



(51) International Patent Classification:
G05B 23/02 (2006.01)

(21) International Application Number:
PCT/US2017/059436

(22) International Filing Date:
01 November 2017 (01.11.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/417,757 04 November 2016 (04.11.2016) US
15/793,358 25 October 2017 (25.10.2017) US

(71) Applicant: **NEC LABORATORIES AMERICA, INC**
[US/US]; 4 Independence Way, Suite 200, Princeton, New
Jersey 08540 (US).

(72) Inventors: **CHENG, Wei**; 3011 Quail Ridge Drive, Plains-
boro, New Jersey 08536 (US). **CHEN, Haifeng**; 11 Black-
hawk Court, West Windsor, New Jersey 08550 (US).
JIANG, Guofei; 5 Danby Court, Princeton, New Jersey
08540 (US).

(74) Agent: **KOLODKA, Joseph**; NEC Laboratories America,
Inc., 4 Independence Way, Suite 200, Princeton, New Jersey
08540 (US).

(81) Designated States (*unless otherwise indicated, for every
kind of national protection available*): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ,
CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO,
DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN,
HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP,
KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME,
MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ,
OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,
SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every
kind of regional protection available*): ARIPO (BW, GH,
GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ,
UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ,
TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK,
EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV,
MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM,
TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW,
KM, ML, MR, NE, SN, TD, TG).

(54) Title: CONTENT-AWARE ANOMALY DETECTION AND DIAGNOSIS

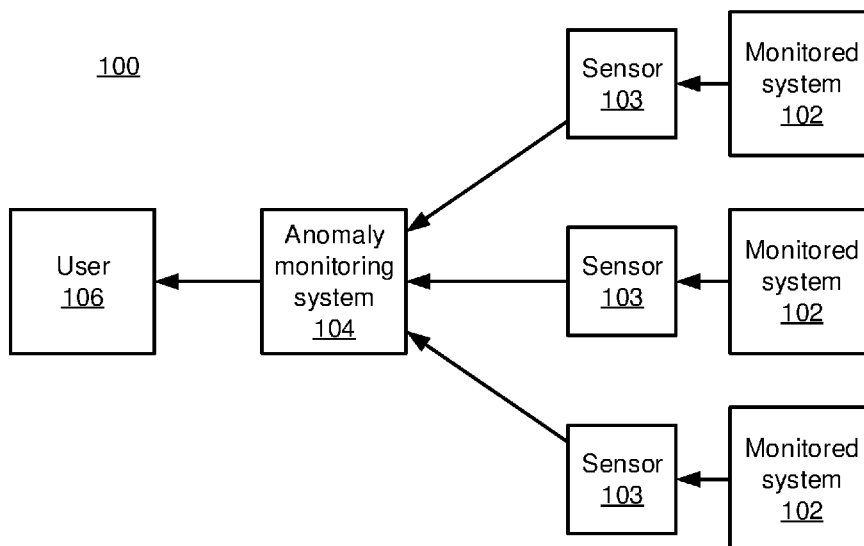


FIG. 1

(57) Abstract: Methods and systems for detecting a system fault include determining a network of broken correlations for a current timestamp, relative to a predicted set of correlations, based on a current set of sensor data. The network of broken correlations for the current timestamp is compared to networks of broken correlations for previous timestamps to determine a fault propagation pattern. It is determined whether a fault has occurred based on the fault propagation pattern. A system management action is performed if a fault has occurred.



Published:

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

CONTENT-AWARE ANOMALY DETECTION AND DIAGNOSIS

RELATED APPLICATION INFORMATION

[0001] This application claims priority to 62/417,757, filed on November 4, 2016, incorporated herein by reference herein its entirety.

BACKGROUND

Technical Field

[0002] The present invention relates to anomaly detection and, more particularly, to detection of anomalous behavior in computerized systems using a progression of deviations from expected behavior.

Description of the Related Art

[0003] Fault diagnosis is the process of discovering information about system failures. Previous efforts to perform fault diagnosis have been based on the system invariant model, but have not been able to consider the content of vanishing correlations in a system and the temporal patterns thereof. System status is generally quite dynamic and time-evolving, resulting in false positives for conventional fault detection.

SUMMARY

[0004] A method for detecting a system fault includes determining a network of broken correlations for a current timestamp, relative to a predicted set of correlations, based on a current set of sensor data. The network of broken correlations for the current timestamp is compared to networks of broken correlations for previous timestamps to determine a fault

propagation pattern. It is determined whether a fault has occurred based on the fault propagation pattern. A system management action is performed if a fault has occurred.

[0005] A system for detecting a fault includes an invariant graph module configured to determine a network of broken correlations for a current timestamp, relative to a predicted set of correlations, based on a current set of sensor data. An invariant comparison module includes a processor configured to compare the network of broken correlations for the current timestamp to networks of broken correlations for previous timestamps to determine a fault propagation pattern and to determine whether a fault has occurred based on the fault propagation pattern. A fault management module is configured to perform a management action if a fault has occurred.

[0006] These and other features and advantages will become apparent from the following detailed description of illustrative embodiments thereof, which is to be read in connection with the accompanying drawings.

