(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 September 2018 (13.09.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/164767 A1

(51) International Patent Classification:
*H04L 29/06* (2006.01)

(21) International Application Number:
PCT/US2018/012858

(22) International Filing Date:
09 January 2018 (09.01.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/454,144 09 March 2017 (09.03.2017) US

(71) Applicant: **GENERAL ELECTRIC COMPANY** [US/US]; 1 River Road, Schenectady, NY 12345 (US).

(72) Inventors: **MESTHA, Lalit, Keshav**; One Research Circle, Niskayuna, NY 12309 (US). **ANUBI, Olugbenga**; One Research Circle, Niskayuna, NY 12309 (US). **ABBASZADEH, Masoud**; One Research Circle, Niskayuna, NY 12309 (US).

(74) Agent: **STANLEY, Scott, R.** et al.; General Electric Company, Global Patent Operation, 901 Main Avenue, 3rd Floor, Norwalk, CT 06851 (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,

WO 2018/164767 A1

(54) Title: CYBER-ATTACK DETECTION AND NEUTRALIZATION

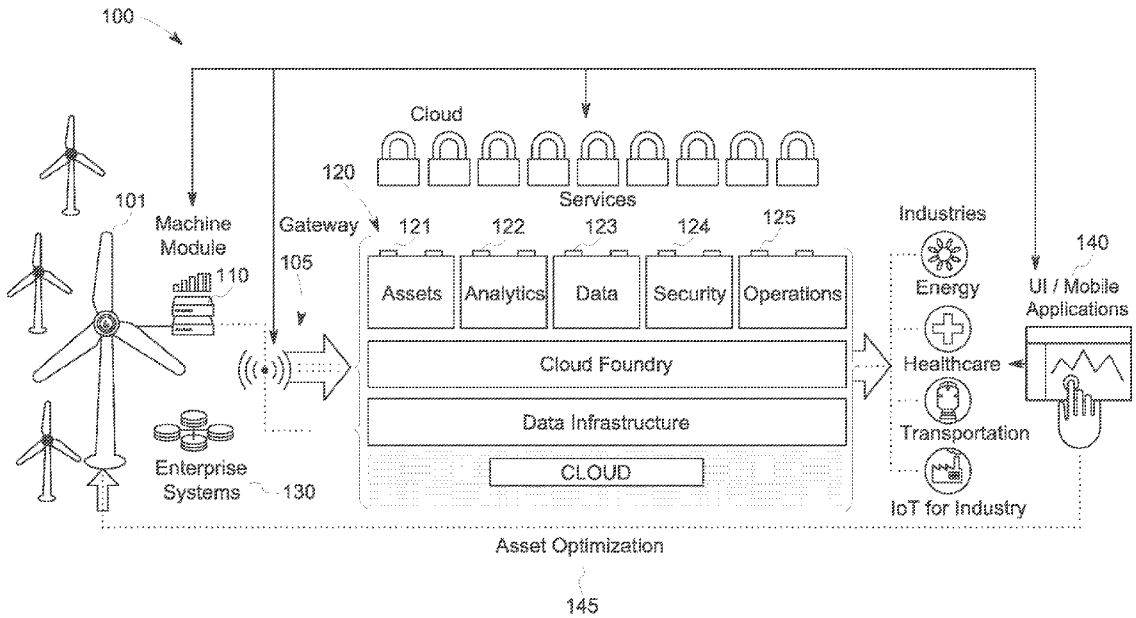



FIG. 1

(57) Abstract: The example embodiments are directed to a system and method for neutralizing abnormal signals in a cyber-physical system. In one example, the method includes receiving input signals comprising time series data associated with an asset and transforming the input signals into feature values in a feature space, detecting one or more abnormal feature values in the feature space based on a predetermined normalcy boundary associated with the asset, and determining an estimated true value for each abnormal feature value, and performing an inverse transform of each estimated true value to generate neutralized signals comprising time series data and outputting the neutralized signals.

*[Continued on next page]*

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).

**Declarations under Rule 4.17:**

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

**Published:**

— *with international search report (Art. 21(3))*

# CYBER-ATTACK DETECTION AND NEUTRALIZATION

## CROSS-REFERENCE TO RELATED APPLICATION(S)

**[0001]** This application claims the benefit under 35 USC §120 of US Patent Application Serial No. 15/447,861, filed on March 2, 2017, the entire disclosure of which is incorporated by reference herein for all purposes.

## GOVERNMENT INTEREST

**[0002]** This invention was made with Government support under contract number DEOE0000833 awarded by the United States Department of Energy. The Government has certain rights in this invention.

## BACKGROUND

**[0003]** Machine and equipment assets, generally, are engineered to perform particular tasks as part of a business process. For example, assets can include, among other things and without limitation, industrial manufacturing equipment on a production line, drilling equipment for use in mining operations, wind turbines that generate electricity on a wind farm, transportation vehicles, and the like. As another example, assets may include healthcare machines and equipment that aid in diagnosing patients such as imaging devices (e.g., X-ray or MRI systems), monitoring devices, and the like. The design and implementation of these assets often takes into account both the physics of the task at hand, as well as the environment in which such assets are configured to operate.

**[0004]** Low-level software and hardware-based controllers have long been used to drive machine and equipment assets. However, the rise of inexpensive cloud computing, increase in sensor capabilities, decrease in sensor costs, and the proliferation of mobile technologies have generated new opportunities for creating novel industrial and healthcare based assets with improved sensing technology and which are capable of transmitting data that can then be distributed throughout a network. As a consequence, there are new opportunities to enhance the business value of some assets through the use of novel industrial-focused hardware and software.

[0005] To prevent infection, human beings have developed various environmental precautions (*e.g.*, public health programs, clean water etc.) as well as external physical protection mechanisms (e.g., washing hands, protective clothing etc.). Despite these protections in place, various pathogens are able to attack the human body. In response, biological structures of a person's immune system may constantly defend against these pathogens to keep the body functioning. In a similar fashion, to protect industrial control systems, information technology mechanisms (*e.g.*, anti-virus software, trusted boot, etc.) has been developed as well as operational technology mechanisms (*e.g.*, Opshield, SecurityST, etc.) have been developed.

