

METHOD AND SYSTEM FOR FAULT DETECTION AND DIAGNOSTIC FOR A BUILDING MANAGEMENT SYSTEM

CROSS-REFERENCE TO RELATED APPLICATION

- 5 [0001] This application claims the benefit of priority of Singapore Patent Application No. 10202100115U, filed on 6 January 2021, the content of which being hereby incorporated by reference in its entirety for all purposes.

TECHNICAL FIELD

- 10 [0002] The present invention generally relates to a method and a system for fault detection and diagnostic for a building management system.

BACKGROUND

- [0003] A building management system may include various sub-systems such as chiller system, lighting system, elevator system associated to a building, which may be prone to faults. For example, chiller system faults in the building could reduce the thermal comfort, reduce the energy efficiency, and increase the operating cost. In order to solve all issues caused by the chiller system faults, fault detection and diagnostic (FDD) should be implemented. In general, there are several methods to implement FDD, e.g. model-based method or expert-based method.

- [0004] Chiller data obtained either directly from sensors installed in the chiller system or from building management system (BMS) to capture chiller system key performance indicators (KPIs) are usually without label. It is difficult for user to know whether the data are of normal chiller operation or there may be potential faults. Typically, a chiller FDD system model-based method requires a lot of chiller historical labelled data to train accurate models. Therefore, user needs to do data labelling to classify chiller data into normal operation data or different groups of fault operation data. This requires a lot of effort as it is very time consuming to label the chiller data if it is performed manually.

- 30 [0005] On the other hand, the expert-based method uses pre-established knowledge library comprised of collection of rules built upon industry best practice and manufacturer specification sheet to detect and diagnose chiller faults. Since the knowledge library is usually created based on default chiller operation environment, it might not be able to capture the actual operation environment of the chiller system.

Actual chiller operation environment could change according to climate condition, cooling load, different set-points, etc., and hence such method might not be able to perform accurate FDD. Typically, the knowledge library may be manually updated. However, the manual process of updating each rule in the knowledge library requires a lot of effort and can be overwhelming if the number of rules is large in order to capture all possible chiller faults.

[0006] With a lot of rules in the knowledge library, the expert-based method is required to check all of them when FDD is implemented. Likewise, the model-based method also requires a lot of computation resources and it is very time consuming in order to capture all possible chiller fault models both in training and FDD processes. Therefore, both methods are resource intensive in order to cover all possible chiller faults.

[0007] With the above-mentioned utilization of a lot of resources to do manual data labelling, mismatch of chiller operation environment setting and resource intensive of both methods to cover all possible chiller faults, a need therefore exists to provide an improved method and system for fault detection and diagnostic for a building management system, that seek to overcome, or at least ameliorate, one or more of the deficiencies in the conventional methods, such as but not limited to, achieve a more seamless and efficient fault detection and diagnostic for a building management system. It is against this background that the present invention has been developed.

SUMMARY

[0008] According to a first aspect of the present invention, there is provided a method of fault detection and diagnostic for a building management system, using at least one processor, the method comprising:

obtaining operational data in relation to a plurality of system parameters associated to a plurality of operational components of the building management system; determining an operation status among a plurality of types of operation statuses, including a normal operation status and a plurality of types of fault operation statuses with respect to the operational data based on a fault condition repository configured based on expert knowledge information, the fault condition repository comprising a plurality of fault conditions in relation to the plurality of system parameters, wherein each fault condition corresponds to a type of fault operation status; and

labelling the operational data based on the operation status determined to produce labelled data in relation to the operation status determined.

[0009] According to a second aspect of the present invention, there is provided a system for fault detection and diagnostic for a building management system, the system comprising:

a memory; and

at least one processor communicatively coupled to the memory and configured to:

obtain operational data in relation to a plurality of system parameters associated to a plurality of operational components of the building management system;

determine an operation status among a plurality of types of operation statuses, including a normal operation status and a plurality of types of fault operation statuses with respect to the operational data based on a fault condition repository configured based on expert knowledge information, the fault condition repository comprising a plurality of fault conditions in relation to the plurality of system parameters, wherein each fault condition corresponds to a type of fault operation status; and

label the operational data based on the operation status determined to produce labelled data in relation to the operation status determined.

[0010] According to a third aspect of the present invention, there is provided a computer program product, embodied in one or more non-transitory computer-readable storage mediums, comprising instructions executable by at least one processor to perform a method of fault detection and diagnostic for a building management system according to the first aspect.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] Embodiments of the present invention will be better understood and readily apparent to one of ordinary skill in the art from the following written description, by way of example only, and in conjunction with the drawings, in which:

FIG. 1A depicts a flow diagram of a method of fault detection and diagnostic (FDD) for a building management system, according to various embodiments of the present invention;

FIG. 1B depicts another flow diagram of a method of fault detection and diagnostic for a building management system, according to various embodiments of the present invention;

FIG. 1C depicts yet another flow diagram of a method of fault detection and diagnostic for a building management system, according to various embodiments of the present invention;

FIG. 2 depicts a schematic block diagram of a system for fault detection and diagnostic for a building management system, according to various embodiments of the present invention, such as corresponding to the method shown in FIG. 1A;

FIG. 3 depicts a schematic block diagram of an exemplary computer system in which a system for fault detection and diagnostic for a building management system, according to various embodiments of the present invention, may be realized or implemented;

FIG. 4A depicts a schematic diagram of a Building Management System (BMS) including an FDD system, a chiller system and a Building Energy Management System (BEMS), according to various example embodiments;

FIG. 4B illustrates a schematic illustrating the components in a chiller system, according to various example embodiments;

FIG. 4C shows a schematic diagram of sensors installed in each component in a chiller system, according to various example embodiments;

FIG. 5A shows a block diagram illustrating modules within the FDD system, according to various example embodiments;

FIG. 5B shows a block diagram illustrating components of the storage of the FDD system, according to various example embodiments;

FIG. 6 shows an example flow diagram illustrating the exemplary steps of chiller relationship map creation in a chiller relationship map creation module, according to various example embodiments;

FIG. 7 shows a simplified example of the chiller relationship map, according to various example embodiments of the present invention;

FIG. 8 shows an example of a more comprehensive chiller relationship map of two chiller components, according to various example embodiments of the present invention;

FIG. 9 shows an example flow diagram illustrating the exemplary steps of normal chiller KPIs tree creation in the normal chiller KPIs tree creation module, according to various example embodiments of the present invention;

FIG. 10 shows an example of the normal chiller KPIs tree, according to various example embodiments of the present invention;

FIG. 11 shows an example flow diagram illustrating the exemplary steps of knowledge library creation in a knowledge library creation module, according to various example embodiments of the present invention;

FIG. 12 shows an example representation on how by using the chiller relationship map and the normal chiller KPIs tree, a fault type list of the knowledge library can be created, according to various example embodiments of the present invention;

FIG. 13 shows another example representation on how by using the chiller relationship map and the normal chiller KPIs tree, the fault type list of the knowledge library can be created, according to various example embodiments of the present invention;

FIG. 14 shows an example of the fault type list in the knowledge library, according to various example embodiments of the present invention;

FIG. 15 shows an example of the data structure of the fault rule library in the knowledge library, according to various example embodiments of the present invention;

FIG. 16 shows a flow diagram illustrating the exemplary steps of chiller FDD process of a FDD process module, according to various example embodiments of the present invention;

FIG. 17 shows an example of the data structure of the chiller operational data, according to various example embodiments of the present invention;

FIG. 18A shows an example of the data structure of the FDD report, according to various example embodiments of the present invention;

FIG. 18B shows another example of the data structure of the FDD report, according to various example embodiments of the present invention;

FIG. 19 shows a flow diagram illustrating the exemplary steps of finding abnormal chiller KPIs utilizing normal chiller KPIs, according to various example embodiments of the present invention;

FIG. 20 shows a flowchart illustrating a new rule creation process in a new rule creation module, according to various example embodiments of the present invention;

FIG. 21 shows an example of the data structure of a rule-model creation status table, according to various example embodiments of the present invention;

5 FIG. 22 shows an example of the data structure of a rule-model pairing table, according to various example embodiments of the present invention;

FIG. 23 shows an example of a new rule creation process by using the chiller relationship map, according to various example embodiments of the present invention;

10 FIG. 24 shows another example of the new rule creation process by the using chiller relationship map, according to various example embodiments of the present invention;

FIG. 25 shows a flowchart illustrating a validation process in a validation module, according to various example embodiments of the present invention;

15 FIG. 26 is a flowchart illustrating a fault handling process, according to various example embodiments of the present invention;

FIG. 27 shows an example flow diagram illustrating a model creation process, according to various example embodiments of the present invention;

20 FIG. 28A shows a flow diagram illustrating the exemplary steps of a data labelling process in a data labelling module, according to various example embodiments of the present invention;

FIG. 28B shows an example table illustrating the labelled data with respect to the chiller operational data, according to various example embodiments of the present invention;

25 FIG. 29 shows an example flow diagram illustrating the exemplary steps of a model training process in a model training module, according to various example embodiments of the present invention;

FIG. 30 shows an example flow diagram illustrating the exemplary steps of executing fault predicting model process, according to various example embodiments of the present invention;

30 FIG. 31 shows an example of the data structure of a matching score table, according to various example embodiments of the present invention;

FIG. 32 shows an example flow diagram illustrating the exemplary steps of updating knowledge library process, according to various example embodiments of the present invention; and

FIG. 33 shows a diagram illustrating a process for updating threshold parameters, according to various example embodiments of the present invention.

DETAILED DESCRIPTION

5 [0012] Various embodiments of the present invention provide a method and system for fault detection and diagnostic for a building management system.

[0013] FIG. 1A depicts a flow diagram of a method 100a of fault detection and diagnostic for a building management system, using at least one processor. The method 100a comprises: obtaining (at 102), operational data in relation to a plurality of system
10 parameters associated to a plurality of operational components of the building management system; determining (at 104), an operation status among a plurality of types of operation statuses, including a normal operation status and a plurality of types of fault operation statuses with respect to the operational data based on a fault condition repository configured based on expert knowledge information, the fault condition
15 repository comprising a plurality of fault conditions in relation to the plurality of system parameters, wherein each fault condition corresponds to a type of fault operation status; and labelling (at 106), the operational data based on the operation status determined to produce labelled data in relation to the operation status determined.

