

METHOD AND SYSTEM FOR MACHINE FAILURE PREDICTION

Field of Invention

The invention relates to machine failure prediction techniques, especially, a method and system for machine failure prediction involving very large temporal dependencies, typically more than 100 time steps.

Background

It is of high importance to accurately predict machine failures/faults which occur very rarely, i.e. after a very long time gap, as such prediction allows the user to take preventive actions to avoid the potential machine failures. This is particularly important for equipment which is used to provide basic services such as tele-communication, transportation etc.

Various methods have been used to predict machine failures based on history of machine failures and additional causal parameters like ambient temperature, humidity, overload and age of the equipment etc. In general, these methods predict machine failures based on either statistical or artificial intelligence (AI) techniques. It is much easier to predict a machine failure which occurs after a few time steps e.g. 10 time steps. However, as the inter-failure gap increases, the existing methods, e.g. methods using hidden Markov model or genetic algorithm, cannot provide correct and accurate predictions.

Recurrent Neural Network (RNN) with back propagation of errors as training method has been used for machine condition monitoring and diagnosis.

It will be understood by a person skilled in the art that the RNN can also be used for machine failures prediction since machine failure prediction can be treated as a sequence modelling process. However, the RNN based methods with random initial weights using Back propagation through time (BPTT) cannot handle
5 machine failure prediction with a long inter-failure gap, e.g. more than 15 time steps, mainly due to vanishing gradient or exploding gradient problem. To overcome this difficulty, specialized architectures have been proposed, however, use of specialized architecture will increase complexity and sometimes these architectures can only work in special situations.

10 **Summary of Invention**

Embodiments of the invention provide a machine failure prediction solution which can handle very long time gaps between two successive failures, e.g. more than 1000 time steps.

According to one aspect of the invention, a method for machine failure
15 prediction is provided. The method comprises:

identifying, by a first processor in a first computer system, a plurality of basic memory depth values based on a composite sequence of a machine failure history;

ascertaining, by the first processor, a set of weight values for at least one
20 of the plurality of basic memory depth values according to a pre-stored table

which is stored in a memory and includes a plurality of mappings wherein each mapping relates a basic memory depth value to one set of weight values; and

predicting, by the first processor, a future failure using a Back Propagation Through Time (BPTT) trained Recurrent Neural Network (RNN) based on the
5 ascertained weight values for the at least one of the plurality of basic memory depth values,

wherein a first set of weight values related to a first basic memory depth value in the pre-stored table is ascertained based on a second set of weight values related to a second basic memory depth value, wherein the second basic
10 memory depth value is less than the first basic memory depth value by a predetermined value.

According to a second aspect of the invention, a system for machine failure prediction is provided. The system comprises: a first processor and a memory communicably coupled thereto,

15 wherein the first memory is configured to store data to be executed by the first processor,

wherein the first processor is configured to

identify a plurality of basic memory depth values based on a machine failure history;

ascertain a set of weight values for at least one of the plurality of basic memory depth values according to a pre-stored table which is stored in a memory and includes a plurality of mappings wherein each mapping relates a basic memory depth value to one set of weight values; and

- 5 predict a future failure using a Back Propagation Through Time (BPTT) trained Recurrent Neural Network (RNN) based on the ascertained weight values for the at least one of the plurality of basic memory depth values,

 wherein a first set of weight values related to a first basic memory depth value in the pre-stored table is ascertained based on a second set of weight values related to a second basic memory depth value, wherein the second basic memory depth value is less than the first basic memory depth value by a predetermined value.

10

Embodiments of the invention use a standard Back Propagation Through Time (BPTT) trained Recurrent Neural Network (RNN) and an iterative method to solve machine failure prediction problems. These problems have very complex composite sequences of machine failure history and the composite sequences involve very large time gaps of machine failures. The iterative method is used to ascertain weight values of RNN related to elementary sequences or initial weight values of RNN related to a composite sequence. With this iterative method, the initial weight values of RNN ascertained for solving a complex machine failure prediction problem are close to the final weight values when convergence occurs. Thus, the vanishing gradient or exploding gradient

15

20

problem mentioned above would be effectively avoided. With the method for machine failure prediction provided in the embodiments of the invention, even very large time gap machine failure prediction problems, e.g. memory depth is more than 3000 time steps and inter failure gap is more than 1000 time steps,
5 can be solved in an acceptable time period.