[0006] However, information technology and operational technology protection mechanisms are inadequate. As a result, cyber threats continue to grow in number and sophistication. Attackers have shown the ability to find and exploit residual faults and use them to formulate cyber-attacks. Most software systems in use today run substantially similar software. As a result, a successful cyber-attack can bring down a large number of installations running similar software. Furthermore, as more software is made available (*e.g.*, through the cloud), the situation can only get worse. Accordingly, what is needed is a better method for protecting control systems against cyber-attacks.

SUMMARY

[0007] Embodiments described herein improve upon the prior art by providing an attack detection and neutralization system and method capable of masking the effects of a cyber-attack on a physical asset such as a machine or equipment. The system may receive an incoming signal sensed from or about a physical asset, filter the signal to detect and remove an attack signature, and output the signal to a control system. In some aspects, the method can be implemented as software that is deployed on a cloud platform such as an Industrial Internet of Things (IIoT).

[0008] In an aspect of an embodiment, provided is a method including receiving input signals from an asset comprising time series data associated with the asset and transforming the input signals into feature values in a feature space, detecting one or more abnormal feature values in the feature space based on a predetermined normalcy boundary associated with the asset, and determining an estimated true value for each abnormal feature value, and

performing an inverse transform of each estimated true value to generate neutralized signals comprising time series data and outputting the neutralized signals.

**[0009]** In an aspect of another embodiment, provided is computing system including a receiver configured to receive input signals from an asset comprising time series data associated with the asset, a processor configured to transform the input signals into feature values in a feature space, detect one or more abnormal feature values in the feature space based on a predetermined normalcy boundary associated with the asset, determine an estimated true value for each abnormal feature value, and perform an inverse transform of each estimated true value to generate neutralized signals comprising time series data, and an output configured to output the neutralized signals.

**[0010]** In an aspect of another example embodiment, provided is a non-transitory computer readable storage medium having stored therein instructions that when executed cause a processor to perform a method including receiving input signals from an asset comprising time series data associated with the asset and transforming the input signals into feature values in a feature space, detecting one or more abnormal feature values in the feature space based on a predetermined normalcy boundary associated with the asset, and determining an estimated true value for each abnormal feature value, and performing an inverse transform of each estimated true value to generate neutralized signals comprising time series data and outputting the neutralized signals.

**[0011]** Other features and aspects may be apparent from the following detailed description taken in conjunction with the drawings and the claims.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0012]** Features and advantages of the example embodiments, and the manner in which the same are accomplished, will become more readily apparent with reference to the following detailed description taken in conjunction with the accompanying drawings.

**[0013]** FIG. 1 is a diagram illustrating a cloud-computing environment associated with industrial systems in accordance with an example embodiment.

**[0014]** FIG. 2 is a diagram illustrating a threat neutralizer for filtering input signals in accordance with an example embodiment.

[0015] FIG. 3 is a diagram illustrating a boundary and performance estimation process for threat detection in accordance with an example embodiment.

[0016] FIG. 4 is a diagram illustrating a method for neutralizing cyber-based threats in accordance with example embodiments.

[0017] FIG. 5 is a diagram illustrating an example of a simulated performance of a boundary and performance constrained resilient estimator in accordance with an example embodiment.

[0018] FIG. 6 is a diagram illustrating a computing device for neutralizing cyber-based threats in accordance with an example embodiment.

[0019] Throughout the drawings and the detailed description, unless otherwise described, the same drawing reference numerals will be understood to refer to the same elements, features, and structures. The relative size and depiction of these elements may be exaggerated or adjusted for clarity, illustration, and/or convenience.

## DETAILED DESCRIPTION

[0020] In the following description, specific details are set forth in order to provide a thorough understanding of the various example embodiments. It should be appreciated that various modifications to the embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the disclosure. Moreover, in the following description, numerous details are set forth for the purpose of explanation. However, one of ordinary skill in the art should understand that embodiments may be practiced without the use of these specific details. In other instances, well-known structures and processes are not shown or described in order not to obscure the description with unnecessary detail. Thus, the present disclosure is not intended to be limited to the embodiments shown.

[0021] The example embodiments are directed to a system and method for identifying and neutralizing threats that occur in systems, for example, control systems for industrial manufacturing. The method described herein may achieve resiliency to cyber-attacks to provide uninterrupted, equipment-safe, controlled operation while attacks are in progress. As described herein, resiliency is the ability of a control system to force signals (or states)

required for operational normalcy. As in a human body, once the method described herein detects a threat (*e.g.*, an infection), the method destroys the threat. Analogous to biomimicry stated above, the example embodiments are directed to a method that extends domain-level security by creating a cyber-based "immune system" specifically desirable to industrial systems.

**[0022]** The methods herein are not only able to identify threats, but may also block the threats by automatically neutralizing their effects on the control system even while the attacks are in progress. Neutralization of the affects may be achieved at key signals by estimating true operational states of the system with new boundary and performance constrained resilient estimators (BPRE). Furthermore, true signals for operational normalcy may be computed from estimated states enabling the system to continue to function properly. The method and system described herein can deter threats by replicating protection mechanisms adapted by biological systems. How the human body fights off pathogens and other infections is a natural marvel. It automatically detects and triggers a self-defense mechanism to fight. The example embodiments replicate a similar kind of automatic detection and trigger functions for masking or otherwise neutralizing the attack signals. The neutralization approach disclosed here can also be used for securing any system where continuous data stream is available for monitoring purpose.

**[0023]** The neutralization methods described herein may be implemented as software such as an application or a service and may be incorporated within an industrial system such as a control system, a computer, a server, a cloud platform, a machine, an equipment, an aircraft, a locomotive, and the like. While progress with machine and equipment automation has been made over the last several decades, and assets have become 'smarter,' the intelligence of any individual asset pales in comparison to intelligence that can be gained when multiple smart devices are connected together, for example, in the cloud. As described herein, an asset is used to refer to equipment and/or a machine used in fields such as energy, healthcare, transportation, heavy manufacturing, chemical production, printing and publishing, electronics, textiles, and the like. Aggregating data collected from or about multiple assets can enable users to improve business processes, for example by improving effectiveness of asset maintenance or improving operational performance if appropriate industrial-specific data collection and modeling technology is developed and applied.

[0024] For example, an asset can be outfitted with one or more sensors configured to monitor respective operations or conditions thereof. Data from the sensors can be added to the cloud platform. By bringing such data into a cloud-based environment, new software applications and control systems informed by industrial process, tools and expertise can be constructed, and new physics-based analytics specific to an industrial environment can be created. Insights gained through analysis of such data can lead to enhanced asset designs, enhanced software algorithms for operating the same or similar assets, better operating efficiency, and the like. However, when data is transferred from an edge of the cloud to the cloud platform, attacks may occur. The attacks may affect the processing of signals received from the edge of the cloud. The example embodiments provide a neutralization process that is capable of filtering signals that are transmitted from the edge to the cloud platform back-end, as well as transmitting in other situations.