[0014] In relation to the above-mentioned labelled data in relation to the operation
20 status determined, it may be labelled data in relation to data values with respect to the plurality of system parameters obtained under a type of fault operation status, or labelled data in relation to data values with respect to the plurality of system parameters obtained under a normal operation status. For example, in the case that the operational data is obtained when a fault has occurred and a type of fault operation status is
25 determined (or detected or identified) based on the fault condition repository, the operational data obtained may be subsequently labelled with respect to the type of fault operation status determined. The labelled data may be used to train a fault model (which may also be interchangeably referred to as a fault predicting model) for detecting the type of fault operation status (e.g., after validation of the labelled data). Accordingly,
30 an expert-based method may be used to label the operational data to produce the labelled data where such labelled data may be subsequently used for training a new fault model or re-training an existing fault model. Accordingly, various embodiments provide interaction between an expert-based method and a model-based method.

[0015] In various embodiments, the above-mentioned determining an operation status with respect to the operational data based on the fault condition repository comprises determining a type of fault operation status, and the method 100a further comprises invoking (or executing), for the type of fault operation status determined, a fault model associated to the type of fault operation status determined for detecting a fault corresponding to the type of fault operation status. In other words, the expert-based method may also be used to select a fault model in FDD execution. The fault model may be for detecting a fault corresponding to such type of fault operation status determined by the expert-based method. In this regard, the method 100a does not need to run all fault models to detect one fault.

[0016] In various embodiments, the above-mentioned determining a type of fault operation status produces a first detection result, while the above-mentioned invoking, for the type of fault operation status determined, a fault model associated to the type of fault operation status produces a second detection result. The method 100a further comprises matching the first detection result and the second detection result to obtain a match result, determining whether the match result satisfy a predetermined fault condition repository update condition; and updating the fault condition repository if the match result does not satisfy the predetermined fault condition repository update condition. Accordingly, various embodiments advantageously facilitate automatic update of the fault condition repository. In various embodiments, each system parameter of the plurality of system parameters comprises a threshold parameter associated to the system parameter defined in the fault condition repository, and each fault condition of the plurality of fault conditions is configured based on one or more of the threshold parameters.

[0017] In various embodiments, the above-mentioned determining a type of fault operation status comprises determining at least one violated system parameter of the plurality of system parameters corresponding to at least one threshold parameter respectively of the threshold parameters being violated based on the operational data, in response to the at least one violated system parameter determined, determining whether there is an existing type of fault operation status corresponding to the at least one violated system parameter based on the plurality of fault conditions, and in response to determining that there is an existing type of fault operation status corresponding to the at least one violated threshold parameter, determining the existing type of fault operation status as the operation status.

- [0018] The expert knowledge information may be, or include, for example, manufacturer or product specification sheet, information created according to best practice system operation knowledge that have been verified and validated by the system expert and/or expert knowledge (e.g., consultant, service engineer). In various embodiments, the fault condition repository comprises correlation information of the plurality of system parameters and fault type parameter association information of the plurality of types of fault operation statuses. For each type of fault operation status, the fault type parameter association information indicates one or more system parameters of the plurality of system parameters being associated to the type of fault operation status. For example, for a particular type of fault operation status, the fault type parameter association information may indicate a set of associated system parameters for such type of fault operation status. The fault type parameter association information may further indicate the respective threshold parameter which is associated with each system parameter.
- [0019] As for the correlation information of the plurality of system parameters, it may indicate, for each system parameter of the plurality of system parameters, one or more correlated system parameters correlating with the system parameter. Said differently, the correlation information of the plurality of system parameters may indicate, for each system parameter of the plurality of system parameters, one or more correlated system parameters with which the system parameter is correlated. For example, the correlation information of the plurality of system parameters may be or include a relationship map of the system parameters. In various embodiments, the correlation information of the plurality of system parameters may further indicate which system parameter of the plurality of system parameters belongs to which system component of the plurality of system components. The correlation information of the plurality of system parameters may also indicate a relationship of each system parameter with one or more operational components of the plurality of operational components.
- [0020] The correlation information of the plurality of system parameters and the fault type parameter association information of the plurality of types of fault operation statuses may be configured based on the expert knowledge information. For example, the correlation information and the fault type parameter association information may be constructed based on system manufacturer specification, industry best practice and/or chiller expert knowledge. For example, the expert knowledge information may

include a prebuilt relationship map obtained from manufacturer specification, in a non-limiting example.

[0021] In various embodiments, the method 100a further comprises determining, for each violated system parameter of the at least one violated system parameter, one or more correlated system parameters with which the at least one violated threshold parameter is correlated based on the correlation information, determining, for each violated system parameter of the at least one violated system parameter, an existing type of fault operation status to which the one or more correlated system parameters are associated based on the fault type parameter association information, and associating, for each violated system parameter of the at least one violated system parameter, the violated system parameter to the existing type of fault operation status to produce an updated set of system parameters associated to the existing type of fault operation status in the fault condition repository. Accordingly, the updated set of system parameters associated to the existing type of fault operation status may be included in the fault type parameter association information and a fault condition corresponding to the existing type of fault operation status in the fault condition repository.

[0022] In various embodiments, when there is no existing type of fault operation status corresponding to the at least one violated system parameter, the method 100a further comprises creating a new fault condition in relation to an unknown type of fault operation status, wherein the new fault condition is created based on the at least one violated system parameter and one or more correlated system parameters correlating with the at least one violated system parameter. The created new fault condition corresponding to a new type of fault operation status may be added into the fault condition repository. The fault type parameter association information in the fault condition repository may also be updated to include the new type of fault operation status and its associated system parameters as well as respective threshold parameters.

[0023] In various embodiments, the above-mentioned updating the fault condition repository if the match result does not satisfy the predetermined fault condition repository update condition comprises updating a fault condition associated to the type of fault operation status determined based on an objective function to obtain a maximum match result between the first detection result and the second detection result, wherein said updating the fault condition comprises adjusting one or more threshold parameters associated to one or more system parameters, respectively, corresponding to the fault condition.

[0024] In various embodiments, the method 100a further comprises performing validation of the labelled data in relation to the operation status determined to produce validated labelled data for training a fault model.

5 [0025] In various embodiments, the method 100a further comprises training the fault model (e.g., is produced/re-produced or created/re-created by being trained) based on the validated labelled data. For example, a new fault model corresponding to a type of fault operation status may be created based on the validated labelled data in relation to the type of fault operation status. In the case a fault model corresponding to a type of fault operation status already exists (i.e., an existing fault model), the validated
10 labelled data may be new validated labelled data in relation to the type of fault operation status and various embodiments may automatically re-train the existing fault model corresponding to the type of fault operation status based on the new validated labelled data (e.g., based on the new validated labelled data and existing validated labelled data which already exists prior to the new validated labelled data).

15 [0026] In various embodiments, the operational data comprises measured data in relation to the plurality of systems parameters of the plurality of operational components obtained from one or more sensors associated to the operational components. For example, the one or more sensors may be located at a building of the building management system, such as installed at the respective operational
20 components, including equipments, of the building. The operational data may further comprise data obtained from a Building Energy Management System (BEMS) associated with the building management system or building.

[0027] Accordingly, various embodiments provide a method and system for a fault detection and diagnostic which employs and integrates a fault condition repository
25 configured based on expert knowledge information (e.g., expert-based method), and machine learning (e.g., model-based method). The method 100a of a fault detection and diagnostic according to various embodiments of the present invention advantageously automatically labels the operational data to produce labelled data (e.g., for fault model training and validation, such as creating a new fault model or re-training existing fault
30 models based on new validated labelled data), automatically updates the fault condition repository (e.g., updates a fault condition, creating a new fault condition) based on actual real operation environment and minimize resources utilization. Through the integration of both expert-based method and model-based method, various embodiments enable a FDD system to start with only a few fault conditions to cover

significant types of system faults, and automatically creates new fault conditions together with its associated fault model only when needed. With few fault conditions in the fault condition repository and their associated fault models (e.g., in the initial stage) various embodiments may advantageously minimize computation resources to perform data labeling, model training and FDD process as compared to traditional FDD implementation.

[0028] FIG. 1B depicts another flow diagram of a method 100b of fault detection and diagnostic for a building management system, using at least one processor. The method 100b is similar to the method 100a, and as such common features is not described in detail in the of brevity. As illustrated in FIG. 1B, at 102, operational data may be obtained. At 104, an operation status with respect to the operational data may be determined. For example, the operation status of the operational data may be in relation to whether the operational data is obtained under normal condition or faulty condition. Based on the results of fault detection using the fault condition repository, at 106, the operational data may be labelled into normal operation status or a type of fault operation status. The fault operation status may be multiple types of fault conditions. At 108, the labelled data may be validated. Based on the validated labelled data, at 110, the fault model may be trained or re-trained. Training is for the fault model that does not exist in the fault model database system, while re-training is for the fault model that is already existing in the fault model database system. At 112, the fault model may be invoked to detect a fault associated to the determined fault type (corresponding to “the type of fault operation status determined” as described hereinbefore) found using the fault condition repository. At 114, the results of detection using the fault condition repository and machine learning model may be compared. If the comparison result, e.g., matching score, falls below a matching score threshold, at 116, the fault condition repository may be updated (e.g., changing the threshold parameter(s) in the fault condition repository). The process may be repeated to determine the operation status of the new operational data using the updated fault condition repository.

[0029] FIG. 1C depicts another flow diagram of a method 100c of fault detection and diagnostic for a building management system, using at least one processor. The method 100c is similar to the method 100a and 100b, and as such common features is not described in detail in the interest of brevity. As illustrated in FIG. 1C, at 108, the labelled data may be validated. If the validated labelled data is sufficient for training a fault model, at 110, the fault model may be trained or re-trained (training is for the fault

type model that does not exist in the fault model database system, while re-training is for the fault model that is already existing in the fault model database system) and, at 112, the fault model may be invoked to detect a fault associated to the determined fault type found using the fault condition repository. Otherwise, at 112, the process directly
5 invokes the fault model to detect a fault associated to the determined fault type found using the fault condition repository.