Brief Description of the Drawings

The invention will be described in detail with reference to the accompanying drawings, in which:

Figure 1A is a flowchart illustrating a method for predicting a machine
10 failure according to a first embodiment of the invention;

Figure 1B shows an example of a composite sequence of a machine failure history;

Figure 1C shows as an example of a composite sequence including two basic memory depth values 5 and 6;

15 Figure 2 illustrates the architecture of a RNN trained with Back Propagation Through Time (BPTT) used for machine failure prediction according to one embodiment of the invention;

Figure 3A is a graph illustrating weight variation between elementary problems with basic memory depth values 225 and 230;

Figure 3B is a graph illustrating weight variation between elementary problems with basic memory depth values 225 and 425;

Figure 4 is a flow chart illustrating the method for generating the pre-stored table according to one embodiment of the invention;

5 Figure 5A is a table illustrating the basic memory depth values 10-1000, and the corresponding minimum number of iterations for epochs for convergence, the predetermined error limit/threshold value and time gap for failure;

10 Figure 5B is a table illustrating weight values related to the initial memory depth value 10 when convergence occurs;

Figure 5C is a table illustrating a set of weight values ascertained for an elementary problem with a basic memory depth value 12 when convergence occurs;

15 Figure 6A is a table illustrating a group of composite sequences which are predicted by using the method proposed in Figure 1;

Figure 6B is a graph in planar coordinates illustrating weight variations between calculated composite weight values and simulated composite weight values for composite sequence 375-390-475-375-390;

20 Figure 7 is a table illustrating simulation results of two composite sequences with non-perfect circular pattern;

Figure 8A is a flow chart illustrating a method for predicting machine failure according to a second embodiment of the invention;

Figure 8B shows an example of a composite sequence;

Figure 8C shows six intermediate composite sequences generated
5 according to the second embodiment of the invention.

Detailed Description of Embodiments of the Invention

In the following description, numerous specific details are set forth in order to provide a thorough understanding of various illustrative embodiments of the invention. It will be understood, however, to one skilled in the art, that
10 embodiments of the invention may be practiced without some or all of these specific details. It is understood that the terminology used herein is for the purpose of describing particular embodiments only, and is not intended to limit the scope of the invention.

As mentioned above, in embodiments of the invention, a solution for
15 machine failure prediction involving very large temporal dependencies is provided. This solution uses a standard BPTT trained RNN and an iterative method to solve a complex composite machine failure prediction problem.

Figure 1A is a flowchart illustrating a method 100 for predicting a machine failure according to a first embodiment of the invention.

20 In **block 101**, a processor in a computer identifies a plurality of basic memory depth values based on a composite sequence of a machine failure history.

Figure 1B shows an example of a composite sequence of a machine failure history. In this sequence, 0 represents negative failure indicator, i.e. there is no machine failure; 1 represents positive failure indicator, i.e. there is a machine failure. The composite sequence includes a plurality of different basic memory depth value. Each basic memory depth value refers to the number of zeroes between two successive positive failure indicators. As shown in Figure 1(b), in this example, the composite sequence includes three basic memory depth values: 375, 390 and 475.

In addition, it is to be appreciated by a skilled person in the art that the composite sequence of a machine failure history may be inputted by a user.

In **block 102**, the processor in the computer ascertains weight values for each identified basic memory depth value based on a pre-stored table. Typically, the processor directly retrieves the weight values for each identified basic memory depth value from the pre-stored table.

The pre-stored table includes a plurality of mappings. Each of the mappings relates a basic memory depth value to one set of weight values. The pre-stored table is generated using an iterative method which will be explained in detail below. In general, a first set of weight values related to a first basic memory depth value in the pre-stored table is ascertained based on a second set of weight values related to a second basic memory depth value in the pre-stored table, wherein the second basic memory depth value is less than the first basic memory depth value by a predetermined value. The predetermined value

may be any integer not greater than 5. It should be noted that as the size of the pre-stored table is too big, it is not included in the description.