[0025] Assets described herein can include or can be a portion of an Industrial Internet of Things (IIoT). In an example, an IIoT connects assets including machines and equipment, such as turbines, jet engines, healthcare machines, locomotives, oil rigs, and the like, to the Internet and/or a cloud, or to each other in some meaningful way such as through one or more networks. The examples described herein can include using a "cloud" or remote or distributed computing resource or service. The cloud can be used to receive, relay, transmit, store, analyze, or otherwise process information for or about one or more assets. In an example, a cloud computing system includes at least one processor circuit, at least one database, and a plurality of users or assets that are in data communication with the cloud computing system. The cloud computing system can further include or can be coupled with one or more other processor circuits or modules configured to perform a specific task, such as to perform tasks related to asset maintenance, analytics, data storage, security, or some other function.

[0026] However, the integration of assets with the remote computing resources to enable the IIoT often presents technical challenges separate and distinct from the specific industry and from computer networks, generally. A given machine or equipment based asset may need to be configured with novel interfaces and communication protocols to send and receive data to and from distributed computing resources. Assets may have strict requirements for cost, weight, security, performance, signal interference, and the like, in which case enabling such an interface is rarely as simple as combining the asset with a general-purpose computing

device. To address these problems and other problems resulting from the intersection of certain industrial fields and the IIoT, embodiments provide a cloud platform that can receive and deploy applications from many different fields of industrial technologies. In order for these applications to successfully consume data from machines and equipment also connected to the cloud, the embodiments provide a threat neutralization method that can be implemented within the cloud and that can filter signals between an edge of the cloud and the cloud platform enabling the control of the asset to be maintained as normal.

**[0027]** The Predix™ platform available from GE is a novel embodiment of an Asset Management Platform (AMP) technology enabled by state of the art cutting edge tools and cloud computing techniques that enable incorporation of a manufacturer's asset knowledge with a set of development tools and best practices that enables asset users to bridge gaps between software and operations to enhance capabilities, foster innovation, and ultimately provide economic value. Through the use of such a system, a manufacturer of assets can be uniquely situated to leverage its understanding of assets themselves, models of such assets, and industrial operations or applications of such assets, to create new value for industrial customers through asset insights.

**[0028]** FIG. 1 illustrates a cloud computing environment associated with industrial systems in accordance with an example embodiment. FIG. 1 illustrates generally an example of portions of an asset management platform (AMP) 100. As further described herein, one or more portions of an AMP can reside in a cloud computing system 120, in a local or sandboxed environment, or can be distributed across multiple locations or devices. The AMP 100 can be configured to perform any one or more of data acquisition, data analysis, or data exchange with local or remote assets, or with other task-specific processing devices. The AMP 100 includes an asset community (*e.g.*, gas turbines, wind turbines, healthcare machines, industrial systems, manufacturing systems, oil rigs, etc.) that is communicatively coupled with the cloud computing system 120. In an example, a machine module 110 receives information from, or senses information about, at least one asset member of the asset community, and configures the received information for exchange with the cloud computing system 120. In an example, the machine module 110 is coupled to the cloud computing system 120 or to an enterprise computing system 130 via a communication gateway 105.

**[0029]** In an example, the communication gateway 105 may include or may use a wired or wireless communication channel that extends at least from the machine module 110 to the cloud computing system 120. The cloud computing system 120 may include several layers, for example, a data infrastructure layer, a cloud foundry layer, and modules for providing various functions. In FIG. 1, the cloud computing system 120 includes an asset module 121, an analytics module 122, a data acquisition module 123, a data security module 124, and an operations module 125, but the embodiments are not limited thereto. Each of the modules includes or uses a dedicated circuit, or instructions for operating a general purpose processor circuit, to perform the respective functions. In an example, the modules 121-125 are communicatively coupled in the cloud computing system 120 such that information from one module can be shared with another. In an example, the modules 121-125 are co-located at a designated datacenter or other facility, or the modules 121-125 can be distributed across multiple different locations.

**[0030]** An interface device 140 (*e.g.*, user device, workstation, tablet, laptop, appliance, kiosk, and the like) can be configured for data communication with one or more of the machine module 110, the gateway 105, and the cloud computing system 120. The interface device 140 can be used to monitor or control one or more assets. As another example, the interface device 140 may be used to develop and upload applications to the cloud computing system 120. As another example, the interface device 140 may be used to access analytical applications deployed on the cloud computing system 120. In an example, information about the asset community may be presented to an operator at the interface device 140. The information about the asset community may include information from the machine module 110, information from the cloud computing system 120, and the like. The interface device 140 can include options for optimizing one or more members of the asset community based on analytics performed at the cloud computing system 120.

**[0031]** The example of FIG. 1 includes the asset community with multiple wind turbine assets, including the wind turbine 101. However, it should be understood that wind turbines are merely used in this example as a non-limiting example of a type of asset that can be a part of, or in data communication with, the first AMP 100. Examples of other assets include gas turbines, steam turbines, heat recovery steam generators, balance of plant, healthcare machines and equipment, aircraft, locomotives, oil rigs, manufacturing machines and

equipment, textile processing machines, chemical processing machines, mining equipment, and the like.

**[0032]** FIG. 1 further includes the device gateway 105 configured to couple the asset community to the cloud computing system 120. The device gateway 105 can further couple the cloud computing system 120 to one or more other assets or asset communities, to the enterprise computing system 130, or to one or more other devices. The AMP 100 thus represents a scalable industrial solution that extends from a physical or virtual asset (*e.g.*, the wind turbine 101) to a remote cloud computing system 120. The cloud computing system 120 optionally includes a local, system, enterprise, or global computing infrastructure that can be optimized for industrial data workloads, secure data communication, and compliance with regulatory requirements.

**[0033]** The cloud computing system 120 can include the operations module 125. The operations module 125 can include services that developers can use to build or test Industrial Internet applications, and the operations module 125 can include services to implement Industrial Internet applications, such as in coordination with one or more other AMP modules. In an example, the operations module 125 includes a microservices marketplace where developers can publish their services and/or retrieve services from third parties. In addition, the operations module 125 can include a development framework for communicating with various available services or modules. The development framework can offer developers a consistent look and feel and a contextual user experience in web or mobile applications. Developers can add and make accessible their applications (services, data, analytics, etc.) via the cloud computing system 120.