[0030] The process may then continue similar to the method 100b. For example, at 114, the results of detection using the fault condition repository and machine learning model may be compared. If the comparison or matching result, e.g., matching score,
10 falls below a matching score threshold, at 116, the fault condition repository may be updated (e.g., changing the threshold parameter(s) in the fault condition repository). The process may be repeated to determine the operation status of the new operational data using the updated fault condition repository.

[0031] In various other embodiments, a pre-trained fault model (machine learning model) may be registered (or added) into the fault model database system, and may be
15 invoked directly to update the fault condition repository (e.g., to obtain the above-mentioned matching result and updating the fault condition repository if the matching result does not satisfy the matching score threshold (predetermined fault condition repository update condition)).

[0032] In various other embodiments, the fault model re-training process may be
20 skipped depending on which one the system trust most, fault condition repository or machine learning model. For example, some well-trained, proven to be accurate machine learning model may be added into the system (fault model database system), and need not be re-trained. Instead, it may be used to improve the fault conditions in
25 the fault condition repository. In other words, the fault model re-training process may be a configurable step to be turned on/off.

[0033] FIG. 2 depicts a schematic block diagram of a system 200 for fault detection and diagnostic for a building management system, according to various embodiments of the present invention, such as corresponding to the method 100a of fault detection and diagnostic for a building management system as described hereinbefore according
30 to various embodiments of the present invention. The system 200 comprises a memory 202, and at least one processor 204 communicatively coupled to the memory 202 and configured to: obtain operational data in relation to a plurality of system parameters associated to a plurality of operational components of the building management system;

determine an operation status among a plurality of types of operation statuses, including a normal operation status and a plurality of types of fault operation statuses with respect to the operational data based on a fault condition repository configured based on expert knowledge information, the fault condition repository comprising a plurality of fault conditions in relation to the plurality of system parameters, wherein each fault condition corresponds to a type of fault operation status; and label the operational data based on the operation status determined to produce labelled data in relation to the operation status determined.

[0034] It will be appreciated by a person skilled in the art that the at least one processor 204 may be configured to perform the required functions or operations through set(s) of instructions (e.g., software modules) executable by the at least one processor 204 to perform the required functions or operations. Accordingly, as shown in FIG. 2, the system 200 may comprise an operational data obtaining module (or an operational data obtaining circuit) 206 configured to obtain operational data in relation to a plurality of system parameters associated to a plurality of operational components of the building management system; an operation status determining module (or an operation status determining circuit) 208 configured to determine an operation status among a plurality of types of operation statuses, including a normal operation status and a plurality of types of fault operation statuses with respect to the operational data based on a fault condition repository configured based on expert knowledge information, the fault condition repository comprising a plurality of fault conditions in relation to the plurality of system parameters, wherein each fault condition corresponds to a type of fault operation status; and a data labelling module (or a data labelling circuit) 210 configured to label the operational data based on the operation status determined to produce labelled data in relation to the operation status determined.

[0035] It will be appreciated by a person skilled in the art that the above-mentioned modules are not necessarily separate modules, and one or more modules may be realized by or implemented as one functional module (e.g., a circuit or a software program) as desired or as appropriate without deviating from the scope of the present invention. For example, two or more of the operational data obtaining module 206, the operation status determining module 208, and the data labelling module 210 may be realized (e.g., compiled together) as one executable software program (e.g., software application or simply referred to as an “app”), which for example may be stored in the

memory 202 and executable by the at least one processor 204 to perform the functions/operations as described herein according to various embodiments.

[0036] In various embodiments, the system 200 corresponds to the method 100a as described hereinbefore with reference to FIG. 1A, therefore, various functions or operations configured to be performed by the at least one processor 204 may correspond to various steps of the method 100a described hereinbefore according to various embodiments, and thus need not be repeated with respect to the system 200 for clarity and conciseness. In other words, various embodiments described herein in context of the methods are analogously valid for the respective systems, and vice versa.

[0037] For example, in various embodiments, the memory 202 may have stored therein the operational data obtaining module 206, the operation status determining module 208, and/or the data labelling module 210, which respectively correspond to various steps of the method 100a as described hereinbefore according to various embodiments, which are executable by the at least one processor 204 to perform the corresponding functions/operations as described herein.

[0038] A computing system, a controller, a microcontroller or any other system providing a processing capability may be provided according to various embodiments in the present disclosure. Such a system may be taken to include one or more processors and one or more computer-readable storage mediums. For example, the system 200 described hereinbefore may include a processor (or controller) 204 and a computer-readable storage medium (or memory) 202 which are for example used in various processing carried out therein as described herein. A memory or computer-readable storage medium used in various embodiments may be a volatile memory, for example a DRAM (Dynamic Random Access Memory) or a non-volatile memory, for example a PROM (Programmable Read Only Memory), an EPROM (Erasable PROM), EEPROM (Electrically Erasable PROM), or a flash memory, e.g., a floating gate memory, a charge trapping memory, an MRAM (Magnetoresistive Random Access Memory) or a PCRAM (Phase Change Random Access Memory).

[0039] In various embodiments, a “circuit” may be understood as any kind of a logic implementing entity, which may be special purpose circuitry or a processor executing software stored in a memory, firmware, or any combination thereof. Thus, in an embodiment, a “circuit” may be a hard-wired logic circuit or a programmable logic circuit such as a programmable processor, e.g., a microprocessor (e.g., a Complex Instruction Set Computer (CISC) processor or a Reduced Instruction Set Computer

(RISC) processor). A “circuit” may also be a processor executing software, e.g., any kind of computer program, e.g., a computer program using a virtual machine code, e.g., Java. Any other kind of implementation of the respective functions which will be described in more detail below may also be understood as a “circuit” in accordance with various alternative embodiments. Similarly, a “module” may be a portion of a system according to various embodiments in the present invention and may encompass a “circuit” as above, or may be understood to be any kind of a logic-implementing entity therefrom.

[0040] Some portions of the present disclosure are explicitly or implicitly presented in terms of algorithms and functional or symbolic representations of operations on data within a computer memory. These algorithmic descriptions and functional or symbolic representations are the means used by those skilled in the data processing arts to convey most effectively the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities, such as electrical, magnetic or optical signals capable of being stored, transferred, combined, compared, and otherwise manipulated.

[0041] Unless specifically stated otherwise, and as apparent from the following, it will be appreciated that throughout the present specification, discussions utilizing terms such as “obtaining”, “determining”, “labelling”, or the like, refer to the actions and processes of a computer system, or similar electronic device, that manipulates and transforms data represented as physical quantities within the computer system into other data similarly represented as physical quantities within the computer system or other information storage, transmission or display devices.

[0042] The present specification also discloses a system (e.g., which may also be embodied as a device or an apparatus), such as the system 200, for performing the operations/functions of the methods described herein. Such a system may be specially constructed for the required purposes, or may comprise a general purpose computer or other device selectively activated or reconfigured by a computer program stored in the computer. The algorithms presented herein are not inherently related to any particular computer or other apparatus. Various general-purpose machines may be used with computer programs in accordance with the teachings herein. Alternatively, the construction of more specialized apparatus to perform the required method steps may be appropriate.

[0043] In addition, the present specification also at least implicitly discloses a computer program or software/functional module, in that it would be apparent to the person skilled in the art that the individual steps of the methods described herein may be put into effect by computer code. The computer program is not intended to be limited to any particular programming language and implementation thereof. It will be appreciated that a variety of programming languages and coding thereof may be used to implement the teachings of the disclosure contained herein. Moreover, the computer program is not intended to be limited to any particular control flow. There are many other variants of the computer program, which can use different control flows without departing from the spirit or scope of the invention. It will be appreciated by a person skilled in the art that various modules described herein (e.g., the operational data obtaining module 206, the operation status determining module 208, and/or the data labelling module 210) may be software module(s) realized by computer program(s) or set(s) of instructions executable by a computer processor to perform the required functions, or may be hardware module(s) being functional hardware unit(s) designed to perform the required functions. It will also be appreciated that a combination of hardware and software modules may be implemented.

[0044] Furthermore, one or more of the steps of a computer program/module or method described herein may be performed in parallel rather than sequentially. Such a computer program may be stored on any computer readable medium. The computer readable medium may include storage devices such as magnetic or optical disks, memory chips, or other storage devices suitable for interfacing with a general purpose computer. The computer program when loaded and executed on such a general-purpose computer effectively results in an apparatus that implements the steps of the methods described herein.

[0045] In various embodiments, there is provided a computer program product, embodied in one or more computer-readable storage mediums (non-transitory computer-readable storage medium), comprising instructions (e.g., the operational data obtaining module 206, the operation status determining module 208, and/or the data labelling module 210) executable by one or more computer processors to perform a method 100a of fault detection and diagnostic for a building management system as described hereinbefore with reference to FIG. 1A. Accordingly, various computer programs or modules described herein may be stored in a computer program product receivable by a system therein, such as the system 200 as shown in FIG. 2, for execution

by at least one processor 204 of the system 200 to perform the required or desired functions.

[0046] The software or functional modules described herein may also be implemented as hardware modules. More particularly, in the hardware sense, a module is a functional hardware unit designed for use with other components or modules. For example, a module may be implemented using discrete electronic components, or it can form a portion of an entire electronic circuit such as an Application Specific Integrated Circuit (ASIC). Numerous other possibilities exist. Those skilled in the art will appreciate that the software or functional module(s) described herein can also be implemented as a combination of hardware and software modules.