In **block 103**, the processor in the computer ascertains initial composite weight values by calculating weighted average of the ascertained weight values
5 for all of the identified basic memory depth values.

Firstly, we will explain why the weighted average of the ascertained weight values of the basic memory depth values is a suitable estimate for ascertaining initial weight values of a composite sequence.

Take the composite sequence in Figure 1C as an example, this composite
10 sequence includes two basic memory depth values 5 and 6. It has been found that the final weight values for modelling this composite sequence are very close to weighted average of weight values for modelling elementary sequences with basic memory depth values 5 and 6. Hence, weighted average should be a suitable estimate for ascertaining initial weight values of a composite sequence.
15 This conclusion is of crucial importance since it allows to model arbitrarily composite sequence using weight values related to basic memory depth values. This concept is also applicable when the composite sequence contains more than 2 basic memory depth values. Since weights related to basic memory depth values can be pre-computed, initial weight values for a composite sequence can
20 be obtained by looking up for corresponding weight values for related to basic memory depth values and then calculating weighted average of the weight

values related to the basic memory depth values. Thus, the process for modelling a composite sequence would be greatly simplified.

In this embodiment, the weighted average of the ascertained basic weight values for all of the identified basic memory depth values is calculated based on the count of appearances of each identified basic memory depth value in the composite sequence, i.e. the number of appearances/occurrences of each identified basic memory depth value in the composite sequence. Assuming the weight values ascertained for the basic memory depth values 375, 390 and 475 are respectively W1, W2 and W3, the weighted average of the ascertained weight values is calculated according to the following Equation (1):

$$(W1 + 2 * W2 + W3) / 4 \quad (1)$$

In **block 104**, the processor in the computer predicts a future machine failure using a BPTT trained RNN based on the ascertained composite weight values for the composite sequence.

Figure 2 illustrates the architecture of a RNN trained with Back Propagation Through Time (BPTT) used for machine failure prediction according to one embodiment of the invention. Referring to Figure 2, the RNN has multiple layers of neurons including input layer with a plurality of input units, hidden layer with a plurality of hidden units and an output layer with an output unit. In the RNN, hidden units of time period (t-1), also called state units, are also used as input at time period (t). This simply means that the network at time period (t) is

fed with the earlier state of the network at time period (t-1) to ensure that current prediction is influenced by earlier actual values.

In embodiments of the invention, the RNN is used to predict machine failure only based on a failure history sequence without other inputs like performance counter, temperature, wind speed etc. In addition, cue unit in input layer is used to improve the RNN to remember long time gaps. Experiments have proved that use of cue unit can significantly reduce number of iterations for convergence.

To solve a machine failure prediction problem involving very large temporal dependencies, the number of state units to be used must be adequate for inputting the ascertained weight values. In one embodiment of the invention, before the processor runs the RNN, the processor determines the number of state units based on a maximum value of the identified basic memory depth values. In one example of the embodiment, if the maximum value of the identified basic memory depth values is not more than 350, the processor determines that the number of state units to be used is 60 units; if the maximum value of the identified memory depth values is more than 350, but not more than 1000, the processor determines that the number of state units to be used is 120 units.

Method for generating the pre-stored table

The method for generating the pre-stored table will be explained in detail below. As mentioned above, the pre-stored table is generated using an iterative method, i.e. the set of weight values related to a first basic memory depth value is ascertained based on the set of weight values which has been ascertained for

a second basic memory depth value which is less than the first basic memory depth value by a predetermined value, e.g. any integer not greater than 5.

Firstly, we will explain why this iterative method can be used to effectively ascertain weight values related to a corresponding basic memory depth value.