**[0034]** Information from an asset, about the asset, or sensed by an asset itself may be communicated from the asset to the data acquisition module 123 in the cloud computing system 120. In an example, an external sensor can be used to sense information about a function of an asset, or to sense information about an environment condition at or near an asset. The external sensor can be configured for data communication with the device gateway 105 and the data acquisition module 123, and the cloud computing system 120 can be configured to use the sensor information in its analysis of one or more assets, such as using the analytics module 122. Using a result from the analytics module 122, an operational model can optionally be updated, such as for subsequent use in optimizing the first wind

turbine 101 or one or more other assets, such as one or more assets in the same or different asset community. For example, information about the wind turbine 101 can be analyzed at the cloud computing system 120 to inform selection of an operating parameter for a remotely located second wind turbine that belongs to a different asset community.

**[0035]** The cloud computing system 120 may include a Software-Defined Infrastructure (SDI) that serves as an abstraction layer above any specified hardware, such as to enable a data center to evolve over time with minimal disruption to overlying applications. The SDI enables a shared infrastructure with policy-based provisioning to facilitate dynamic automation, and enables SLA mappings to underlying infrastructure. This configuration can be useful when an application requires an underlying hardware configuration. The provisioning management and pooling of resources can be done at a granular level, thus allowing optimal resource allocation. In addition, the asset cloud computing system 120 may be based on Cloud Foundry (CF), an open source PaaS that supports multiple developer frameworks and an ecosystem of application services. Cloud Foundry can make it faster and easier for application developers to build, test, deploy, and scale applications. Developers thus gain access to the vibrant CF ecosystem and an ever-growing library of CF services. Additionally, because it is open source, CF can be customized for IIoT workloads.

**[0036]** The cloud computing system 120 can include a data services module that can facilitate application development. For example, the data services module can enable developers to bring data into the cloud computing system 120 and to make such data available for various applications, such as applications that execute at the cloud, at a machine module, or at an asset or other location. In an example, the data services module can be configured to cleanse, merge, or map data before ultimately storing it in an appropriate data store, for example, at the cloud computing system 120. A special emphasis may be placed on time series data, as it is the data format that most sensors use.

**[0037]** Security can be a concern for data services that exchange data between the cloud computing system 120 and one or more assets or other components. Some options for securing data transmissions include using Virtual Private Networks (VPN) or an SSL/TLS model. In an example, the AMP 100 can support two-way TLS, such as between a machine module and the security module 124. In an example, two-way TLS may not be supported, and the security module 124 can treat client devices as OAuth users. For example, the security

module 124 can allow enrollment of an asset (or other device) as an OAuth client and transparently use OAuth access tokens to send data to protected endpoints.

**[0038]** The threat detection and neutralization system and method according to various embodiments may be implemented within the security module 124 stored on the cloud computing system 120, within the asset, within an intermediate device between the asset and a control system, and the like. As another example, the threat detection and neutralization method may also or instead be implemented elsewhere such as within an asset, within the cloud computing system 120, within another device within the system, and the like. Raw data may be provided to the cloud computing system 120 via the assets included in the asset community and accessed by applications deployed on the cloud computing system 120. During operation, an asset may transmit sensor data to the cloud computing system 120 and prior to the cloud computing system 120 storing the sensor data, the sensor data may be filtered using the threat detection and neutralization method described herein.

**[0039]** FIG. 2 illustrates a threat neutralizer 230 for filtering input signals in accordance with an example embodiment. The threat neutralizer 230 may be implemented within an asset or a device associated therewith, within a control system or a device associated with or coupled to the control system, within a cloud platform, within an intermediate device, and the like. There is no limitation on a location of the threat neutralizer 230. The threat neutralizer 230 may receive signals that have been sensed from or about an asset such as a gas turbine, a wind turbine, a locomotive, an aircraft, a healthcare machine, an industrial manufacturing machine, drilling machinery/equipment, mining machinery/equipment, and the like. The threat neutralizer receives at least one input signal 210 that includes time series data. The time series data may include measurements or other readings captured at various time intervals. Examples of time series data include speed, intensity, acceleration, weight, force, thrust, and the like.

**[0040]** When deployed and operating, assets may be attacked or otherwise compromised. As a result, the data received from an asset (*e.g.*, input signals 210) may be faulty or otherwise inaccurate. There are many different types of attacks that an asset system may be subjected to such as espionage attacks, denial of service attacks, deception attacks (e.g., covert/stealthy attack, zero dynamics attack, false data injection attack, replay attack, and the like, which are just a short sampling of potential threats that exist. Present information

technology and operational technology approaches do not adequately protect the system automatically. In contrast, as further described herein, the threat neutralizer 230 may automatically detect and neutralize the effects of an attack within an input signal 210 just as a human immune system is capable of detecting and neutralizing an infection or other disease.

**[0041]** As an example, the threat neutralizer 230 may be included in a general system such as a cyber-physical system, a software system, a bio-mechanical system, a network system, a communication system, and/or the like, which contains access to continuous stream of data in the form of time series signals. The time series signals may be generated from output sensor nodes (*e.g.*, physical and/or virtual sensors), actuator nodes (*e.g.*, hard and/or soft actuators generated from open or closed loop system), controller nodes (*e.g.*, controller node signals), reference nodes (*e.g.*, reference signals), and the like. In some examples, logical signals may also be considered as time series signals. A total number of signals that may be used for providing immunity to a system may be equal to a total number of nodes that exist in sensors, actuators, controllers and reference nodes, or it may be more or less nodes. Some or all combinations of these nodes can be used for monitoring and neutralization.

**[0042]** In the example of FIG. 2, the threat neutralizer 230 includes multiple computational blocks including a feature transform 231, a Boundary and Performance constrained Resilient Estimator (BPRE) 233, and an inverse feature transform 235, which are further described herein. Using a dynamic or a static model obtained through system identification or other known modeling methods, the BPRE 233 is designed to compute an estimate of the true value of the feature vector under adversarial attacks. The computational elements inside the threat neutralizer 230 may thus act as a filter to remove attack signatures present in each node. As a result, an output signal 240 (or neutralized signals) may contain true estimates of signals at these nodes for use with the rest of the system for operational normalcy. In this example, y corresponds to a sensed signal, u corresponds to an actuator signal, c corresponds to a control node signal, and r corresponds to a reference signal. These are vectors of such signals. For convenience, underscores from each vector are removed.