[0047] In various embodiments, the system 200 may be realized by any computer system (e.g., desktop or portable computer system) including at least one processor and a memory, such as a computer system 300 as schematically shown in FIG. 3 as an example only and without limitation. Various methods/steps or functional modules (e.g., the operational data obtaining module 206, the operation status determining module 208, and/or the data labelling module 210) may be implemented as software, such as a computer program being executed within the computer system 300, and instructing the computer system 300 (in particular, one or more processors therein) to conduct the methods/functions of various embodiments described herein. The computer system 300 may comprise a computer module 302, input modules, such as a keyboard 304 and a mouse 306, and a plurality of output devices such as a display 308, and a printer 310. The computer module 302 may be connected to a computer network 312 via a suitable transceiver device 314, to enable access to e.g., the Internet or other network systems such as Local Area Network (LAN) or Wide Area Network (WAN). The computer module 302 in the example may include a processor 318 for executing various instructions, a Random Access Memory (RAM) 320 and a Read Only Memory (ROM) 322. The computer module 302 may also include a number of Input/Output (I/O) interfaces, for example I/O interface 324 to the display 308, and I/O interface 326 to the keyboard 304. The components of the computer module 302 typically communicate via an interconnected bus 328 and in a manner known to the person skilled in the relevant art.

[0048] It will be appreciated by a person skilled in the art that the terminology used herein is for the purpose of describing various embodiments only and is not intended to be limiting of the present invention. As used herein, the singular forms "a", "an" and

"the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises" and/or "comprising," when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[0049] In order that the present invention may be readily understood and put into practical effect, various example embodiments of the present invention will be described hereinafter by way of examples only and not limitations. It will be appreciated by a person skilled in the art that the present invention may, however, be embodied in various different forms or configurations and should not be construed as limited to the example embodiments set forth hereinafter. Rather, these example embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the present invention to those skilled in the art.

[0050] Various example embodiments relate to fault detection and diagnostic (FDD) for a building management system. For the sake of simplicity and clarity and unless stated otherwise, various example embodiments will hereinafter be described with respect to a sub-system of the building management system, the sub-system being a chiller system. However, it will be appreciated by a person skilled in the art that the present invention is not limited to a chiller system and may be or include any other sub-systems associated to a building of the building management system, as long as it is operational and susceptible to faults.

[0051] Data-driven fault detection and diagnostics (FDD) of chiller system can be implemented by using a model-based method or an expert-based method. Each of method has its own benefits and drawbacks. Model-based method is able to recognize pattern of data thus it can detect chiller fault once it is given chiller operational data to a trained model. However, it is difficult for a user(s) to know whether the chiller operational data are of normal chiller operation or there may be potential faults, because chiller operational data is usually without label. Therefore, the user(s) needs to do data labelling to classify chiller data into normal operation data or different groups of fault operation data. This requires a lot of effort as it is very time consuming to label the chiller operational data if it is performed manually. Another method is expert-based method. Expert-based method is easier and simpler to implement by building up a rules collection, i.e. knowledge library, based on industry best practice and manufacturer

specification. It will then compare the chiller operational data with the knowledge library to decide whether the chiller system is in normal or fault condition. However, one main drawback is since the knowledge library is usually created based on default chiller operation environment, it might not be able to capture the actual operation environment of the chiller system.

[0052] Various example embodiments integrate model-based method and expert-based method to perform FDD which may be beneficial to complement each other. Through the integration of both expert-based method and model-based method, it can minimize resources utilization.

[0053] Various example embodiments provide a FDD method using a hybrid of expert-based method and model-based method that automatically labels the chiller operation data for model training and validation, automatically updates a knowledge library (e.g., corresponding to the “fault condition repository” as described hereinbefore according to various embodiments) based on actual real chiller operation environment, and minimize resources utilization. Accordingly, various example embodiments disclose the hybrid method and system for FDD that include features to automatically label chiller operational data, automatically updates rules, automatically create new rules and minimizes resources utilization, and accordingly providing a more seamless and efficient FDD system.

[0054] Various example embodiments utilize the knowledge library and automatically labels the chiller operation data into different groups of chiller operation data including, normal and faulty chiller operation data (e.g., corresponding to the “operation status being a normal operation status or a plurality of types of fault operation statuses” as described hereinbefore according to various embodiments). The faulty chiller operation data may be categorized according to different fault types (e.g., different types of fault operation statuses). In various example embodiments, the fault detection and diagnostic system may determine the fault type related to abnormal chiller key performance indicators (KPIs), and execute the FDD process for the corresponding fault type.

[0055] Various example embodiments include a process to validate data labelling process using the FDD results. In the validation process, new validated labelled data are added to existing validated labelled data to be used to train a fault model. If the fault model already existed, the additional validated labelled data may be used to retrain this existing sub-model, hence the model-based method performance can be improved.

Various example embodiments may further calculate the matching score between the model-based method and expert-based method based on the FDD results and decide whether to update the expert-based method. As such, various example embodiments advantageously improve both the model-based method and expert-based method (to
5 self-improve) as time goes by, based on the chiller operational data obtained under actual operation condition.

[0056] The process to update the knowledge library for the expert-based method may be mainly focused on rules (corresponding to the “plurality of fault conditions” as described hereinbefore according to various embodiments) update, e.g. adjusting
10 threshold values (corresponding to the “threshold parameters” as described hereinbefore according to various embodiments) of certain rules. The adjusting of threshold is conducted by optimizing the threshold value with the objective to get the maximum matching score between the model-based method and expert-based method. In addition, various example embodiments may automatically create new rule
15 whenever an unknown fault type is found (or determined). The creation of new rule starts with finding abnormal chiller KPIs during the process of benchmarking all chiller data points in relation to chiller parameters with normal chiller KPIs (corresponding to “determining at least one violated system parameter of the plurality of system parameters corresponding to at least one threshold parameter respectively of the
20 threshold parameters being violated based on the operational data” as described hereinbefore according to various embodiments). Referring to the knowledge library, the system is able to conclude if the abnormal chiller KPIs do not belong to any fault type and hence triggers the system to create new rules. The new rules are created based on the correlation of components of abnormal chiller KPIs (e.g., corresponding to “the
25 new fault condition is created based on the at least one violated system parameter and one or more correlated system parameters correlating with the at least one violated system parameter” as described hereinbefore according to various embodiments).

[0057] Through the integration of both expert-based method and model-based method, various example embodiments enable a chiller FDD system to start with only
30 a few rules to cover significant chiller faults, and automatically creates new rule together with its associated fault model only when needed. With few rules in the knowledge library and their associated fault models various example embodiments advantageously minimize computation resources to perform data labeling, model training and FDD process as compared to traditional FDD implementation. In addition,

various example embodiments automatically re-trains only necessary models needed to be re-trained with the validation process and updates rules by comparing the matching score between expert-based method and model-based method results.

5 [0058] The method and system for fault detection and diagnostic for a building management system will now be described below with respect to a chiller system according to various example embodiments.

[0059] FIG. 4A depicts a schematic diagram of a building management system (BMS) 400 including an FDD system and a chiller system, according to various example embodiments. The system 400 includes a chiller system 410, a fault detection and diagnostic (FDD) system 420, a building energy management system (BEMS) 430, and a communication network 440. The chiller system 410 includes the chiller components, i.e., compressor, condenser, evaporator, etc., with sensors installed to capture the chiller system key performance indicators (KPIs). BEMS 430 is a system to monitor and control the mechanical and electrical equipments of a building that includes, for example, the chiller system, a power system, a lighting system, a water system, a gas system, a fire system, and a security system. The input of the FDD system 420 may come directly from the chiller system 410 and/or from the BEMS 430. In some systems, the BEMS 430 may not be required in order to implement the FDD system 420. The communication network 440 may connect all system components in the overall chiller FDD system 400 and it may support various connectivity, such as Ethernet, Wi-Fi, and/or Zigbee with various protocols, such as BACnet, TCP/UDP, CAN.

[0060] FIG. 4B illustrates a schematic illustrating the components in a chiller system 410. There are three loops in the chiller system 410 as mediums for transferring heat, i.e. condenser water loop 452, refrigerant loop 454 and chilled water loop 456. Two heat exchangers i.e., a condenser 460 and an evaporator 462 exist in the chiller system 410. The evaporator 462 generates chilled water to be distributed to the building 464 whereas the condenser 460 transfers the heat in condenser water which is exhausted to the environment through a cooling tower 458. Chilled water pump 466 and condenser water pump 468 control the flow rate of chilled water flow and condenser water flow, respectively. The compressor 470 is a component to create the refrigerant vapor to be in high-temperature and high-pressure phase. The expansion valve 472 is a component to create the refrigerant to be in low-temperature and low-pressure phase. The healthiness of every component in the chiller system 410 may be important in order to

provide cooling (cool air) to the building 464. One fault in one component could affect the overall performance of the chiller system 410, and subsequently affect the overall building operating cost.

[0061] Chiller faults in the building could reduce the thermal comfort, reduce the energy efficiency, and increase the operating cost. In order to solve all issues caused by the chiller faults, fault detection and diagnostic (FDD) should be implemented. To perform a chiller FDD, the system needs data collected (e.g., corresponding to “obtaining the operational data” as described hereinbefore according to various embodiments) from the sensors installed in the chiller system. FIG. 4C shows a schematic diagram of sensors installed in each component of the chiller system 410. For example, the chiller system 410 may include evaporator sensors 482 which may be or include sensors to measure water entering temperature, water leaving temperature, approach temperature, refrigerant temperature, etc.; condenser sensors 484 which may be or include sensors to measure water entering temperature, water leaving temperature, approach temperature, refrigerant temperature, etc., condenser water pump sensors 486 which may be or include sensors to measure motor speed, water flow, etc.; compressor sensors 488 which may be or include sensors to measure discharge temperature, motor winding temperature, oil reclaim output, etc.; expansion valve sensors 490 which may be or include sensors to measure refrigerant temperature, refrigerant pressure, etc.; chilled water pump sensors 492 which may be or include sensors to measure motor speed, water flow, etc.; electrical sensors 494 which may be or include sensors to measure average line current, line power, line voltage, etc.; and other sensors 496 including sensors to measure chiller control mode, chiller run status, chiller relative humidity, etc. All collected chiller operational data are processed and stored in storage 522 (illustrated in FIG. 5A) through the communication network 440.

[0062] FIG. 5A shows a block diagram illustrating modules within the FDD system 420. The FDD system 420 includes a chiller relationship map creation module 502, a normal chiller KPIs tree creation module 504, a knowledge library creation module 506, a FDD process module 508, a new rule creation module 510, a data labelling module 512, a model training module 514, a validation module 516, a processor 518, a memory 520 and the storage 522. The processor 518 may be the main component of the FDD system that processes data input to generate output of every module. The memory 520 may be a temporary storage component for data that are being processed.