5 In embodiments of the invention, a basic memory depth value corresponds to an elementary sequence/problem for machine failure prediction, i.e. each elementary sequence has a fixed basic memory depth value. It has been found that weight values of RNN for solving two elementary problems do not vary much when the basic memory depth values of the two elementary
10 problems are close to each other. In other words, when a RNN having a specific structure is used to solve two similar elementary problems, e.g. running a first elementary sequence with a basic memory depth value 5 and a second elementary sequence with a basic memory depth value 6, a first and a second set of weight values which are used when convergence occurs are identified
15 respectively for these two elementary sequences. It has been found that there are only minor differences between the corresponding weight values for these two elementary sequences which are being used when convergence occurs.

 With this conclusion, weight values related to an elementary sequence with a very large time gap can be ascertained by using an iterative way.
20 Specifically, weight values related to an elementary sequence with a basic memory depth value 6 can be ascertained using weight values of a RNN which can handle an elementary sequence with a basic memory depth value 5. Then,

weight values for an elementary sequence with a basic memory depth value 7 can be ascertained using weight values of the RNN which can handle the elementary sequence with a basic memory depth value 6. Accordingly, weight values for an elementary sequence with a very large time gap, e.g. more than 5 100 time gaps, can be ascertained iteratively. It should be noted that using this method to ascertain weight values related to an elementary sequence with a very large time gap, the RNN has to remember more parameters, so the number of state units in the RNN are also to be increased.

To further prove the conclusion explained above, the weight values 10 related to elementary problems with basic memory depth values 225 and 230 are compared as shown in Figure 3A. In Figure 3A, values on x-axis indicate indices of weight values and values on y-axis indicate variation percentage of weight values. Weight values are numbered by starting with 1 and moving onto the input layer from left to right. For each input unit, weight values for all 15 connections between that input unit and all hidden units are numbered sequentially. Referring to Figure 3A, majority of the weight values do not vary beyond 10%. In this example, 120 state units are used, the total number of input units is $123=120$ (state units) $+1$ (unit for failure occurrence indicator) $+1$ (cue unit) $+1$ (bias in input layer), and the number of hidden units is $121=120+1$ (bias in 20 hidden layer). Accordingly, the number of weight values is $123 \times 121=14883$. As shown in Figure 3A, only 8.93% weight values vary by more than 5% when the weight values related to the basic memory depth values 225 and 230 are compared. While as shown in Figure 3B which is a graph illustrating weight

variation between elementary problems with basic memory depth values 225 and 425, when the two basic memory depth values are not close to each other, obvious variation of weight values have been found. As shown in Figure 3B, contrary to the situation shown in Figure 3A, in this example, 69.09% weight
5 values vary by more than 5%.

According to the aforementioned conclusion, the basic weight values of RNN for solving an elementary problem with a basic memory depth value n can be used as initial weight values of RNN for solving an elementary problem with a basic memory depth value $n+x$, wherein the value of x may be an integer not
10 greater than 5.

Figure 4 is a flow chart illustrating the method 400 for generating the pre-stored table according to one embodiment of the invention.

In **block 401**, the processor in the computer selects an initial memory depth value, e.g. 10, and initial weight values, e.g. [-1.35, 1.35], for predicting a
15 future machine failure using the RNN.

It should be noted that the processor used for generating the pre-stored table and the processor used for machine failure prediction may be provided as the same processor.

In **block 402**, the processor in the computer predicts an elementary
20 sequence corresponding to the initial memory depth value, i.e. the elementary sequence with a basic memory depth value 10, by running a RNN based on a

predetermined error threshold, a minimum number of epochs for convergence, the selected initial basic memory depth value and random initial weight values until convergence occurs.

It should be noted that in some cases, the minimum number of epochs for
5 convergence may not be necessary.

Figure 5A is a table illustrating the basic memory depth values 10-1000, and the corresponding minimum number of iterations for epochs for convergence, the predetermined error limit/threshold value and time gap for failure. It should be noted that since RNN acts as a function approximator,
10 sometimes failure may not be predicted exactly where expected but within a small number of time gaps deviation (typically 1 to 5). The table in Figure 5A also lists exact time gaps where next failure is predicted when there is a deviation. If there is no deviation, the corresponding value is left empty.