BRIEF DESCRIPTION OF DRAWINGS

[0007] The disclosure will provide details in the following description of preferred embodiments with reference to the following figures wherein:

[0008] FIG. 1 is a block diagram of a system for monitoring anomalies across a set of monitored systems in accordance with an embodiment of the present invention;

[0009] FIG. 2 is a block/flow diagram of a method for detecting faults across a set of monitored systems in accordance with an embodiment of the present invention;

[0010] FIG. 3 is a block/flow diagram of a method for comparing broken invariant graphs over time to determine whether a fault has occurred in accordance with an embodiment of the present invention;

[0011] FIG. 4 is a block diagram of an anomaly monitoring system in accordance with an embodiment of the present invention; and

[0012] FIG. 5 is a block diagram of a processing system in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0013] Embodiments of the present invention provide fault diagnosis by tracking the temporal evolution pattern of vanishing correlations in system monitoring data. Different evolving patterns of vanishing correlations from snapshots at multiple points in time provide the ability to track the early stages of a system failure and can be used to distinguish normal changes in a system's state and real system failures.

[0014] The present embodiments automatically diagnose system failures and determine whether the detected event is a true fault or not. Faults are traced back to collect evidence regarding the fault's origins. To accomplish this, the present embodiments jointly consider multiple snapshots of the system and detect when system invariants are violated. Patterns in such violations are detected and distinguished from random violations, which would be considered false positives that arise from normal system fluctuations.

[0015] Referring now in detail to the figures in which like numerals represent the same or similar elements and initially to FIG. 1, a security system 100 is shown. The security system 100 includes a number of monitored systems 102, each of which may represent, for

example, a different computer, a different computer sub-system, a specific device, or a component of a device. For example, the monitored systems 102 may represent the components of a power generator, with each monitored system 102 representing the operation of that component. The operations of each monitored system 102 are monitored by a respective sensor 103, which collects data regarding the state of the monitored system, any events that occur within the monitored system, any log output from the monitored system, etc. In one example, a sensor 103 may measure physical states of a monitored system 102, such as temperature, vibration, etc.

[0016] Information from the sensors 103 is collected by an anomaly monitoring system 104. The anomaly monitoring system tracks correlations between the sensor outputs, noting in particular when invariants between such sensor outputs are broken. The anomaly monitoring system 104 finds patterns in the breaks in the invariants and uses the discovered patterns to distinguish between random fluctuations, changes in system state, and system failures. The anomaly monitoring system 104 then reports the discovered information to a user 106. In some embodiments, the anomaly monitoring system 104 provides control information to the monitored systems 102 to automatically address system failures.

[0017] Referring now to FIG. 2, a method for system failure analysis is shown. Block 202 trains a model (e.g., a system invariant analysis technology (SIAT) model) based on sensor readings from the monitored sensors 103 during normal system operation. In one example, the model is trained as a network of invariants that profile hidden, pair-wise correlations between different system components.

[0018] Links in an invariant network can be obtained to form an SIAT model. For a pair of time series from a system, $x(t)$ and $y(t)$, each time series being generated by respective

system components, where t represents a time index, the relationship between the two time series can be described by a linear regression function:

$$y(t) + a_1y(t-1) + \dots + a_ny(t-n) = b_0x(t) + \dots + b_mx(t-m)$$

where $[n, m]$ is referred to as the order of the model, which determines how many previous steps affect the current outputs $x(t)$ and $y(t)$. Parameters a_i and b_j indicate how strongly a previous step impacts the current output, which can be learned by the least-square fitting of the linear regression function to the training data.

[0019] Once the parameters are obtained, a fitness score can be used to evaluate how well the learned model fits the observed measurement data. A large fitness score indicates the relationship (x, y) holds well in normal system state. A pair of time series with a fitness score larger than a pre-defined threshold is referred to as an invariant of the system. The invariant network includes all invariants when the system is a normal state. During the system testing period, a broken invariant can be detected by checking if the time series generated by the two connected components still fit the learned model. If the error is large, the invariant (x, y) is declared to be broken. A broken network includes all broken invariants.

[0020] At each time snapshot during system runtime, block 204 collects a new set of sensor data. Block 206 then determines whether the correlations predicted by the model are present in the new sensor data and block 208 generates a broken invariant graph. If a correlation does not exist, this is represented by a broken link in the broken invariant graph, while confirmed predictions are represented by a whole link.

[0021] Block 210 compares the broken invariant graph of the current timestamp to those of previous timestamps to determine patterns in the evolution of the graphs. In particular,

block 210 discriminates between broken invariants that appear to be the result of noise in the system (e.g., transient fluctuations that temporarily disrupt a predicted correlation) and patterns that propagate or that suddenly affect large numbers of correlations at once.

[0022] Block 212 thereby determines the status of the system 100. It is specifically understood that a lack of propagation of broken invariants across the invariant network represents a normal running scenario, where correlations are only temporarily disrupted and resume on their own. In contrast, for real system faults, the disruption of one monitored system 102 may lead to further disruptions in connected monitored systems 102, leading to a propagation of broken correlations as time goes on. Notably, this distinction is not based on the number of broken invariants in a given snapshot, but rather on how the pattern of broken invariants evolves over time.