The storage 522 may be the main component to store the input and output data of the FDD system 420. Other modules are explained in detail in later paragraphs.

[0063] FIG. 5B shows a block diagram illustrating components of the storage 522 of the FDD system 420. The storage 522 may be utilized to store chiller data (e.g., corresponding to the operational data described hereinbefore) 532, chiller relationship map 534, normal chiller KPIs tree 536, knowledge library 538, labelled chiller data 540, fault models 542, FDD report 544 and other tables 546. Other tables 546 may include matching score table, rule-model creation status table and rule-model pairing table. Details on the tables are elaborated in later paragraphs.

10 [0064] FIG. 6 shows an example flow diagram illustrating the exemplary steps of chiller relationship map creation in the chiller relationship map creation module 502. At 602, all chiller KPIs/parameters of the chiller system may be gathered or obtained from chiller manufacturer specification sheet. At 604, a chiller relationship map (e.g., corresponding to the “correlation information of the plurality of system parameters”
15 described hereinbefore according to various embodiments) may be created based on the available chiller parameters with reference to a prebuilt chiller relationship map. The prebuilt chiller relationship map may be created according to the best practice chiller operation knowledge that have been verified and validated by the chiller system expert (e.g., corresponding to the “expert knowledge information” as described hereinbefore
20 according to various embodiments). At 606, the chiller relationship map may be stored in storage. The chiller relationship map, that is created when the chiller system is set up, is the knowledge of relationship of chiller components and their respective KPIs/parameters. The chiller relationship map indicates which parameters are related to or belong to a component, and relatedness of parameters of the same component as well as of different components of the chiller. The chiller KPI/parameter data are not
25 only obtained from the sensors installed in the chiller system, but it can also be derived from the existing chiller KPIs data, e.g. chilled water delta temperature can be derived by subtracting chilled water return temperature with chilled water supply temperature, chiller efficiency can be derived by dividing electricity consumed with generated
30 cooling power, etc.

[0065] FIG. 7 shows a simplified example of the chiller relationship map. It shows the relationship of parameters in one component or parameters of different components (e.g., indicate, for each system parameter of the plurality of system parameters, one or more correlated system parameters correlating with the system parameter). The dash-

line arrows represent parameter relationships of one component, whereas the dotted-line arrows represent parameter relationships of different components.

[0066] FIG. 8 shows an example of a more comprehensive chiller relationship map of two chiller components, i.e. condenser and compressor.

5 [0067] FIG. 9 shows an example flow diagram illustrating the exemplary steps of normal chiller KPIs tree creation in the normal chiller KPIs tree creation module 504. At 902, a tree of all chiller KPIs may be constructed referring to the chiller relationship map. At 904, all normal data ranges (e.g., threshold parameters) of all chiller KPIs may be assigned in the tree. At 906, the created normal chiller KPIs tree may be stored in
10 storage. The normal data ranges can be configured based on product specification sheet or can be derived from the collected chiller operational data under normal operation (corresponding to the normal operation status). During chiller normal operation, the normal range of a parameter can be obtained, e.g., from the minimum to the maximum values of the respective chiller data. Every time there is any change in the product
15 specification or user setting on the chiller system, the FDD system updates the normal chiller KPI tree accordingly. Some examples of chiller KPIs with their normal data ranges are as follow: chilled water return temperature normal data range is 10°C – 14°C, chilled water supply temperature normal data range is 5°C – 7°C, etc.

[0068] FIG. 10 shows an example of the normal chiller KPIs tree. It shows how
20 chiller components are associated with their respective KPIs/parameters as well as their associated normal data ranges (e.g., corresponding to the “each system parameter of the plurality of system parameters comprises a threshold parameter associated to the system parameter defined in the fault condition repository” described hereinbefore according to various embodiments).

25 [0069] FIG. 11 shows an example flow diagram illustrating the exemplary steps of knowledge library creation in the knowledge library creation module 506. At 1102, an initial fault type list may be created referring to the chiller relationship map and the normal chiller KPIs tree. At 1104, a fault rule library (corresponding to the plurality of fault conditions as described hereinbefore according to various embodiments)
30 comprised of fault types with their respective rules (e.g., each fault condition corresponds to a type of fault operation status) may be created. At 1106, after the knowledge library (corresponding to the fault condition repository as described hereinbefore according to various embodiments), that contains chiller heuristic

knowledge including the fault type list and the fault rule library, is created, the knowledge library is stored to storage.

[0070] FIG. 12 shows an example representation on how by using the chiller relationship map and the normal chiller KPIs tree, the fault type list (e.g., corresponding to the “fault type parameter association information of the plurality of types of fault operation statuses” described hereinbefore according to various embodiments) of the knowledge library can be created. In other words, the fault type parameter association information of the plurality of types of fault operation statuses may be created based on the chiller relationship map (correlation information of the plurality of system parameters) and the normal chiller KPIs tree (threshold parameters associated to the system parameters, respectively). For example, parameter 1 with normal range 1 1202 and parameter 2 with normal range 2 1204 that belong to component condenser 1206 are correlated and thus it can be used to detect a condenser fault 1208, e.g. condenser fouling.

[0071] FIG. 13 shows another example representation on how by using the chiller relationship map and the normal chiller KPIs tree, the fault type list of the knowledge library can be created when a fault type includes, or is defined by, parameters that belong to different components, i.e. refrigerant fault 1314. The refrigerant fault 1314, e.g., refrigerant leak fault type, can be detected using parameter 8 with normal range 8 1302, parameter 4 with normal range 4 1304, parameter 5 with normal range 5 1306 and parameter 6 with normal range 6 1308. Parameter 8 belongs to compressor 1310, while parameters 4, 5 and 6 belong to evaporator 1312. With the same procedures other fault types are able to be created in the knowledge library. Afterward, the fault type list and the fault rule library of all fault types are recorded in the knowledge library.

[0072] FIG. 14 shows an example of the fault type list in the knowledge library. The fault type list shows different types of fault (corresponding to the plurality of types of fault operation statuses as described hereinbefore according to various embodiments) that can be detected (or determined) and diagnosed by the FDD system 420. It is comprised of sequential number of fault types 1402, fault ID number 1404, fault type name 1406, fault description 1408, required data points with respect to parameters 1410 and parameter normal range 1412. For example, for each type of fault operation status, the fault type parameter association information indicates one or more system parameters of the plurality of system parameters being associated to the type of fault operation status. For example, for one type of fault operation status such as condenser

fouling, the fault type parameter association information may indicate a set of associated system parameters such as parameter 1 and parameter 2 for the condenser fouling type of fault operation status. The fault type parameter association information may further indicate the respective threshold parameter which is associated with each system parameter (e.g., range 1 is associated to parameter 1, range 2 is associated to parameter 2).

[0073] FIG. 15 shows an example of the data structure of the fault rule library in the knowledge library. The fault rule library includes sequential number of fault types 1502, fault ID number 1504, fault type 1506 and fault rule (e.g., corresponding to the fault condition as described hereinbefore according to various embodiments) 1508. The fault rule 1508 may be a collection of rules containing chiller expert knowledge that detects the occurrence of the particular fault (e.g., detecting or determining a type of fault operation status based on the plurality of fault conditions) by applying a collection of IF-THEN-ELSE conditional statements algorithm that refers to the chiller relationship map and the normal chiller KPIs tree.

[0074] FIG. 16 shows a flow diagram illustrating the exemplary steps of chiller FDD process of the FDD process module 508. At 1602, chiller operational data may be obtained from storage.

[0075] FIG. 17 shows an example of the data structure of the chiller operational data. It includes sequential number of chiller data 1702, data timestamp 1704, data points with respect to chiller parameter 1 1706, chiller parameter 2 1708, chiller parameter n-1 1710 and chiller parameter n 1712.

[0076] Referring back to FIG. 16, at 1604, the system then finds abnormal chiller KPIs utilizing normal chiller KPIs tree. At 1606, it is determined whether there is any abnormal chiller KPI found. If there is no abnormal chiller KPI the system goes to step 1614. Otherwise, at 1608, it detects all fault types related to the abnormal chiller KPIs referring to the knowledge library. At 1610, it then checks whether an unknown fault is found. If an unknown fault is found, at 1612, it executes new rule creation module. Otherwise, at 1614, it executes data labelling module. At 1616, it checks the detected (or determined) fault types with FDD report to find resolved faults.

[0077] FIG. 18A shows an example of the data structure of the FDD report. It contains sequential number of occurred faults 1802, fault ID number 1804, fault type name 1806, fault type version 1808, first occurrence timestamp of the fault 1810, expected resolving period 1812, fault flag 1814, resolved status 1816 and resolved

timestamp 1818. The first occurrence timestamp 1810 is the timestamp of the fault when it is first occurred. The expected resolving period 1812 is a time period given to a user to take some actions mitigating the occurred faults. The fault flag 1814 is an indicator of fault detection methods. When both methods, i.e. expert-based method and model-based method detect the fault, it is flagged as “Confirmed”, otherwise it is flagged as “Potential”. The resolved status 1816 is an indicator whether the fault has been resolved or not. The status “YES” indicates that the fault has been resolved, whereas “NO” indicates the fault still exists. The resolved timestamp 1818 is the timestamp when the fault is resolved.