In **block 403**, the final weight values for predicting the elementary
15 sequence corresponding to a current basic memory depth value are stored in a file when convergence occurs.

In order to solve machine failure prediction problem involving large temporal dependencies, the hidden units to be used by the RNN is 120, accordingly, the number of weight values is $123 \times 121 = 14883$. As the number is
20 too big to be included in the description, a simplified example is used to illustrate the ascertained weight values according to the method proposed in embodiments of the invention.

For illustration purpose only, Figure 5B shows a table illustrating weight values related to the initial memory depth value 10 when convergence occurs. In this example, 20 hidden units are used, accordingly, the number of weight values shown in the table is $23 \times 21 = 483$.

5 In **block 404**, the processor in the computer determines whether a predetermined maximum value of basic memory depth value is reached, if yes, the process ends; otherwise, the flow goes to block 405.

 In **block 405**, the current basic memory depth value is increased by a predetermined value, e.g. 1. It should be noted that in other embodiments, the
10 predetermined value may be any integer not greater than 5.

 In **block 406**, the processor in the computer predicts an elementary sequence corresponding to the current basic memory depth value, by running a RNN based on the set of final weight values which is most recently stored in block 403, a predetermined error threshold value and a minimum number of
15 epochs for convergence, until convergence occurs. It should be noted that in some cases, the minimum number of epochs for convergence is not necessary. Then, the flow goes back to block 403.

 Figure 5C shows a table illustrating a set of weight values ascertained for an elementary problem with a basic memory depth value 12 when convergence
20 occurs. This set of weight values are ascertained based on the weight values shown in Figure 5B for an elementary problem with a basic memory depth value 10.

Accordingly, a pre-stored table can be generated based on the iterative method above. This table includes a plurality of mappings, each mapping has a basic memory depth value and a set of corresponding basic weight values ascertained related to this basic memory depth value.

5 **Simulation Results**

Figure 6A is a table illustrating a group of composite sequences which are predicted by using the method proposed in Figure 1. In the table, each of the composite sequences has a perfect circular pattern. For example, the circular pattern of the composite sequence 20-30-60-20-30 is 20-30-60, accordingly, the
10 corresponding composite memory depth value of the composite sequence is $20+30+60=110$. For each composite sequence, the corresponding number of iterations for convergence, time gap to failure and the error limit/threshold value are listed in the table. It can be seen from the last composite sequence that the failures are predicted to happen after more than 1000 time gaps. The simulation
15 results in the table indicate that the method proposed in the embodiments of the invention is suitable for machine failure prediction involving very large time dependencies.

Figure 6B is a graph in planar coordinates illustrating weight variations between composite weight values for composite sequence 375-390-475-375-390
20 calculated using the method proposed in Figure 1 and the simulation results of composite weight values when convergence occurs. It has been shown in Figure 6B that only 20.13% of the calculated weight values vary by more than

5% when compared with the corresponding simulated weight values when convergence occurs.

Figure 7 shows a table illustrating simulation results of two composite sequences with non-perfect circular pattern. For example, the circular pattern of the composite sequence 200-260-360-190-260 is 200-260-360. However, the first value of the second circular pattern in the composite sequence is 190 instead of 200. the circular pattern of the composite sequence 210-250-300-420-200-250-300 is 210-250-300-420,. However, the first value in the second circular pattern is 200 instead of 210. The simulation results shown in Figure 7 indicate that even if the circular pattern of a composite sequence is not perfect, the method proposed by embodiments of the invention can also predict the future machine failure properly. It means that the BPTT trained RNN proposed in embodiments of the invention is not only suitable for solving machine failure prediction problem involving large memory depth, but also holds generalization property.

Figure 8A is a flow chart illustrating a method for predicting machine failure according to a second embodiment of the invention. In this embodiment, the composite sequence to be predicted includes only two basic memory depth values as shown in Figure 8B.

In **block 801**, a processor in a computer identifies the two basic memory depth values from the composite sequence of a machine failure history.