[0023] Toward that end, blocks 210 and 212 begin at the time a broken invariant is detected and trace the contents of broken invariants backwards in time to locate early indications of a fault. Based on a precision and recall curve, block 210 determines if there exists a propagation pattern. This leads to three different cases. If broken invariants illustrate propagation behavior, block 212 determines that the anomaly represents a real fault in the system. If broken invariants appear randomly over a period of time, then block 212 determines that the anomaly represents a false positive. If a number of broken invariants show up instantly, with no evolving pattern, then block 212 determines that the system state has changed (e.g., turning off the system, starting a new workload, etc.), which may represent normal system behavior or may represent a catastrophic failure. The change of a system state reflects the fact that the system's running settings are changed globally and may refer to, e.g., upgrading system equipment or different running parameter settings,

in which the system running statuses are considered to be normal, but which result in very different results from the time series monitoring.

[0024] If block 214 determines that a fault was detected, block 216 performs a system management action. In one embodiment, block 216 simply generates an alert to a human operator 106, including the anomalous timestamp and information regarding the relationships between sensors 103 that are characterized by the broken invariants, making it possible for the user 106 to investigate the fault. Thus, for example, if a pair of sensors 103 are expected to have a particular relationship (e.g., where a first sensor's value is a function of the second sensor's value) and the monitored systems 102 in question have deviated from the expected behavior, the report may indicate which monitored systems 102 are at issue, what the expected behavior was, and what the measured behavior is. The report may also indicate a measure of the magnitude of the anomaly (e.g., the number of broken invariants).

[0025] In another embodiment, block 216 take automatic actions to address the detected fault. Thus, for example, if the anomaly implicates monitored systems 102 that are particularly sensitive or dangerous, block 216 may automatically issue instructions to those monitored systems 102 to slow or stop their processes.

[0026] Because block 212 considers possible propagation of system faults, the present embodiments can also provide early warning when a number of broken invariants is still not high, but shows a clear fault propagation pattern. In addition, the present embodiments can also discover the time when the fault initially occurs by tracking a propagation pattern back to its origin, making it easier to identify the root cause of the fault.

[0027] Referring now to FIG. 3, additional detail on the comparison of broken invariants at different times is shown. In some embodiments, precision and recall curves can be used in pattern recognition to represent the fraction of relevant instances among a set of retrieved instances and a fraction of relevant instances that have been retrieved over the total number of relevant instances, respectively. Thus, if the set of broken invariants at a present time (alternatively, the latest timestamp being considered) is denoted as G_E , and the set of broken invariants at an earlier time t is given as G_t , then the precision is given as $\frac{|G_E \cap G_t|}{|G_t|}$ and the recall is given as $\frac{|G_E \cap G_t|}{|G_E|}$. These values are plotted for different values of t to form the precision and recall curves in block 302.

[0028] The precision and recall curves can be noisy, so smoothing may be applied to each curve by block 304. To accomplish this, block 304 uses a small window to slide over the precision and recall curves, with the values of each in the window being replaced by the value of the majority of values (in alternative embodiments, the average value) within the window.

[0029] Block 306 traces back the precision curve over time until the precision value is below a threshold (e.g., about 0.8). The detected region guarantees that the prevision is at least above some threshold where the content in the snapshots are almost all shown in the reference snapshot, making them more likely to be root causes.

[0030] Within a time range that begins at the time at which the precision value drops below the threshold and ends at the final timestamp, block 308 checks whether the recall curve is increasing, where the increase may be, for example, strict monotonic or piecewise (where the mean of recall values in different time ranges are increasing). If so, block 308 determines a system fault. If no change is present, then block 308 determines a context

switch. If the recall curve is shaking, then block 308 determines a false positive due to system noise. The beginning of the time range is tagged as the initial time of the event.

[0031] Embodiments described herein may be entirely hardware, entirely software or including both hardware and software elements. In a preferred embodiment, the present invention is implemented in software, which includes but is not limited to firmware, resident software, microcode, etc.

[0032] Embodiments may include a computer program product accessible from a computer-usable or computer-readable medium providing program code for use by or in connection with a computer or any instruction execution system. A computer-usable or computer readable medium may include any apparatus that stores, communicates, propagates, or transports the program for use by or in connection with the instruction execution system, apparatus, or device. The medium can be magnetic, optical, electronic, electromagnetic, infrared, or semiconductor system (or apparatus or device) or a propagation medium. The medium may include a computer-readable storage medium such as a semiconductor or solid state memory, magnetic tape, a removable computer diskette, a random access memory (RAM), a read-only memory (ROM), a rigid magnetic disk and an optical disk, etc.

[0033] Each computer program may be tangibly stored in a machine-readable storage media or device (e.g., program memory or magnetic disk) readable by a general or special purpose programmable computer, for configuring and controlling operation of a computer when the storage media or device is read by the computer to perform the procedures described herein. The inventive system may also be considered to be embodied in a computer-readable storage medium, configured with a computer program, where the

storage medium so configured causes a computer to operate in a specific and predefined manner to perform the functions described herein.

[0034] A data processing system suitable for storing and/or executing program code may include at least one processor coupled directly or indirectly to memory elements through a system bus. The memory elements can include local memory employed during actual execution of the program code, bulk storage, and cache memories which provide temporary storage of at least some program code to reduce the number of times code is retrieved from bulk storage during execution. Input/output or I/O devices (including but not limited to keyboards, displays, pointing devices, etc.) may be coupled to the system either directly or through intervening I/O controllers.

