored sensor event streams; and using a detection module to calculate difference between the N monitored sensor event streams and the N monitored reconstructed event streams; and

the detection module sending a detection trigger when the difference between N monitored sensor event streams and N monitored reconstructed event streams exceeds a threshold difference value.

16. The equipment anomaly detection method of claim 15, wherein the cascade of neural networks comprises:

an embedding network being a spiking neural network for receiving the N nominal sensor event streams from the N event stream input connections as input of the embedding network in the

learning state, and outputting M nominal embedded encoding streams as output of the embedding network based on the N nominal sensor event streams, wherein the N is an integer greater than the integer M, and the embedding network learns to generate the output of the embedding network from the input of the embedding network and to form a trained embedding network state of the embedding network using spike-timing dependent plasticity (STDP); and
 a decoding network being a neural network for receiving the M nominal embedded encoding streams as input of the decoding network in the learning state, and outputting the N reconstructed event streams as output of the decoding network based on the M nominal embedded encoding streams, wherein the decoding network learns to generate the output of the decoding network from the input of the decoding network and to form a trained decoding network state of the decoding network based on a rule to reduce difference between the N reconstructed event streams and the N nominal sensor event streams.

17. The equipment anomaly detection method of claim 16, wherein:

the embedding network in the trained embedding network state outputs M monitored embedded encoding streams based on the N monitored sensor event streams; and
 the decoding network in the trained decoding network state outputs the N monitored reconstructed event streams based on the M monitored embedded encoding streams.

18. The equipment anomaly detection method of claim 17, wherein the embedding network learns to generate the output of the embedding network from the input of the embedding network and to form the trained embedding network state of the embedding network using STDP with a symmetric learning window.

19. The equipment anomaly detection method of claim 18, wherein the embedding network comprises embedding neurons with a set of weights W_I , activity $y(t)$ represents output of the embedding network, activity $x(t)$ represents input of the embedding network, $y(t) = W_I \cdot x(t)$, Δw_{ij} represents adjustment to weight w_{ij} in W_I ,

$$\Delta w_{ij} = \eta \left(\int_{\Delta t} x_j(t) y_i(t) dt + \kappa \left(\int_{\Delta t} y_{tgt} - y_i(t) dt \right) w_{ij}^2 \right);$$

η represents a learning rate;

Δt represents relative timing between timing of a postsynaptic spike and timing of a presynaptic spike;

t represents time;

dt indicates an integral with respect to time;

$x_j(t)$ represents the pre-synaptic activity of neuron j ;

$y_i(t)$ represents the post-synaptic activity of neuron i ;

κ represents a post synaptic learning rate;

y_{tgt} represents a target activity for a post-synaptic neuron;

w_{ij} represents a weight applied to $x_j(t)$ for $y_i(t) = w_{ij} \cdot x_j(t)$.

20. The equipment anomaly detection method of claim 17, wherein the embedding network learns to generate the output of the embedding network from the input of the embedding network and to form the trained embedding network state of the embedding network using STDP with an asymmetric learning window.

21. The equipment anomaly detection method of claim 20, wherein the embedding network comprises embedding neurons with a set of weights W_I , activity $y(t)$ represents output of the embedding network, activity $x(t)$ represents input of the embedding network, $y(t) = W_I \cdot x(t)$, Δw_{ij} represents adjustment to weight w_{ij} in W_I ,

$$\Delta w_{ij} = \begin{cases} \eta e^{-w_{ij}+1} x_j(t) & \text{if } \Delta t > 0, \\ -\eta y_i(t) & \text{if } \Delta t < 0 \end{cases}$$

η represents a learning rate;

e represents Euler's number;

Δt represents relative timing between timing of a postsynaptic spike and timing of a presynaptic spike;

t represents time;

$x_j(t)$ represents the activity of pre-synaptic neuron j ;

$y_i(t)$ represents the activity of post-synaptic neuron i ;

w_{ij} represents a weight applied to $x_j(t')$ for $y_i(t') = w_{ij} \cdot x_j(t')$.

22. The equipment anomaly detection method of claim 17, wherein the embedding network utilizes lateral inhibitory connections between embedding neurons of the embedding network to perform first type of winner-takes-all algorithm.

23. The equipment anomaly detection method of claim 17, wherein the embedding network utilizes a population of inhibitory connections between embedding neurons of the embedding network to perform second type of winner-takes-all algorithm.

24. The equipment anomaly detection method of claim 17, wherein the decoding network filters the M nominal embedded encoding streams to obtain a filtered version $\hat{y}(t)$ of the M nominal embedded encoding streams, filters the N nominal sensor event streams to obtain a filtered version $\hat{x}(t)$ of the N nominal sensor event streams, and generates N nominal reconstructed event streams $\hat{z}(t)$ as output of the decoding network based on weighted combination of the filtered version $\hat{y}(t)$ of the M nominal embedded encoding streams, $\hat{z}(t) = W_O \cdot \hat{y}(t)$, and the decoding network

in the learning state performs a self-supervised learning process to adjust the weight W_o to reduce an error $E = \|W_o \cdot \hat{y}(t) - \hat{x}(t)\|^2 + \lambda \|W_o\|^2$;

λ represents a regularization parameter for the ridge regression.

25. A chip, comprising:

a processor, configured to call and run a computer program stored in a memory, to cause a device in which the chip is installed to execute any of the methods of claims 15 to 24.

26. A computer readable storage medium, in which a computer program is stored, wherein the computer program causes a computer to execute any of the methods of claims 15 to 24.