As shown in Figure 8B, the composite sequence in the embodiment includes two basic memory depth values: a smaller value 8 and a larger value 15.

In **block 802**, the processor in the computer retrieves weight values for the smaller value 8 of the two basic memory depth values from a pre-stored table.

It should be noted that in this embodiment, the pre-stored table is generated using the iterative method mentioned above.

In **block 803**, the processor in the computer generated six intermediate composite sequences. Each of the intermediate composite sequences includes two basic memory depth values, one is the smaller value 8, and the other one is an intermediate value.

As shown in Figure 8C, the six intermediate values for the six intermediate composite sequences are respectively 9, 10, 11, 12, 13, and 14.

It should be noted that the number of the intermediate composite sequences and the intermediate value for each intermediate composite sequence in this embodiment are for illustration purpose only and do not limit the scope of the invention.

In **block 804**, the processor in the computer predicts the six intermediate composite sequences according to an ascending order of the intermediate values, by running the BPTT trained RNN based on the weight values

ascertained for the smaller value 8, or final weight values which are used for predicting a preceding intermediate composite sequence which has a smaller intermediate value, until convergence occurs.

Specifically, the weight values for the intermediate composite sequence
5 (1) are ascertained based on the weight values for the smaller value 8 retrieved from the pre-stored table. The weight values for the intermediate composite sequence (2) are ascertained based on final weight values which are used for predicting the intermediate composite sequence (1). The final weight values are the weight values which are being used for predicting the intermediate composite
10 sequence (1) when convergence occurs. The rest may be deduced by analogy, the weight values for the intermediate composite sequences (3) to (6) are ascertained respectively based on final weight values which are used for predicting the intermediate composite sequences (2) to (5).

In **block 805**, the processor in the computer predicts the composite
15 sequence as shown in Figure 8B by running the BPTT trained RNN based on the final weight values which are used for predicting the intermediate composite sequence (6), until convergence occurs.

It is to be appreciated by a person skilled in the art that although for convenience of explanation, the composite sequence in Figure 8B includes two
20 quite small basic memory depth values; the method described in Figure 8A is easily suitable for solving a composite machine failure prediction problem which involves very large temporal dependencies. Further, the final weight values

ascertained for the particular composite sequence which includes only two basic memory depth values using the method shown in Figure 8A may be stored in the pre-stored table and used for ascertaining final weight values of a more complex composite sequence which includes the particular composite sequence.

5 As will be appreciated from the above, embodiments of the invention provide a method for predicting machine failures involving very large time gaps. This method uses a standard RNN trained with BPTT and an iterative method for ascertaining initial weight values to be used for predicting machine failures. It has been tested that the method disclosed in embodiments of the invention can
10 solve a machine failure prediction problem involving very large temporal dependencies, e.g. a composite memory depth value is more than 3000 time steps and the inter failure gap is more than 1000 time steps. It takes very less time for convergence in the RNN since the RNN starts with weight values which are close to the final weight values, e.g. problems with memory depth more than
15 3000 time steps take no more than 40000 epochs for convergence. Further, even if the machine failure prediction problem does not follow a perfect circular pattern, which is common in real life, the method disclosed by embodiments of the invention can be used to predict a future failure properly.

 It is to be understood that the embodiments and features described above
20 should be considered exemplary and not restrictive. Many other embodiments will be apparent to those skilled in the art from consideration of the specification and practice of the invention. The scope of the invention should, therefore, be determined with reference to the appended claims, along with the full scope of

equivalents to which such claims are entitled. Furthermore, certain terminology has been used for the purposes of descriptive clarity, and not to limit the disclosed embodiments of the invention.

Claims

1. A method for machine failure prediction, the method comprising:

identifying, by a first processor in a first computer system, a plurality of
basic memory depth values based on a composite sequence of a machine
5 failure history;

ascertaining, by the first processor, a set of weight values for at least one
of the plurality of basic memory depth values according to a pre-stored table
which is stored in a memory and includes a plurality of mappings wherein each
mapping relates a basic memory depth value to one set of weight values; and

10 predicting, by the first processor, a future failure using a Back Propagation
Through Time (BPTT) trained Recurrent Neural Network (RNN) based on the
ascertained weight values for the at least one of the plurality of basic memory
depth values,

wherein a first set of weight values related to a first basic memory depth
15 value in the pre-stored table is ascertained based on a second set of weight
values related to a second basic memory depth value, wherein the second basic
memory depth value is less than the first basic memory depth value by a
predetermined value.