[0035] Network adapters may also be coupled to the system to enable the data processing system to become coupled to other data processing systems or remote printers or storage devices through intervening private or public networks. Modems, cable modem and Ethernet cards are just a few of the currently available types of network adapters.

[0036] Referring now to FIG. 4, additional detail on the anomaly monitoring system 104 is shown. The system 104 includes a hardware processor 402 and memory 404. A sensor interface 406 communicates with the sensors 103 and stores sensor information in the memory 404 in any appropriate format (e.g., in a time-indexed database). The sensor interface 406 may represent a network interface that communicates over an appropriate wired or wireless medium and may, additionally or alternatively, include dedicated interfaces configured to communicate with specific sensors 103. The system 104 also includes one or more functional modules that may, in some embodiments, be implemented as software that is stored in memory 404 and executed by the hardware processor 402. In

other embodiments, the modules may be implemented as one or more application-specific integrated chips or field programmable gate arrays.

[0037] A training module 408 trains a prediction model 410 based on sensor information collected during a period of normal operation. The prediction model 410 is stored in memory 404 and establishes expected correlations between different sensor outputs and system states.

[0038] Invariant graph module 412 then collects sensor information for a current timestamp and determines which invariants predicted by the predicted model 410 have been broken. These broken invariant graphs are stored in the memory 404. Invariant comparison module 414 compares the broken invariant graphs over a time range to determine patterns in fault propagation, in particular discriminating between true faults and false positives.

[0039] Fault management module 416 takes an appropriate automatic management action in response to a fault being detected. In some embodiments, the fault management module 416 generates an alert for a user 106 using a user interface 418. In other embodiments, the fault management module 416 automatically takes action with respect to the fault, for example adjusting operational parameters or shutting down systems to prevent a fault from propagating or to prevent damage to the monitored systems 102.

[0040] Referring now to FIG. 5, an exemplary processing system 500 is shown which may represent the anomaly monitoring system 104. The processing system 500 includes at least one processor (CPU) 504 operatively coupled to other components via a system bus 502. A cache 506, a Read Only Memory (ROM) 508, a Random Access Memory (RAM) 510, an input/output (I/O) adapter 520, a sound adapter 530, a network adapter

540, a user interface adapter 550, and a display adapter 560, are operatively coupled to the system bus 502.

[0041] A first storage device 522 and a second storage device 524 are operatively coupled to system bus 502 by the I/O adapter 520. The storage devices 522 and 524 can be any of a disk storage device (e.g., a magnetic or optical disk storage device), a solid state magnetic device, and so forth. The storage devices 522 and 524 can be the same type of storage device or different types of storage devices.

[0042] A speaker 532 is operatively coupled to system bus 502 by the sound adapter 530. A transceiver 542 is operatively coupled to system bus 502 by network adapter 540. A display device 562 is operatively coupled to system bus 502 by display adapter 560.

[0043] A first user input device 552, a second user input device 554, and a third user input device 556 are operatively coupled to system bus 502 by user interface adapter 550. The user input devices 552, 554, and 556 can be any of a keyboard, a mouse, a keypad, an image capture device, a motion sensing device, a microphone, a device incorporating the functionality of at least two of the preceding devices, and so forth. Of course, other types of input devices can also be used, while maintaining the spirit of the present principles. The user input devices 552, 554, and 556 can be the same type of user input device or different types of user input devices. The user input devices 552, 554, and 556 are used to input and output information to and from system 500.

[0044] Of course, the processing system 500 may also include other elements (not shown), as readily contemplated by one of skill in the art, as well as omit certain elements. For example, various other input devices and/or output devices can be included in processing system 500, depending upon the particular implementation of the

same, as readily understood by one of ordinary skill in the art. For example, various types of wireless and/or wired input and/or output devices can be used. Moreover, additional processors, controllers, memories, and so forth, in various configurations can also be utilized as readily appreciated by one of ordinary skill in the art. These and other variations of the processing system 500 are readily contemplated by one of ordinary skill in the art given the teachings of the present principles provided herein.

[0045] The foregoing is to be understood as being in every respect illustrative and exemplary, but not restrictive, and the scope of the invention disclosed herein is not to be determined from the Detailed Description, but rather from the claims as interpreted according to the full breadth permitted by the patent laws. It is to be understood that the embodiments shown and described herein are only illustrative of the principles of the present invention and that those skilled in the art may implement various modifications without departing from the scope and spirit of the invention. Those skilled in the art could implement various other feature combinations without departing from the scope and spirit of the invention. Having thus described aspects of the invention, with the details and particularity required by the patent laws, what is claimed and desired protected by Letters Patent is set forth in the appended claims.

WHAT IS CLAIMED IS:

1. A method for detecting a system fault, comprising:
determining a network of broken correlations for a current timestamp, relative to a predicted set of correlations, based on a current set of sensor data;
comparing the network of broken correlations for the current timestamp to networks of broken correlations for previous timestamps to determine a fault propagation pattern;
determining whether a fault has occurred based on the fault propagation pattern using a processor; and
performing a system management action if a fault has occurred.
2. The method of claim 1, wherein determining the network of broken correlations comprises determining which predicted correlations from a prediction model are satisfied and which are not.
3. The method of claim 2, further comprising training the prediction model based on sensor data for a period of normal system operation.
4. The method of claim 1, wherein comparing the network of broken correlations for the current timestamp to networks of broken correlations for previous timestamps comprises determining a precision curve and a recall curve.