- 10 **[0078]** FIG. 18B shows another example of the data structure of the FDD report. It shows a different way on how the results of the FDD process may be presented. It may contain sequential number of occurred faults 1820, fault type name 1822, fault description 1824, number of occurrences of the fault 1826, last occurrence timestamp 1828, fault flag 1830, fault priority 1832, possible causes of the occurred fault 1834, abnormal data points detected during fault 1836, table view button 1838, graph view button 1840, recommendations to mitigate the fault 1842 and resolved status 1844. The last occurrence timestamp 1828 is the timestamp of the fault when it last occurred. The fault flag 1830 is an indicator of fault detection methods. When both methods, i.e. expert-based method and model-based method detect the fault, it is flagged as “Confirmed”, otherwise it is flagged as “Potential”. The fault priority 1832 represents the indicator how crucial is the fault to be mitigated. The fault priority may be “High”, “Medium” or “Low”. The “High” priority fault is required to be mitigated as soon as possible. The “Medium” priority fault can be taken into action after the “High” priority fault has been processed to be mitigated. The “Low” priority has the least impact to the chiller system, so it can be taken into action after the “High” and “Medium” priority faults have been processed. The possible causes 1834 describes some possibilities of what have caused the fault to occur. The abnormal data points 1836 shows all data points that contribute to the occurrence of the fault. The table view button 1838, when it is clicked or selected, it will show all the occurrences of a fault type in table form, whereas the graph view button 1840, when it is clicked or selected, it will show all occurrences of a fault type in graph form. The recommendations 1842 describes some possible recommended actions to mitigate the fault. The status 1844 is an indicator whether the fault has been resolved or not. The status “resolved” indicates that the fault has been resolved, whereas “remain” indicates the fault still exists.

- [0079] Referring back to FIG. 16, at 1618, the system checks whether there is any resolved fault. The resolved faults can be found by comparing the current detected fault types with all fault types existing in the FDD report. If there is a fault in the FDD report missing in the current detected fault types, it can be concluded that the fault has been resolved. If there is a resolved fault, at 1620, it executes the validation module and at 1622, it checks whether there is any remaining fault. Otherwise, at 1622, it checks whether there is any remaining fault. If there is any remaining fault, at 1624, it then executes fault handling process. Otherwise, it repeats the procedure at 1602. The FDD process in the FDD system always keeps running to perform chiller FDD continuously.
- 10 [0080] FIG. 19 shows a flow diagram illustrating the exemplary steps of finding abnormal chiller KPIs utilizing normal chiller KPIs 1604. According to various example embodiments, at 1902, it starts with a first parameter of chiller normal KPIs which is chiller efficiency. At 1904, it will check whether the efficiency is within the range of range 0, e.g., 0.7 – 1.2 kW/RT. For example, the range 0 refers to a variable that stores the normal range to a certain parameter. In this case, range 0 refers to the normal range of the chiller efficiency. If the efficiency is within the range of range 0, the chiller is in normal operation (e.g., normal operation status may be determined) and the process stops and goes back to main flow diagram. Otherwise, at 1906, it flags the chiller efficiency as abnormal chiller KPI (e.g., corresponding to the “at least one violated system parameter of the plurality of system parameters corresponding to at least one threshold parameter respectively of the threshold parameters being violated based on the operational data” as described hereinbefore according to various embodiments) and continues to check the next chiller KPIs. For example, chiller efficiency may be the main parameter of the chiller system that can be observed to see whether the chiller system is working normally or not. Accordingly, it may be the first parameter to be checked, in a non-limiting example. When the chiller efficiency is within the normal range the FDD method is not executed, and it will wait until the next chiller data arrive in the system. When chiller efficiency exceeds the normal range, the FDD method is executed starting from flagging which chiller parameters are abnormal, and so forth.
- 20
- 25
- 30
- [0081] At 1908, it checks parameter 1. At 1910, it checks whether parameter 1 is within the range of range 1. If it is not, at 1912, it flags parameter 1 as abnormal KPI. If parameter 1 is within the range of range 1, it then continues to next other parameters. The same procedures are carried out for parameter 2 at 1914, parameter 3 at 1920, and

other chiller KPIs at 1926. At 1932, it then stores all flagged abnormal chiller KPIs to memory.

[0082] FIG. 20 shows a flowchart illustrating the new rule creation process in the new rule creation module 510. In the case there is no fault type found related to the affected abnormal KPIs in the FDD process, the system executes the new rule creation process. In various example embodiments, the new rule creation process may include updating an existing rule (the existing rule is modified and it becomes a new rule by modifying the existing rule, e.g. addition of parameter in the existing rule) and creating a new rule that did not exist in the knowledge library. At 2002, it picks an abnormal chiller KPI. At 2004, it then obtains the correlation between the abnormal chiller KPI and the existing fault types. The existing fault types refer to all fault types with the corresponding fault rules and corresponding chiller parameters that have been stored in the database system. The correlation between the abnormal chiller KPI and the existing fault types may be obtained with the assistance of the chiller relationship map in the knowledge library. At 2006, it checks whether the abnormal chiller KPI is correlated with the existing fault types. If the abnormal chillers KPI correlates to the existing fault types, it adds the chiller KPI to existing fault types at 2008 and goes to step 2012. Otherwise, at 2010, it creates a new rule comprised of the abnormal chiller KPI and its associated chiller KPIs. Said differently, the new rule is created based on the parameter associated with the abnormal KPI or value (e.g., violated system parameter), as well as one or more correlated parameters (e.g., correlating with the violated system parameter) based on the chiller relationship map (corresponding to “the new fault condition is created based on the at least one violated system parameter and one or more correlated system parameters correlating with the at least one violated system parameter” as described hereinbefore according to various embodiments). After that, it updates the fault type list and the fault rule in the knowledge library at 2012, updates rule-model creation status table at 2014 and updates rule-model pairing table at 2016. At 2018, it then checks whether all abnormal chiller KPIs have been checked. If it has checked all abnormal chiller KPIs the process stops. Otherwise, it goes back to step 2002.

[0083] In relation to modifying an existing rule, for example, in the case a Condenser Fouling fault type is the existing fault type with the fault rule (corresponding to the “fault condition” as described): Condenser Fouling is detected if Refrigerant Temp. is bigger than x value and Approach Temperature is bigger than y value, when a new abnormal chiller parameter that is related to the existing Condenser Fouling fault

type, i.e. Refrigerant Pressure, a new rule may be created by modifying the existing fault rule. It becomes Condenser Fouling is detected if Refrigerant Temp. is bigger than x value and Approach Temperature is bigger than y value and Refrigerant Pressure is bigger than z value.

- 5 **[0084]** FIG. 21 shows an example of the data structure of a rule-model creation status table. The rule-model creation status table contains sequential number of faults 2102, fault ID number 2104, fault type name 2106, rule creation status 2108, model creation status 2110. The rule creation status 2108 defines the fault whether it is an existing fault type or a new fault type. For existing fault, it flags as “Update Existing”.
10 As for the new fault, it is flagged as “New”. The model creation status 2110 defines the current status of the model creation process. For models that are being trained, it flags as “Under Retraining” if it is of existing fault model, “Under Training” if it is of new fault model, whereas the models that have been trained is flagged as “Completed”. For model that is not able to be trained because it has no sufficient labelled data to perform
15 model training, it is flagged as “Under Data Labelling”.

- [0085]** FIG. 22 shows an example of the data structure of a rule-model pairing table. The rule-model pairing table contains sequential number of faults 2202, fault id number 2204, fault type name 2206, rule version 2208, model version 2210, pairing version 2212, and last updated timestamp 2214. The rule version 2208 defines the version
20 number of fault rule and it starts with 1. Every time the system updates the rule, the version number increases. The model version 2210 defines the version number of fault model and it starts with 0. The model version 0 means the model has not been created for this fault type. Every time the system retrains the model, the version number increases. The pairing version 2212 defines the version of the pairing of rule version
25 2208 and model version 2210. For example, each fault rule has an associated fault model, the method stores the rule-model pair. Any rule or model can be updated to be a newer version, hence a table may log the pairing including the version of the rule-model pair. The last updated timestamp 2214 defines the last timestamp of pairing version 2212 change.

- 30 **[0086]** FIG. 23 shows an example of the new rule creation process by using the chiller relationship map. More particularly, the new rule creation process may be in relation to modifying an existing rule, i.e., updating the existing rule. For example, condenser fouling fault 2310 is a known or existing fault type in the condenser component 2308 that can be detected with KPIs (or parameters) of refrigerant

temperature 2302 and condenser approach temperature 2304. During the FDD process, the system finds out another abnormal KPI, i.e., the refrigerant pressure parameter 2306 that is correlated to parameters refrigerant temperature 2302 and condenser approach temperature 2304 (dotted lines in the chiller relationship map) of the fault type condenser fouling fault 2310. Therefore, the system updates the existing fault type condenser fouling fault 2310 by adding the KPI (parameter refrigerant pressure) 2306 (e.g., corresponding to the “associating, for each violated system parameter of the at least one violated system parameter, the violated system parameter to said existing type of fault operation status to produce an updated set of system parameters associated to the existing type of fault operation status” as described hereinbefore according to various embodiments) in the fault type list and the fault rule in the knowledge library. Accordingly, for the next execution to detect Condenser Fouling, chiller parameters refrigerant temperature 2302, condenser approach temperature 2304 and refrigerant pressure 2306 need to be considered.

[0087] FIG. 24 shows another example of the new rule creation process by the using chiller relationship map. In this case, the system also finds an abnormal chiller KPI, i.e. oil reclaim parameter 2402 in compressor component 2406. However, this abnormal chiller KPI does not correlate to any known fault types in the knowledge library. Therefore, the system creates a new rule, e.g. excess oil fault rule 2408, that includes the oil reclaim KPI or parameter 2402 and other associated chiller KPI or parameter, i.e. oil pressure parameter 2014 (e.g., corresponding to the “new fault condition is created based on the at least one violated system parameter and one or more correlated system parameters correlating with the at least one violated system parameter” as described hereinbefore according to various embodiments) and updates the fault type list and the new fault rule in the knowledge library.