**[0043]** *Feature Transform (Forward and Inverse):* For a given scenario, time series signals (e.g., ~45 seconds) may be captured and pre-processed. These signals may be processed by the feature transform 231 into feature extraction algorithms (e.g., principal component analysis) to reduce the dimensionality or order of the system. For example, time

series data from sensor nodes may be converted into feature vectors. The procedure is same for all other signals. When principal components are used as features as in the current implementation, weights become the features in reduced dimensions within the feature space. In the pre-processing step performed by the feature transform 231, temporal normalization of node data may be performed. As an example, when principal components are used as features, the normalized output may be expressed as a weighted linear combination of basis functions. In this feature extraction algorithm, weights are considered as features for y. The forward and inverse feature transforms can be written as:

**[0044]** Forward feature transform:

$$y = y_0 + \sum_{j=1}^{M} w_j \psi_j$$

$$w_j = \psi_j^T (y - y_0)$$

**[0045]** Inverse feature transform:

$$\hat{y} = y_0 + \sum_{j=1}^{M} \hat{w}_j \psi_j$$

Equation 1

**[0046]** Because of the orthonormal nature of the basis vectors, each feature may be computed during forward transform using Equation 1 shown above. In this example, $y_0$ = average sensor output, $w_j$ = *jth weight* (*i.e.*, PCA feature), and $\psi_j$ = jth basis vector. $\hat{w}_j$ = *jth weight* (true feature). $\hat{y}$ is the estimated sensor output in original time space, the true signal for normalcy under attack which is obtained by estimating true feature from boundary and performance constrained resilient estimator and solving the inverse feature transform equation in Equation 1, unit 235. The basis vectors can be obtained using ensemble data set from the system collected over time (i.e., historical data) or with perturbations on a virtual model (*e.g.*, digital twin) of the asset.

**[0047]** *System Identification (Dynamic Modeling):* Under normal operation, features are extracted from overlapping batch of a time series data by the Feature Transform unit 231. The process may be continued by the Feature Transform unit 231 over each overlapping batch

resulting in a new time series of feature evolution in the feature space. Then, the feature time series are used for performing system identification (i.e., dynamic modeling). A selected subset of the features may be used for dynamic modeling using state space system identification methods. For example, in a gas turbine system, out of 15 PCA features for each sensor node, 5 PCA features may be used.

**[0048]** The dynamic models are in the state space format. The dynamic modeler may use multivariate vector autoregressive model (VAR) for fitting dynamic models into feature time series data. If this approach is not adequate, another model may be used such as a tuned feature-based digital twin model, or a static model computed from feature evolution data or a priori to obtain various matrices representing state space form (i.e., system matrix, control matrix, output matrix and feedforward matrix; i.e., *{A, B, C}* matrices). In the example of a gas turbine system, the time-series evolution of the features may be modeled using the following state-space dynamics:

$$x(k+1) = Ax(k) + b$$
$$y(k) = Cx(k) + e(k) \quad \text{Equation 2}$$

**[0049]** In the example of Equation 2, *e(k)* is the attack component of the "measured" feature vector, y(k). Here, x(k) is the "true" feature vector to be determined. System matrix, *A*, control vector, *b*, and output matrix, *C*, are computed offline using experimental and simulated data and may be updated in real-time using real-time features obtained from actual sensor measurements.

**[0050]** *Boundary and Performance Constrained Resilient Estimator (BPRE):* The example embodiments attempt to estimate true states using a modified resilient estimator called Boundary and Performance Constrained Resilient Estimator (BPRE) 233. Using a model represented by the operator, $\Omega(A, b, C)$, such as for the example state-space model shown above in Equation 2, the BPRE 233 is designed to estimate the true value of the feature vector under adversarial attacks. The BPRE 233 is a resilient estimator that may solve the following optimization problem:

$$\underset{\mathbf{x}_k \in \mathbb{R}^n, k \in \{1,\ldots,T\}}{\text{minimize}} \quad \sum_{k=1}^{T} \|\mathbf{y}_k - \Omega(A,b,C)\mathbf{x}_k - \mathbf{y}_b\|_q + \tfrac{1}{2}\rho \sum_{j=1}^{T-1} \left\|\mathbf{x}_j^{\text{prev}} - \mathbf{x}_j\right\|_2^2$$

subject to

$$\begin{aligned} &\text{score}\,(\mathbf{x}_k, \mathbf{p}_1) \leq -\epsilon \\ &|f\,(\mathbf{x}_k, \mathbf{p}_2)\,| \leq \varepsilon \\ &\qquad\qquad \text{for } k \in \{1,\ldots,T-1\} \end{aligned}$$

**[0051]** With the parameters of the optimization:

$$\Omega(A,b,C) = \begin{bmatrix} \Psi_1 & & \\ & \ddots & \\ & & \Psi_n \end{bmatrix} \Phi \qquad \text{and} \qquad \mathbf{y}_b = \mathbf{y}_0 + \begin{bmatrix} \Psi_1 & & \\ & \ddots & \\ & & \Psi_n \end{bmatrix} \mathbf{w}_0$$

Equation 3

**[0052]** In this example, $\boldsymbol{y}_k$ is the sequence of measured/observed values, $\boldsymbol{x}_k$ is the feature vector in the current time step, $\boldsymbol{x}_k^{prev}$ is the feature vector in the previous time step, $\Omega(A,b,C)$ is an operator representing state space model or a transfer function from the feature space to the signal space. $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are respective parameter values for the boundary and invariance relationship. $q$ is used to describe optimization approach such as $l_1$ when $q=1$, $l_2$ when $q=2$ well-known in the optimization literature. The first constraint in the optimization may ensure that the final estimate falls within the safety envelop with some factor of safety given by $\epsilon$ which can be an arbitrary positive number, or something more intuitive like the distance of the farthest support vector from the multi-dimensional decision boundary. The BPRE 233 may compute a threat/attack boundary and also compute a performance constrained resilient estimator from the normal data values received historically from the asset. FIG. 3 illustrates conceptually an example of a boundary and performance estimation process for threat detection in accordance with an example embodiment. In the example of FIG. 3, a graphical view 300 of the boundary constraint is shown with inner margin boundary 306 specified by positive $\epsilon$. Negative $\epsilon$ represents outer margin boundary 308. Also, $\epsilon$=0 represents hard decision boundary 304. The "score" function in Equation 3 depends on a particular algorithm used to learn the safety envelop. For instance:

$$\text{score}(\mathbf{x}, \mathbf{p}) = \phi(\mathbf{x})^{\top} \mathbf{p}_{[1:n]} + p_0 \qquad \text{Equation 4}$$