5. The method of claim 4, wherein comparing the network of broken correlations for the current timestamp to networks of broken correlations for previous timestamps further comprises smoothing the precision curve and the recall curve.

6. The method of claim 4, wherein comparing the network of broken correlations for the current timestamp to networks of broken correlations for previous timestamps further comprises determining a time range between the current timestamp and a first timestamp at which a value of the precision curve drops below a threshold.

7. The method of claim 6, wherein comparing the network of broken correlations for the current timestamp to networks of broken correlations for previous timestamps further comprises determining a behavior of the recall curve within the time range.

8. The method of claim 7, wherein determining whether a fault has occurred comprises determining that a fault has occurred if the recall curve increases monotonically in the time range.

9. The method of claim 1, wherein performing a system management action comprises changing a state of the system in response to the detected fault.

10. A system for detecting a fault, comprising:

an invariant graph module configured to determine a network of broken correlations for a current timestamp, relative to a predicted set of correlations, based on a current set of sensor data;

an invariant comparison module comprising a processor configured to compare the network of broken correlations for the current timestamp to networks of broken correlations for previous timestamps to determine a fault propagation pattern and to determine whether a fault has occurred based on the fault propagation pattern; and

a fault management module configured to perform a management action if a fault has occurred.

11. The system of claim 10, wherein the invariant graph module is further configured to determine which predicted correlations from a prediction model are satisfied and which are not.

12. The system of claim 11, further comprising a training module configured to train the prediction model based on sensor data for a period of normal system operation.

13. The system of claim 10, wherein the invariant comparison module is further configured to determine a precision curve and a recall curve.

14. The system of claim 13, wherein the invariant comparison module is further configured to smooth the precision curve and the recall curve.

15. The system of claim 13, wherein the invariant comparison module is further configured to determine a time range between the current timestamp and a first timestamp at which a value of the precision curve drops below a threshold.

16. The system of claim 15, wherein the invariant comparison module is further configured to determine a behavior of the recall curve within the time range.

17. The system of claim 16, wherein the invariant comparison module is further configured to determine that a fault has occurred if the recall curve increases monotonically in the time range.

18. The system of claim 10, the fault management module is further configured to change a state of the system in response to the detected fault.