[0088] FIG. 25 shows a flowchart illustrating a validation process in the validation module 516. The validation may be with respect to the validation of the data labelling process which is an iterative and continuous process. In the validation process, at 2502, the system picks a resolved fault type to be processed. At 2504, it then updates the FDD report by changing the resolved status of the resolved fault types to “YES”. At 2506, it checks whether the associated fault model of this fault type is available. If it is not available, at 2516, it validates the labelled chiller data related to this fault type, then it goes to step 2514. Otherwise, at 2508, it retrieves the existing validated labelled data. At 2510, it combines the new validated labelled chiller data to the existing validated

labelled chiller data. For example, the labelled data is automatically validated when it is determined that the associated fault model of the fault type is available, and the new validated labelled chiller data and the existing validated labelled chiller data may be combined. The validation with the FDD results may be an iterative and continuous process. The validation with the FDD results is performed by comparing the current results of the FDD method with the previous results. If the current results do not show the same occurred fault type because the fault type has been mitigated (assuming that the building facility has performed the recommended actions), the FDD method updates the resolved status of the fault type as “YES”. This event shows that the labelled chiller data are valid labelled chiller data since by executing the recommended actions the fault is mitigated. In other words, if the fault model of a certain fault type already exists, the new validated labelled data can be added to the existing validated labelled data for retraining purpose. The addition of training data to train the machine learning model is useful to improve the performance of the machine learning model. On the other hand, if the fault model of a certain fault type does not exist, the labelled data is validated by updating the “validation status” to be “1”, as illustrated in a table shown in FIG. 28B.

[0089] At 2512, it executes the model training module to retrain the model. At 2514, it then checks whether all fault types have been checked. If it has not checked all fault types, it goes back to step 2502 to process the next and repeat the same procedure. Otherwise, it stops and goes back to main flow diagram.

[0090] FIG. 26 is a flowchart illustrating the fault handling process 1624. At 2602, a fault type may be picked from the detected (or determined) fault types to be further processed. At 2604, the system then checks whether the fault exists in the FDD report. If it exists, at 2606, it checks whether the fault first occurrence time exceeds the expected resolving period. If the fault occurrence time exceeds the expected resolving time, at 2608, it reminds the user to take action to mitigate the fault and proceeds to step 2616. Otherwise, at 2610, the system checks whether the associated fault model of this fault type is available. For example, a respective fault model may be associated to a corresponding fault condition of the expert-based method that is also associated to a type of the fault operation status. If it is not available, at 2612, it executes model creation process and proceeds to step 2616. Otherwise, at 2614, the system executes fault predicting model (corresponding to the “invoking, for the type of fault operation status determined, a fault model associated to the type of fault operation status determined for detecting a fault corresponding to the type of fault operation status” as

described hereinbefore according to various embodiments) and proceeds to step 2616. At 2616, if the system has processed all fault types, the process stops and goes back to main flow diagram. Otherwise, it proceeds to the next fault type to be processed at 2602 and repeats the same procedure.

5 **[0091]** FIG. 27 shows an example flow diagram illustrating the model creation process 2612. At 2702, it checks the status of the model to determine whether it is under training. If it is not, at 2704, it then checks whether the validated labelled data is sufficient to do training. In a non-limiting example, the determination as to whether there is sufficient labelled data is made according to a predetermined number of the
10 validated labelled data, e.g. validated labelled data that equals to two weeks data. Otherwise, the process stops and goes back to the main flow diagram. At 2706, if the validated labelled data is sufficient, it then executes the model training module 514.

[0092] FIG. 28A shows a flow diagram illustrating the exemplary steps of a data labelling process in the data labelling module 512. At 2802, it starts with obtaining
15 detection results from the expert-based method. At 2804, it then labels the chiller operational data with the corresponding normal type or faulty type and stores the labelled chiller data to storage at 2806. The labelling includes any data value with respect to a parameter obtained under a normal or fault operation condition (or status). Said differently, the expert-based method is used to detect or determine whether the
20 chiller operational data is of a normal operation condition or fault operation condition. The expert-based method may be used to identify both normal type with respect to normal chiller operational data and a fault type with respect to abnormal chiller operational data.

[0093] Once it is determined based on the expert-based method whether the chiller
25 operational data is of normal operation condition or fault operation condition, the chiller operational data may be labelled according to the results (e.g., with respect to normal type and fault type, respectively) to be used for the training of the machine learning model. The data labelling process also considers the missing data and outliers to avoid inaccurate data labelling. Data cleaning process, to fill the missing data and correct the
30 outliers, may be carried out before the data labelling process.

[0094] FIG. 28B shows an example table illustrating the labelled data with respect to the chiller operational data. For each data, if it is a normal data, the data is labeled as “0” in the column “label” with remarks as “normal” as shown in data with identifier 1. The abnormal data is labeled as “1”, “2”, etc., with the respective remarks of “fault type

1”, “fault type 2”, etc. For example, the expert-based method detects a fault type 1 for data with identifier 361, hence the column “label” is labeled with “1” with column “remarks” filled with “fault type 1” (or type 1 of fault operation status). The column “validation status” is used to flag the labelled data whether it has been validated or not.

- 5 The value of “1” means the labelled data has been validated, whereas “0” means the labelled data has not been validated. The training of the machine learning model uses the validated labelled data. As illustrated, each fault model is associated to a corresponding fault rule of the expert-based method, the corresponding fault rule is used to identify a fault type (corresponding to a type of fault operation status) and label
10 operational data with respect to that fault type to train the corresponding fault model.

- [0095] FIG. 29 shows an example flow diagram illustrating the exemplary steps of a model training process in the model training module 514. At 2902, the validated labelled chiller data may be obtained. At 2904, the fault prediction model (or machine learning model) may be trained and at 2906, the trained model may be validated.
15 Validating the model may include validation of the result of the model compared with the validated labelled data. For example, in machine learning, to train the model, the labelled data to be trained needs to be prepared. This labelled data may be separated into three groups, i.e. training, validation and testing data sets. The training data set will be used to train the machine learning model, the validation data set to validate the
20 created model, whereas the testing data set will be used to test. The validation process may be carried out to tune the parameters of a classifier of the machine learning model.

- [0096] At 2908, the validation result may be checked. The validation of a trained model is considered okay when the accuracy score reached to a predetermined value, e.g. 80% or above. If the validation result is not okay, the process stops, and it goes
25 back to main flow diagram. Otherwise, at 2910, the system stores the model to storage. The system updates the rule-model creation status table at 2912 and updates the rule-model pairing table at 2914.

- [0097] FIG. 30 shows an example flow diagram illustrating the exemplary steps of executing fault predicting model process 2614. At 3002, the fault predicting model may be executed. At 3004, the FDD report may be updated. For example, updating the FDD
30 report may include logging the results of the fault predicting model process whether one or more fault(s) is/are detected by the model-based method. When any fault is detected by the model-based method, the Flag status may be updated, i.e. from Potential to Confirmed (as illustrated in the FDD report in FIG. 18). At 3006, the matching score

table may be updated. Afterward, at 3008, it checks whether the matching score is less than a score threshold. If it is, at 3010, it then updates the knowledge library (corresponding to the “updating the fault condition repository if the match result does not satisfy the predetermined fault condition repository update condition” as described hereinbefore according to various embodiments). During update of the knowledge library, the threshold parameter(s) of one or more rules may be updated according to the results of the optimization, for example, in order to obtain the optimal matching score. The process of updating the knowledge library is described with respect to FIG. 32. Otherwise, the process stops and goes back to main flow diagram. The score threshold to decide whether the knowledge library is required to be updated or not, may be predefined. Accordingly, the result(s) of the machine learning model in detecting a fault is used to update the matching score. The matching score is a score to evaluate the matched results between the expert-based method and machine learning model method in detecting a fault type. The value of the matching score has a predetermined threshold which triggers the process of updating knowledge library 3010. In this manner, the machine learning model is used to perform the updates of rules (corresponding to the “fault conditions” as described hereinbefore according to various embodiments). The fault rule may be adjusted by changing the threshold of each fault rule. Threshold here refers to the normal data range of a chiller parameter (corresponding to the “threshold parameter associated to the system parameter” as described hereinbefore according to various embodiments), whether it is lower bound, upper bound or range value. The process of deciding the new threshold value is done by using the previous chiller data. Using the previous chiller data, the matching score can be evaluated again after threshold adjustment. The threshold that results in the highest matching score is chosen as a new threshold and it is then used to update the fault rule.

[0098] FIG. 31 shows an example of the data structure of the matching score table. The matching score table may include sequential number of occurred faults 3102, fault ID number 3104, fault type name 3106, matching score 3108. The matching score 3108 is the percentage of the result of expert-based method compared with result of model-based method. Every time a fault occurs, the system updates the matching score. Accuracy of 100% shows that the expert-based method always detects the same fault type with the model-based method.

[0099] FIG. 32 shows an example flow diagram illustrating the exemplary steps of updating knowledge library process 3010. At 3202, the fault rule may be adjusted by

optimizing a threshold parameter with the objective to get the maximum matching score between model-based method and expert-based method. The objective refers to getting the maximum matching score between the results of the expert-based method and the machine learning model method. The optimization is performed to find the maximum matching score as the optimization objective subject to the allowable threshold value of each chiller parameter belonged to a corresponding fault rule as follows:

$$\text{maximize } y = f(x_i)$$

subject to:

$$lb_i < x_i < ub_i, \quad (i = 1, 2, 3, \dots, M)$$

where: y = the objective function which is to maximize the matching score value, x_i = the threshold variable of chiller parameter i belonged to a corresponding fault rule, lb_i = the lower bound of the threshold variable of chiller parameter i , ub_i = the upper bound of the threshold variable of chiller parameter i .

[00100] Referring to FIG. 33, for example, when the process of updating of a fault rule is triggered, the process illustrated is executed with the existing chiller operational data to find the combination of threshold $x_1, x_2, \dots, x_n$, so as to maximize the matching score value. The chiller operational data may include the collection of data points with respect to the chiller parameters that are recorded periodically. Accordingly, the example diagram illustrates not only one sample of data but several samples of data that have been collected for a certain period of time, e.g., within two weeks.

[00101] Referring back to FIG. 32, at 3204, the threshold of the related fault rule may be updated in the fault rule library. At 3206, the updated knowledge library may be stored to storage. At 3208, the rule-model creation status table may be updated. At 3210, the rule-model pairing table may be updated.

[00102] Accordingly, various example embodiments provide a hybrid chiller FDD system, which is a more seamless and efficient FDD system that automatically labels chiller operational data, automatically updates rules, automatically creates new rules and minimizes resources utilization.