**[0053]** Equation 4 defines the safety envelop obtained using a support vector machine (SVM) with kernel function $\phi(.)$. The second constraint is related to achieving normal performance by the system. Here, the ensuing solution may satisfy an implicit invariance relationship inherent to the system that was operating normally and is shown graphically in FIG. 3 by a thick curve 302 inside the normal operating space. This constraint is termed "performance constraint" and may be required during optimization. The value $\varepsilon$ is used to capture the uncertainty due to noise in the system. For example, in the case of PCA, the invariance relation is captured by the singular vectors corresponding to the smallest singular values of the feature covariance of data matrix. For example, let $Y = [\, y_1 \ \cdots \ y_T \,]$ be the data matrix constructed using ensemble data set from the system collected over time (i.e., historical data) or with perturbations on digital twin models of the asset with each row being the corresponding instantaneous time-series measurement. Then the invariance relationship is captured by the linear inequality $l \leq VY \leq u$, where $(l,u)$ are the minimum and maximum values in the matrix $VY$ and the columns of $V$ are the singular vectors of $Y^TY$ corresponding to the smallest singular values.

**[0054]** In FIG. 3, the feature vectors corresponding to normal operating signals 301a are not considered potential threats while the feature vectors positioned outside the boundary 304 are considered threats 301b or attack signals. These points outside the boundary may also be referred to as attack points 301b. These attack points may be due to cyberattacks moving the previously normal feature vectors represented by points anywhere near or on the normal performance region 302 to anywhere to the attack region designated by points 301b. Referring now to both FIGS. 2 and 3, if the feature vectors are outside the boundary 304, indicating an abnormal event 301b, the BPRE estimator 233 can find an estimate of the feature vector along performance constraints inside the boundary 304 that preserves the normal operation of the asset with respect to the control system or other device connected to the asset.

**[0055]** The optimization used in Equation 3 contains two weighted objectives. The first objective involves a $q$-norm ($q \in \{0, 1, 2, \ldots\}$) of the error between observed and modeled features. Although, $q = 0$ is ideal for this problem, it results in an NP-hard problem. Consequently, in order to have a well-behaved convex program, the value $q \geq 1$ may be used, where the particular value used in any case may depend on the desired error characteristics

for that case. For instance, the value $q = 1$ may provide the closest approximation to the ideal case but results in a mixed-norm problem while the value $q = 2$ is further from ideal but unifies the norm in the objective and the overall optimization problem is a subclass of the well-known second order cone program. The measured, as well as estimated "true" feature vectors may then be transformed into time series signal by inverting the feature transform.

**[0056]** An example of solving for the boundary and performance constraint shown above in Equation 3 is described as follows. It should be appreciated that the following equations are just one example, and many examples are possible. Often it will be useful to normalize sensor signals to some nominal operating condition temporally and spatially before performing feature transform. Consider, for example, temporal normalization of monitoring node data that may be performed as follows at every sample along a time axis:

$$y_N = \frac{(y_{nom} - y_{UN})}{\bar{y}_{nom}} \qquad \text{Equation 5}$$

**[0057]** Where $y_N$ is the normalized signal, $y_{UN}$ is the un-normalized signal and $y_{nom}$ is the sensor signal for nominal operating condition and $\bar{y}_{nom}$ is a temporal average of a time series signal for normal operating conditions. In this case, to obtain feature transform a temporal average is computed for a batch length (e.g., 45 seconds). De-normalization involves obtaining the sensor signals in original un-normalized space by inverting the equation above.

**[0058]** Feature transform involves transforming signals from a Hilbert space (H) to a finite dimensional Euclidean space. In order to facilitate easy inversion, this transformation may be done via orthogonal functional basis set. For the signal $y \in \mathbb{H}$, the feature, $w \in \mathbb{R}^n$, is given by:

$$w(t) = \arg\min_{x \in \mathbb{R}^n} \int_{t-T}^{t} \left( y(\tau) - \sum_{j=1}^{n} x_j \Psi_j(\eta_R(\tau)) \right)^2 d\tau \quad (2) \qquad \text{Equation 6}$$

where $\eta_k : [t-T,\ t] \mapsto [-1,\ 1]$ and $\Psi_j \in \mathbb{H}^{[-1,\ 1]}, j = 1, \ldots, n$ are orthogonal basis functions. Hence, the solution to the above optimization problem is given by:

$$w_j = \frac{< \Psi_j, y >}{< \Psi_j, \Psi_j >} \qquad \text{Equation 7}$$

**[0059]** The inverse feature transform is then given by:

$$\hat{y}(t) = \sum_{j=1}^{n} w_j(t) \Psi_j(\eta_k(t)). \qquad \text{Equation 8}$$

**[0060]** In general, there are many available choices for the orthogonal basis; from trigonometric, to orthogonal polynomials like Legendre, Laguerre, Hermite and Chebyshev. However, the signals considered in this paper are sampled at specific intervals and for each signal, there is a design of experiment done to collect a finite data set consisting of points that characterizes the operating space of that signal. Consequently, PCA is used to obtain an orthonormal set which serves as the functional basis for this application. The feature transform for the $i^{th}$ monitoring nodes is then obtained as:

$$yK_i = y0_i + \Psi_i w_i \qquad \text{Equation 9}$$

**[0061]** where $y0_i \in \mathbb{R}^{n_T}$ is the ensemble average for the signal, $\Psi_i \in \mathbb{R}^{n_T \times n}$ is a matrix whose columns are the basis vectors, $w_i \in \mathbb{R}^{n_T}$ is the corresponding feature, and $n_T$ is the number of sample points in the interval $[t-T,\ t]$. It is noteworthy that the features are obtained from a sliding window batch of data as elucidated in (Equation 6). Hence, the features themselves are time series signals.