1/5

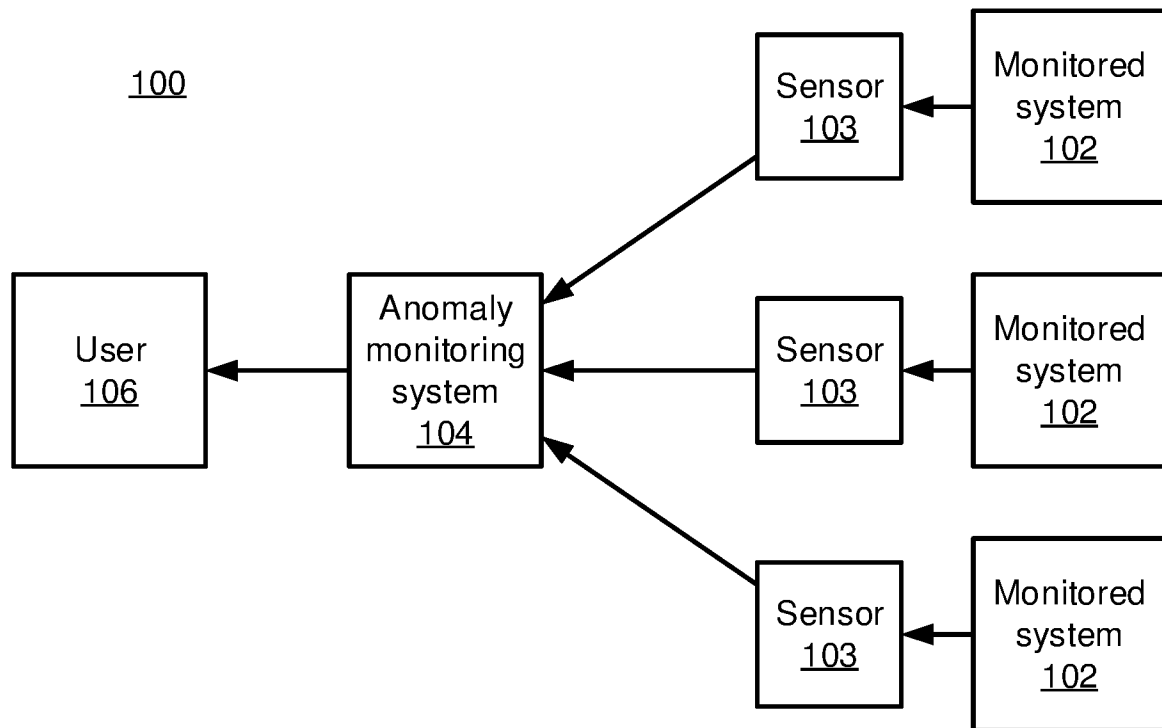


FIG. 1

2/5

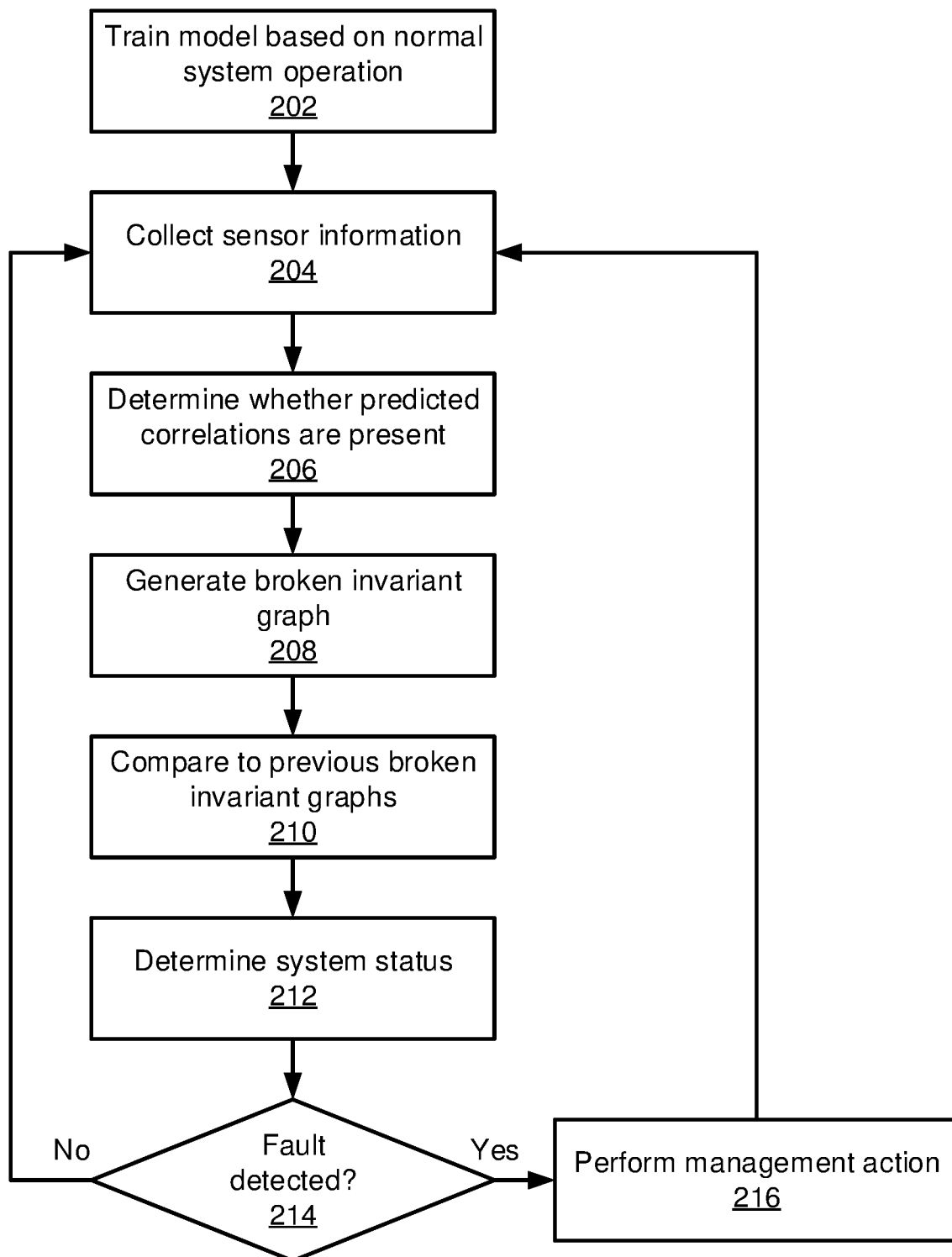


FIG. 2

3/5

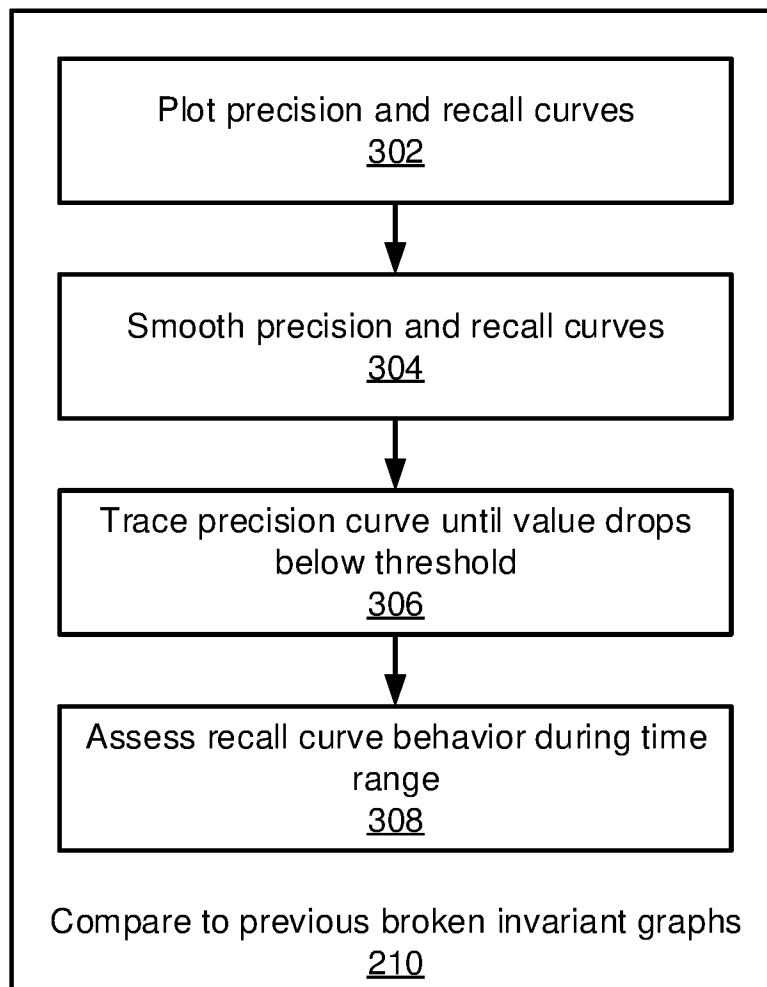


FIG. 3

4/5

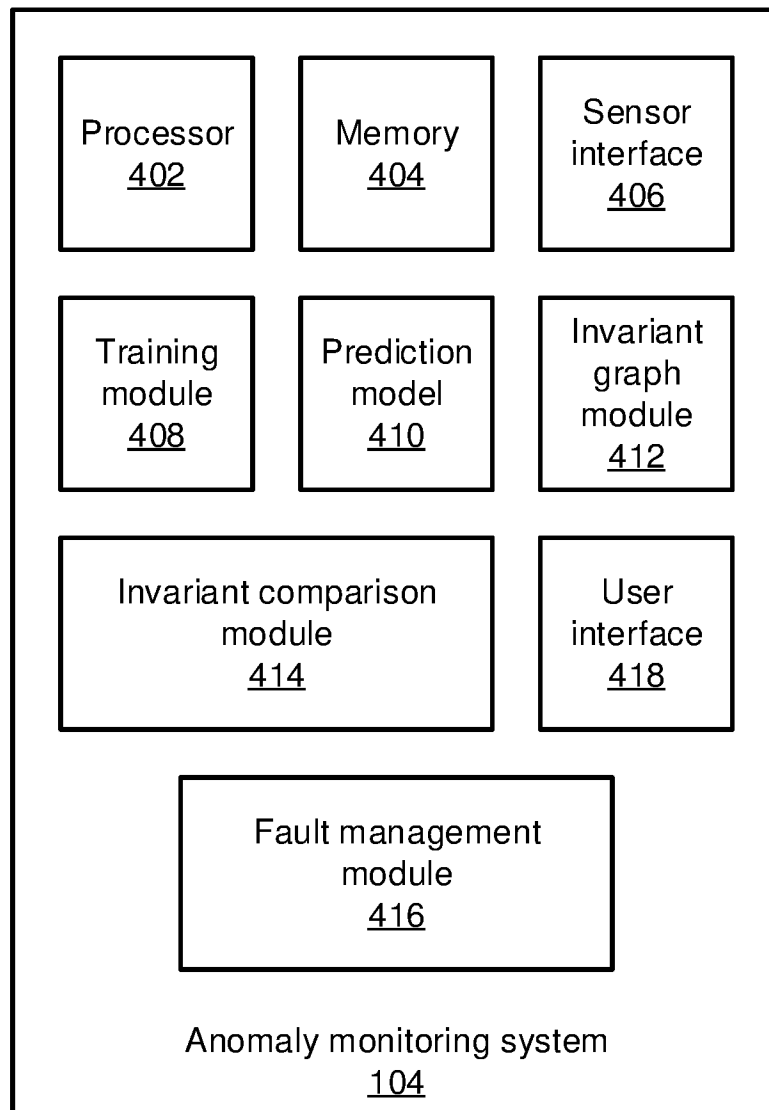


FIG. 4

5/5

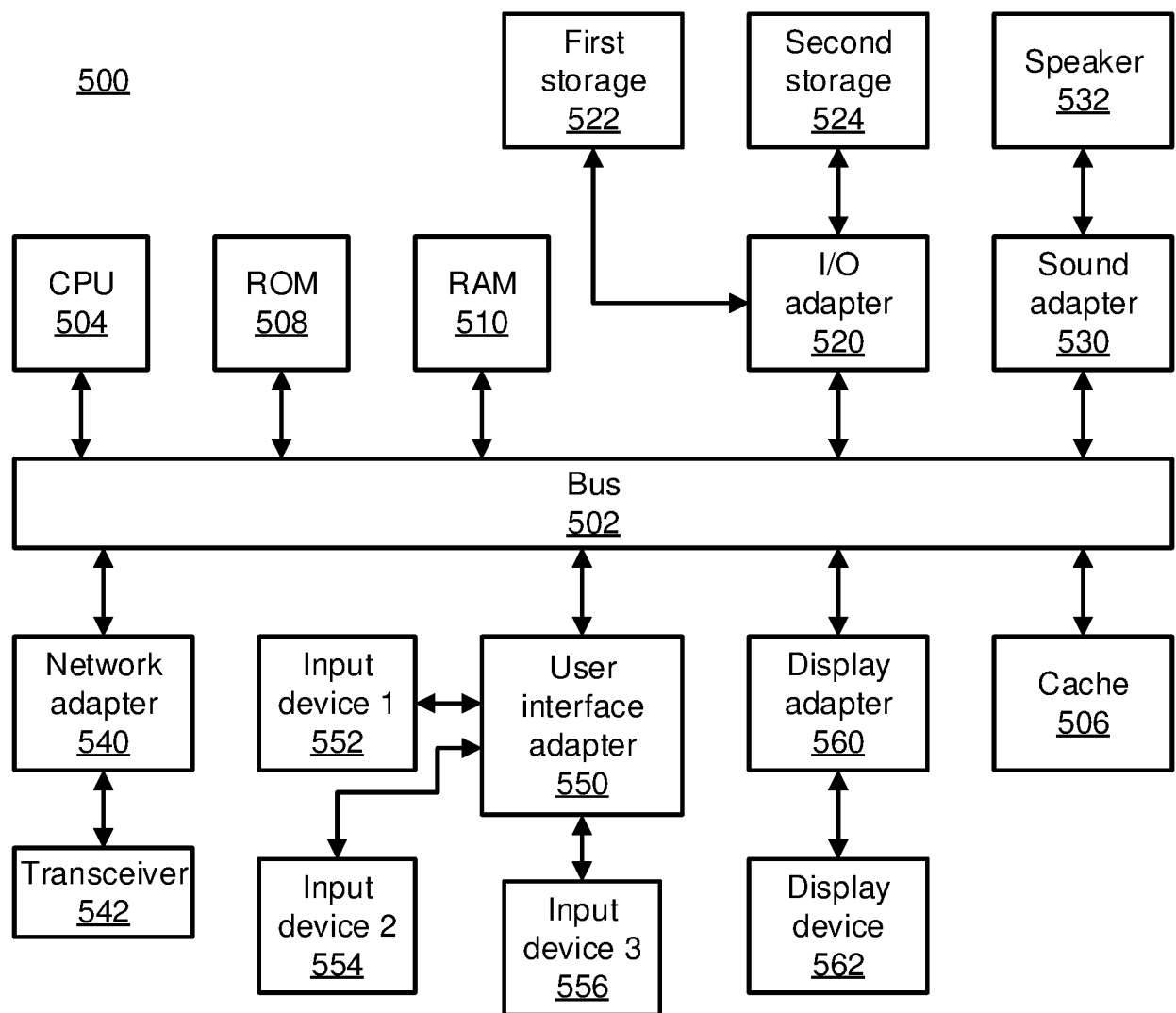


FIG. 5

A. CLASSIFICATION OF SUBJECT MATTER**G05B 23/02(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G05B 23/02; G06N 7/00; G06F 11/07; H04L 1/22; H04L 12/24; G06K 9/00; G06N 5/02

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: network, fault, detection, correlation, predicted-set, pattern

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2016-0285700 A1 (FUTUREWEI TECHNOLOGIES, INC.) 29 September 2016 See paragraphs [0030], [0053]-[0056], [0085]; and claims 1-4.	1-3, 9-12, 18