[00103] As described, the building chiller system is a complex system, i.e. system that comprises several correlated subsystems. In this system, a fault in a component may affect the overall performance of the system and a fault in a critical component might stop the operation of the whole system. Various example embodiments may be used to implement fault detection and diagnostics for building chiller system, either it

is water-cooled chiller or air-cooled chiller, in which the FDD system automatically labels chiller data into different type of faults (e.g. condenser fouling, degradation of chiller efficiency, compressor leaking, etc.), automatically updates rules and minimizes resources utilization so that a more seamless and efficient chiller FDD system can be achieved.

[00104] According to various other example embodiments, the method of fault detection and diagnostic as described may be used for a building microgrid system which includes several components, i.e. solar photovoltaic (PV), battery storage system, diesel generator, gas engine, etc. to provide power to the building. Various other example embodiments may also be used to implement FDD for the building microgrid system, in which the FDD system automatically labels components data into different type of faults (e.g. voltage swelling, voltage sagging, battery state of charge under safety range, etc.), automatically updates rules related to the components in the building microgrid and minimizes resources utilization so that a more seamless and efficient microgrid FDD system can be achieved.

[00105] According to various other example embodiments, the method of fault detection and diagnostic as described may be used for an elevator system which may include a lift car, motor engine, etc., to be used for passenger or cargo vertical movement assistance. Multiple data points with respect to system parameters may be collected to detect lift movement. Various example embodiments may also be used to implement FDD for the elevator system, in which the FDD system automatically labels components data into different type of faults (e.g. load overweight, degradation of motor efficiency, etc.), automatically updates rules related to the components in the elevator system and minimizes resources utilization so that a more seamless and efficient elevator FDD system can be achieved.

[00106] While embodiments of the invention have been particularly shown and described with reference to specific embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the scope of the invention as defined by the appended claims. The scope of the invention is thus indicated by the appended claims and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced.

CLAIMS

What is claimed is:

- 5 1. A method of fault detection and diagnostic for a building management system, using at least one processor, the method comprising:
- obtaining operational data in relation to a plurality of system parameters associated to a plurality of operational components of the building management system;
- determining an operation status among a plurality of types of operation statuses,
- 10 including a normal operation status and a plurality of types of fault operation statuses with respect to the operational data based on a fault condition repository configured based on expert knowledge information, the fault condition repository comprising a plurality of fault conditions in relation to the plurality of system parameters, wherein each fault condition corresponds to a type of fault operation status; and
- 15 labelling the operational data based on the operation status determined to produce labelled data in relation to the operation status determined.
2. The method according to claim 1, wherein:
- said determining an operation status with respect to the operational data based
- 20 on the fault condition repository comprises determining a type of fault operation status; and
- the method further comprises invoking, for the type of fault operation status determined, a fault model associated to the type of fault operation status determined for detecting a fault corresponding to the type of fault operation status.
- 25 3. The method according to claim 2, wherein:
- said determining a type of fault operation status produces a first detection result;
- said invoking, for the type of fault operation status determined, a fault model associated to the type of fault operation status produces a second detection result; and
- 30 the method further comprises:
- matching the first detection result and the second detection result to obtain a match result;
- determining whether the match result satisfy a predetermined fault condition repository update condition; and

updating the fault condition repository if the match result does not satisfy the predetermined fault condition repository update condition.

4. The method according to claim 2 or 3, wherein each system parameter of the plurality of system parameters comprises a threshold parameter associated to the system parameter defined in the fault condition repository, and each fault condition of the plurality of fault conditions is configured based on one or more of the threshold parameters.
5. The method according to claim 4, wherein said determining a type of fault operation status comprises:
- determining at least one violated system parameter of the plurality of system parameters corresponding to at least one threshold parameter respectively of the threshold parameters being violated based on the operational data;
- in response to the at least one violated system parameter determined, determining whether there is an existing type of fault operation status corresponding to the at least one violated system parameter based on the plurality of fault conditions; and
- in response to determining that there is an existing type of fault operation status corresponding to the at least one violated threshold parameter, determining the existing type of fault operation status as the operation status.
6. The method according to claim 5, wherein when there is no existing type of fault operation status corresponding to the at least one violated system parameter, further comprising creating a new fault condition in relation to an unknown type of fault operation status, wherein the new fault condition is created based on the at least one violated system parameter and one or more correlated system parameters correlating with the at least one violated system parameter.
7. The method according to claim 5, wherein the fault condition repository comprises correlation information of the plurality of system parameters, and fault type parameter association information of the plurality of types of fault operation statuses, wherein for each type of fault operation status, the fault type parameter association information indicates one or more system parameters of the plurality of system

parameters being associated to the type of fault operation status, and the method further comprises:

5 determining, for each violated system parameter of the at least one violated system parameter, one or more correlated system parameters with which the at least one violated threshold parameter is correlated based on the correlation information;

determining, for each violated system parameter of the at least one violated system parameter, an existing type of fault operation status to which the one or more correlated system parameters are associated based on the fault type parameter association information; and

10 associating, for each violated system parameter of the at least one violated system parameter, the violated system parameter to said existing type of fault operation status to produce an updated set of system parameters associated to said existing type of fault operation status in the fault condition repository.

15 8. The method according to claim 4, wherein said updating the fault condition repository if the match result does not satisfy the predetermined fault condition repository update condition comprises:

20 updating a fault condition associated to the type of fault operation status determined based on an objective function to obtain a maximum match result between the first detection result and the second detection result, wherein said updating the fault condition comprises adjusting one or more threshold parameters associated to one or more system parameters, respectively, corresponding to the fault condition.

25 9. The method according to claim 1, further comprising performing validation of the labelled data in relation to the operation status determined to produce validated labelled data for training a fault model.

10. The method according to claim 9, further comprising training the fault model based on the validated labelled data.

30 11. The method according to any one of claims 1 to 10, wherein the operational data comprises measured data in relation to the plurality of systems parameters of the plurality of operational components obtained from one or more sensors associated to the operational components.

12. A system for fault detection and diagnostic for a building management system, the system comprising:

a memory; and

5 at least one processor communicatively coupled to the memory and configured to:

obtain operational data in relation to a plurality of system parameters associated to a plurality of operational components of the building management system;

10 determine an operation status among a plurality of types of operation statuses, including a normal operation status and a plurality of types of fault operation statuses with respect to the operational data based on a fault condition repository configured based on expert knowledge information, the fault condition repository comprising a plurality of fault conditions in relation to the plurality of system parameters, wherein each fault condition corresponds to a
15 type of fault operation status; and

label the operational data based on the operation status determined to produce labelled data in relation to the operation status determined.

20 13. The system according to claim 12, wherein:

said determine an operation status with respect to the operational data based on the fault condition repository comprises determine a type of fault operation status; and

the at least one processor is further configured to invoke, for the type of fault operation status determined, a fault model associated to the type of fault operation status
25 determined for detecting a fault corresponding to the type of fault operation status.

14. The system according to claim 13, wherein:

said determine a type of fault operation status produces a first detection result;

said invoke, for the type of fault operation status determined, a fault model

30 associated to the type of fault operation status produces a second detection result; and

the at least one processor is further configured to:

match the first detection result and the second detection result to obtain a match result;

determine whether the match result satisfy a predetermined fault condition repository update condition; and

update the fault condition repository if the match result does not satisfy the predetermined fault condition repository update condition.

5

15. The system according to claim 13 or 14, wherein each system parameter of the plurality of system parameters comprises a threshold parameter associated to the system parameter defined in the fault condition repository, and each fault condition of the plurality of fault conditions is configured based on one or more of the threshold parameters.

10

16. The system according to claim 15, wherein said determine a type of fault operation status comprises:

determine at least one violated system parameter of the plurality of system parameters corresponding to at least one threshold parameter respectively of the threshold parameters being violated based on the operational data;

15

in response to the at least one violated system parameter determined, determine whether there is an existing type of fault operation status corresponding to the at least one violated system parameter based on the plurality of fault conditions; and

20

in response to determining that there is an existing type of fault operation status corresponding to the at least one violated threshold parameter, determining the existing type of fault operation status as the operation status.

17. The system according to claim 16, wherein when there is no existing type of fault operation status corresponding to the at least one violated system parameter, the at least one processor is further configured to create a new fault condition in relation to an unknown type of fault operation status, wherein the new fault condition is created based on the at least one violated system parameter and one or more correlated system parameters correlating with the at least one violated system parameter.

25
30

18. The system according to claim 16, wherein the fault condition repository comprises correlation information of the plurality of system parameters, and fault type parameter association information of the plurality of types of fault operation statuses, wherein for each type of fault operation status, the fault type parameter association

information indicates one or more system parameters of the plurality of system parameters being associated to the type of fault operation status, and the at least one processor is further configured to:

5 determine, for each violated system parameter of the at least one violated system parameter, one or more correlated system parameters with which the at least one violated threshold parameter is correlated based on the correlation information;

determine, for each violated system parameter of the at least one violated system parameter, an existing type of fault operation status to which the one or more correlated system parameters are associated based on the fault type parameter association
10 information; and

associate, for each violated system parameter of the at least one violated system parameter, the violated system parameter to said existing type of fault operation status to produce an updated set of system parameters associated to said existing type of fault operation status in the fault condition repository.

15

19. The system according to claim 15, wherein said updating the fault condition repository if the match result does not satisfy the predetermined fault condition repository update condition comprises:

20 updating a fault condition associated to the type of fault operation status determined based on an objective function to obtain a maximum match result between the first detection result and the second detection result, wherein said updating the fault condition comprises adjusting one or more threshold parameters associated to one or more system parameters, respectively, corresponding to the fault condition.

25 20. The system according to claim 12, wherein the at least one processor is further configured to perform validation of the labelled data in relation to the operation status determined to produce validated labelled data for training a fault model.

21. The system according to claim 20, wherein the at least one processor is further
30 configured to train the fault model based on the validated labelled data.

22. The system according to any one of claims 12 to 21, wherein the operational data comprises measured data in relation to the plurality of systems parameters of the

plurality of operational components obtained from one or more sensors associated to the operational components.

23. A computer program product, embodied in one or more non-transitory
5 computer-readable storage mediums, comprising instructions executable by at least one processor to perform a method of fault detection and diagnostic for a building management system according to any one of claims 1 to 11.