2. The method according to claim 1, further comprising:

determining, by the first processor, the number of state units to be used by the BPTT trained RNN based on a maximum value of the identified basic memory depth values.

3. The method according to claim 2, wherein the number of state units is further
5 determined based on the following rules:

if the maximum value of the identified basic memory depth values is not greater than 350, the number of state units is 60 units;

if the maximum value of the identified basic memory depth values is more than 350 units, but not greater than 1000 units, the number of state units is 120
10 units.

4. The method according to any preceding claim, wherein the pre-stored table is generated according to the following steps:

a. selecting, by a second processor in a second computer, an initial basic memory depth value and initial weight values;

15 b. predicting, by the second processor, an elementary sequence corresponding to the initial basic memory depth value by running a RNN based on a predetermined error threshold, the selected initial basic memory depth value and initial weight values until convergence occurs;

c. storing, by the second processor, a set of final weight values which are used for predicting the elementary sequence related to a current basic memory depth value when convergence occurs;

d. increasing, by the second processor, the current basic memory depth value by the predetermined value if a predetermined maximum value of the basic memory depth value is not reached;

e. predicting, by the second processor, an elementary sequence corresponding to the current basic memory depth value by running a RNN based on a predetermined error threshold, the current basic memory depth value and the set of final weight values which is most recently stored in step c, until convergence occurs; and

f. repeating, by the second processor, the steps c to e until the predetermined maximum value of the basic memory depth value is reached.

5. The method according to claim 4, wherein in step b and/or step e, the elementary sequence is further predicted based on a predetermined minimum number of iterations for convergence.

6. The method according to any preceding claim, wherein the predetermined value is an integer not greater than 5.

7. The method according to any preceding claim, wherein the step of ascertaining a set of weight values for at least one of the plurality of basic memory depth values according to a pre-stored table, comprises:

ascertaining a set of weight values for each of the plurality of basic memory depth values according to the pre-stored table; and

the method further comprises:

ascertaining a set of initial composite weight values by calculating
5 weighted average of weight values ascertained for each of the plurality of basic memory depth values; and

wherein the future failure is predicted based on the ascertained initial composite weight values.

8. The method according to claim 7, wherein the weighted average of weight
10 values is calculated based on a count of appearances for each basic memory depth value in the composite sequence.

9. The method according to any one of claim 1 to claim 6, wherein the composite sequence includes two basic memory depth values,

wherein the step of ascertaining a set of weight values for at least one of
15 the plurality of basic memory depth values according to a pre-stored table, comprises:

ascertaining a set of weight values for a smaller value of the two basic memory depth values according to the pre-stored table; and

the step of predicting a future failure, comprises:

generating at least one intermediate composite sequence, wherein each intermediate composite sequence includes two basic memory depth values, one is the smaller value of the two basic memory depth values, the other one is an intermediate value between the two basic memory depth values;

5 predicting the at least one intermediate composite sequence one by one according to an ascending order of the corresponding intermediate value of the at least one intermediate composite sequence, by running the BPTT trained RNN based on the set of weight values ascertained for the smaller value, or final weight values which are used for predicting an preceding intermediate composite
10 sequence, until convergence occurs; and

 predicting the composite sequence by running the BPTT trained RNN based on final weight values which are used for predicting a last one of the at least one intermediate composite sequence, until convergence occurs.

10. The method according to any one of claim 4 to claim 9, wherein the first
15 processor and the second processor are provided as the same processor.