**[0062]** For the boundary and performance constrained resilient estimator, a labeled data set may be generated that consists of attack and normal scenarios. The data set may be generated from realistic runs on high fidelity industry standard models. The different normal and attack scenarios may be selected by the system domain experts. First, a decision boundary that separates attack from safe is constructed using various kernel methods. The BPRE may combine concepts from compressed sensing with the binary classifier decision function to estimate the "true" features of the time series

signals. The objective is to find the features that best explains the measurement and are also classified as "normal" by the decision function. Normal features represent safe operation with reasonable operational performance. As a result, the approach successfully fuses domain knowledge with compressed sensing. Another merit of this approach is that the domain knowledge, in the form of the decision function, can be updated online whenever new knowledge becomes available.

**[0063]** In this example, to perform the boundary decision function, first, feature transform is done independently for each monitoring node. These nodal features are referred to as local features. The local features from all monitoring nodes are then stacked into one big vector, on which further dimensionality reduction is carried out to obtain what is referred to as global features. The local features capture time-related signatures from the signals from each monitoring nodes while the global features capture cross-relational signatures among monitoring nodes. The two-level feature transform is described by the following:

$$y_N = y_0 + \Psi w \quad \text{Equation 10}$$

$$w = w_0 + \Phi g \quad \text{Equation 11}$$

**[0064]** where

$$y_N = \begin{bmatrix} y_{N_1} \\ \vdots \\ y_{N_{n_s}} \end{bmatrix},$$

$$\Psi = \begin{bmatrix} \Psi_1 & & \\ & \ddots & \\ & & \Psi_{n_s} \end{bmatrix},$$

$$w = \begin{bmatrix} w_1 \\ \vdots \\ w_{n_s} \end{bmatrix}$$

with $n_s$ the number of monitoring nodes, $w_i \in \mathbb{R}^{n_L}$ the ith local feature vector, $\Psi_i \in \mathbb{R}^{n_T \times n_L}$ the ith local base matrix, $g \in \mathbb{R}^{n_G}$ the global features vector,

$\Phi \in \mathbb{R}^{n_T \times n_G}$ the global basis matrix, and $n_L \& n_G$ the numbers of local and global features respectively. The 0th terms are ensemble averages of corresponding quantities. After combining the equations above, the conversion from global feature space to signal space is given by:

$$y_N = \underbrace{y_0 + \Psi w_0}_{y_b} + \underbrace{\Psi \Phi}_{\Omega} g, \quad \text{Equation 12}$$

The parameters $y_0$, $w_0$, $\Psi$ and $\Phi$ are obtained offline and can be updated independently of the resilient estimator.

**[0065]** Next, using the attack/safe labels of the data set together with global features, a kernel-based binary classifier is constructed. The classification decision function (or score) is of the form

$$s(g) = \sum_{i=1}^{m} q_i \kappa(x_i, g) + b_s \quad \text{Equation 13}$$

**[0066]** where $\kappa(\cdot,\cdot)$ is the kernel function which computes dot products in the high dimensional RKHS, $x_i$ are the support vectors, $q_i$ and $b_s$ are the associated weights and bias. The decision function based on the score above may be:

$$\text{Status}(g) = \begin{cases} \text{attack} & \text{if} \quad s(g) > 0 \\ \text{normal} & \text{if} \quad s(g) < 0 \end{cases} \quad \text{Equation 14}$$

**[0067]** For the resilient estimator, suppose that the measured signal is given by:

$$y_N = y_N^* + y_a, \quad \text{Equation 15}$$

**[0068]** where $y_a$ is the malicious addition due to an adversarial agent, the objective is to estimate the true signal $y_N^*$. The attack signal $y_a$ is not limited to a constant bias and could be any dynamically changing and intelligently designed malicious signal. It is assumed that the attack is sparse. In other words, the ratio of compromised nodes is such that there is enough redundancies in the system to facilitate reconstruction. The exact level of redundancy needed to ensure accurate reconstruction of compromised signal is a topic of future work. Equivalently, Equation 15 may be written as:

$$\mathbf{w} = \mathbf{w}^* + \mathbf{w}_0. \qquad \text{Equation 16}$$

**[0069]** Consequently, the resilient estimator may be cast as the following optimization problem, as one example solution:

$$\begin{aligned} &\text{Minimize} && \|\mathbf{w}_k - \mathbf{w}_0 - \Phi\mathbf{g}\|_{\ell_0} \\ &\text{Subject to} && \\ & && s(\mathbf{g}) < 0 \end{aligned} \qquad \text{Equation 17}$$

**[0070]** In the example of Equation 17, the subscript k is used to indicate the time dependence of the local features. The optimization problem above is NP-hard mainly due to the index minimization objective. This type of objective has been shown to be able to identify attack signals. While, the constraint $s(\mathbf{g}) < 0$ improves the overall estimation problem by appending domain-level knowledge, it introduces additional complexity because it is in general non-convex. In what follows, a new problem composed of a convex relaxation of the objective and a local approximation of the kernel decision function is described.

**[0071]** Most approaches in literature suggest replacing the index minimization objective with a $\ell_1$-norm. Here, if the matrix $\Phi$ satisfies certain conditions, this convexification is loss-less. Thus, the objective is replaced with $\|\mathbf{w}_k - \mathbf{w}_0 - \Phi\mathbf{g}\|_{\ell_1}$.

**[0072]** Next, the local approximation of the kernel-decision function is described. The normal constraint $s(g) < 0$ is equivalent to:

$$-\sum_{q_i<0} |q_i|\kappa(\mathbf{x}_i, \mathbf{g}) < -\sum_{q_j>0} |q_j|\kappa(\mathbf{x}_j, \mathbf{g}). \qquad \text{Equation 18}$$

**[0073]** In this example, since sgn ($q_i$) is really the label on the ith support vector, the constraint separates the support vectors into two sets and requires that the solution be closer to the "normal" set, where closeness is measured by the dot product in RKHS given by the kernel function. Subsequently, it is assumed that the kernel function is isotropic, for example, $\kappa(\mathbf{x}_i, \mathbf{g}) = f_\kappa(\|\mathbf{x}_i - \mathbf{g}\|_Q^2),\ Q \succeq 0.$ Here, the function $f_\kappa : \mathbb{R}_+ \mapsto \mathbb{R}_+$ is continuous and monotonically decreasing. It is clear at this

point that the non-convexity is due to the RHS of Equation 18. Hence, the following approximate may be considered:

$$\hat{s}(\mathbf{g}) \triangleq \sum_{q_i<0} \|\mathbf{x}_i - \mathbf{g}\|_{Q_i}^2 + \sum_{q_j>0} \mathbf{q}_j^T (\mathbf{x}_j - \mathbf{g}),$$

Equation 19

**[0074]** where $Q_i$ and $q_i$ are parameters that are determined locally at each time step. The convex LHS is approximated with quadratic and the non-convex RHS is approximated with linear function. A local binary classification problem is then solved with two requirements: preserve the label of the current measurement and reproduce the labels of the support vectors as much as possible with more emphasis placed on the ones closest to the current operating point.