Y		4-8, 13-17
Y	US 2013-0132331 A1 (ADAM KOWALCZYK et al.) 23 May 2013 See paragraph [0013]; and claims 1, 10-11.	4-8, 13-17
A	US 2015-0378806 A1 (NEC CORPORATION) 31 December 2015 See claim 1; and figures 2-3.	1-18
A	US 2004-0168100 A1 (MARINA K. THOTTAN et al.) 26 August 2004 See paragraphs [0149]-[0151]; and claim 1.	1-18
A	US 2013-0034263 A1 (YUANYUAN DING et al.) 07 February 2013 See claim 1; and figures 2, 6, 8.	1-18



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

08 March 2018 (08.03.2018)

Date of mailing of the international search report

08 March 2018 (08.03.2018)

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

CHIN, Sang Bum

Telephone No. +82-42-481-8398



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2017/059436

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2016-0285700 A1	29/09/2016	CN 107409075 A EP 3259881 A1 WO 2016-150395 A1	28/11/2017 27/12/2017 29/09/2016
US 2013-0132331 A1	23/05/2013	AU 2011-226739 A1 AU 2011-226740 A1 US 2013-0198118 A1 WO 2011-109863 A1 WO 2011-109864 A2	04/10/2012 04/10/2012 01/08/2013 15/09/2011 15/09/2011
US 2015-0378806 A1	31/12/2015	EP 2963552 A1 EP 2963552 A4 JP 6183449 B2 WO 2014-132611 A1	06/01/2016 27/07/2016 23/08/2017 04/09/2014
US 2004-0168100 A1	26/08/2004	AU 2004902 A WO 02-46928 A1 WO 02-46928 A9	18/06/2002 13/06/2002 17/04/2003
US 2013-0034263 A1	07/02/2013	US 8948522 B2	03/02/2015