11. A system for machine failure prediction, the system comprising: a first processor and a memory communicably coupled thereto,

 wherein the first memory is configured to store data to be executed by the first processor,

20 wherein the first processor is configured to

identify a plurality of basic memory depth values based on a machine failure history;

ascertain a set of weight values for at least one of the plurality of basic memory depth values according to a pre-stored table which is stored in a memory and includes a plurality of mappings wherein each mapping relates a basic memory depth value to one set of weight values; and

predict a future failure using a Back Propagation Through Time (BPTT) trained Recurrent Neural Network (RNN) based on the ascertained weight values for the at least one of the plurality of basic memory depth values,

wherein a first set of weight values related to a first basic memory depth value in the pre-stored table is ascertained based on a second set of weight values related to a second basic memory depth value, wherein the second basic memory depth value is less than the first basic memory depth value by a predetermined value.

12. The system according to claim 11, wherein the first processor is further configured to determine the number of state units be used by the BPTT trained RNN based on a maximum value of the identified basic memory depth values.

13. The system according to claim 12, wherein the first processor is further configured to determine the number of state units based on the following rules:

20

if the maximum value of the identified basic memory depth values is not greater than 350, the number of state units is 60 units;

if the maximum value of the identified basic memory depth values is more than 350 units, but not greater than 1000 units, the number of state units is 120 units.

14. The system according to any one of claim 11 to claim 13, further comprising: a second processor in a second computer, wherein the second processor is configured to generate the pre-stored table according to the following steps:

a. select an initial basic memory depth value and initial weight values;

10 b. predict an elementary sequence corresponding to the initial basic memory depth value by running a RNN based on a predetermined error threshold, the selected initial basic memory depth value and initial weight values until convergence occurs;

15 c. store a set of final weight values which are used for predicting the elementary sequence related to a current basic memory depth value when convergence occurs;

d. increase the current basic memory depth value by the predetermined value if a predetermined maximum value of the basic memory depth value is not reached;

e. predict an elementary sequence corresponding to the current basic memory depth value by running a RNN based on a predetermined error threshold, the current basic memory depth value and the set of final weight values which is most recently stored in step c, until convergence occurs; and

5 f. repeat the steps c to e until the predetermined maximum value of the basic memory depth value is reached.

15. The system according to claim 14, wherein the second processor is further configured to predict the elementary sequence based on a predetermined minimum number of iterative for convergence.

10 16. The system according to any one of claim 11 to claim 15, wherein the predetermined value is an integer not greater than 5.

17. The system according to any one of claim 11 to claim 16, wherein the first processor is further configured to ascertain a set of weight values for each of the plurality of basic memory depth values according to the pre-stored table,

15 ascertain a set of initial composite weight values by calculating weighted average of weight values ascertained for each of the plurality of basic memory depth values; and

 predict the future failure based on the ascertained initial composite weight values.

18. The system according to claim 17, wherein the first processor is further configured to calculate the weighted average of the weight values based on a count of appearances for each basic memory depth value in the composite sequence.

- 5 19. The system according to any one of claim 11 to claim 16, wherein the composite sequence includes two basic memory depth values,

wherein the first processor is further configured to

ascertain a set of weight values for a smaller value of the two basic memory depth values according to the pre-stored table;

- 10 generate at least one intermediate composite sequence, wherein each intermediate composite sequence includes two basic memory depth values, one is the smaller value of the two basic memory depth values, the other one is an intermediate value between the two basic memory depth values;

- predict the at least one intermediate composite sequence one by one
15 according to an ascending order of the corresponding intermediate value of the at least one intermediate composite sequence, by running the BPTT trained RNN based on the set of weight values ascertained for the smaller value, or final weight values which are used for predicting an preceding intermediate composite sequence, until convergence occurs; and

predict the composite sequence by running the BPTT trained RNN based on final weight values which are used for predicting a last one of the at least one intermediate composite sequence, until convergence occurs.

20. The system according to any one of claim 14 to claim 19, wherein the first
5 processor and the second processor are provided as the same processor.

21. A non-transitory computer readable medium comprising computer program code for machine failure prediction, wherein the computer program code, when executed, is configured to cause a processor in a computer system perform a method according to any one of claim 1 to claim 10.