**[0075]** Therefore, the local classification problem considered may be:

Minimize:

$$\frac{1}{2}\left(\sum_{q_i<0} \|Q_i\|_F^2 + \sum_{q_j>0} \|\mathbf{q}_j\|^2 + \gamma \sum_{i=1}^{m} d(\mathbf{x}_i, \mathbf{g}_{k-1}) e_i^2\right)$$

Subject to:

$$s_k \hat{s}(\mathbf{g}_k) \geq 1$$
$$s_i \hat{s}(\mathbf{x}_i) - 1 + e_i = 0,\ i = 1, \ldots, m,$$

Equation 20

**[0076]** where,

$\mathbf{g}_{k-1}$ is estimated global feature from the previous step. It is used as an indication of the current operating condition,

$d(\mathbf{x}_i, \mathbf{g}_{k-1})$ is a semi-positive valued function used to penalize the importance of the ith support vector.

$(\mathbf{g}_k, s_k)$ are the respective global features and SVM score corresponding to the current measurement

$(s_i, e_i)$ are the respective SVM score and degree of misclassification of the ith support vector.

[0077] The first two terms in the objective corresponds to Vapnik-Chervonenkis (VC) dimension which helps to improve the generalization of the decision function. For simplicity and faster computation, the $Q_i$ s are restricted to a class of diagonal matrices ($Q_i = \mathrm{diag}(q_i)$). As a result, the optimization problem may reduce to:

$$\begin{aligned} &\text{Minimize:} \\ &\quad \tfrac{1}{2}\left(\|\mathbf{q}\|^2 + \gamma\|\mathbf{e}\|_D^2\right) \\ &\text{Subject to:} \\ &\quad 1 - s_k K_k \mathbf{q} \le 0 \\ &\quad SK\mathbf{q} - \underline{\mathbf{1}} + \mathbf{e} = 0 \end{aligned} \qquad \text{Equation 21}$$

$$\begin{aligned} &\text{where } \mathbf{q} = [\mathbf{q}_1^\top, \ldots, \mathbf{q}_m^\top]^\top,\ \mathbf{e} = [e_1, \ldots, e_m]^\top, \\ &D = \mathrm{diag}([d(\mathbf{x}_1, \mathbf{g}_{k-1}), \ldots, d(\mathbf{x}_m, \mathbf{g}_{k-1})]), \\ &K_k = [(\mathbf{x}_1 - \mathbf{g}_k)^\top \mathrm{diag}(\mathbf{x}_1 - \mathbf{g}_k), \ldots, (\mathbf{x}_m - \mathbf{g}_k)^\top \mathrm{diag}(\mathbf{x}_m - \mathbf{g}_k)], \\ &S = \mathrm{diag}([s_1, \ldots, s_m]), \text{ and} \end{aligned}$$

$$K = \begin{bmatrix} \underline{\mathbf{0}}^\top & \cdots & (\mathbf{x}_1 - \mathbf{x}_m)^\top \mathrm{diag}(\mathbf{x}_1 - \mathbf{x}_m) \\ \vdots & \ddots & \vdots \\ (\mathbf{x}_m - \mathbf{x}_1)^\top \mathrm{diag}(\mathbf{x}_m - \mathbf{x}_1) & \cdots & \underline{\mathbf{0}}^\top \end{bmatrix}$$

[0078] This is a simple linearly-constrained quadratic program and its dual:

$$\underset{\alpha \ge 0}{\text{Minimize}} \frac{1}{2}a\alpha^2 - b\alpha, \qquad \text{Equation 22}$$

[0079] where

$$\begin{aligned} H &= I + \gamma K^\top D K \\ a &= K_k^\top H^{-1} K_k \\ b &= 1 - \gamma s_k K_k^\top H^{-1} K^\top S D \underline{\mathbf{1}}, \end{aligned}$$

has the closed-form solution

$$\alpha^* = \max\left\{\frac{b}{a}, 0\right\}.$$

Equation 23

**[0080]** Hence, the parameters of $\delta(\cdot)$ as given by the solution of the primal problem in Equation 21 are:

$$\mathbf{q}^* = H^{-1}\left(\gamma K^\top S D \mathbf{1} + \alpha^* s_k K_k\right).$$

Equation 24

$$H^{-1} = V\left[\begin{array}{c|c} (I + \gamma\Sigma_1\Lambda\Sigma_1)^{-1} & \\ \hline & I \end{array}\right]V^\top$$

Equation 25

**[0081]** In this example, the inverse shown in Equation 25, only involves online computation of the inverse of a smaller diagonal matrix.

**[0082]** The relaxed and local approximate optimization problem for the resilient problem is:

$$\begin{aligned} &\text{Minimize} && \|\mathbf{w}_k - \mathbf{w}_0 - \Phi\mathbf{g}\|_{\ell_2} \\ &\text{Subject to} && \\ & && \mathbf{g}^\top P\mathbf{g} + \mathbf{p}^\top\mathbf{g} + r \le 0, \end{aligned}$$

Equation 26

**[0083]** where

$$\begin{aligned} P &= \sum_{q_i<0} Q_i \\ \mathbf{p} &= -2\sum_{q_i<0} Q_i\mathbf{x}_i - \sum_{q_j>0}\mathbf{q}_j \\ r &= \sum_{q_i<0}\|\mathbf{x}_i\|^2_{Q_i} + \sum_{q_j>0}\mathbf{x}_i^\top\mathbf{q}_j. \end{aligned}$$

Equation 27

**[0084]** At this point, the matrix P can in general be indefinite – since there is nothing saying otherwise. One could explicitly impose that constraint in the optimization problem in Equation 21, in which case the ability to derive a closed-form solution is lost and one resorts to iterative solution for the local approximation problem. This may be a good option if there is enough processing speed to handle iterative solutions in both the local approximation and resilient estimation problems. On the other hand, one could allow indefinite P in favor of a

closed-form solution for the local approximation problem – as done already. If that is the case, the optimization problem in Equation 26 can be further relaxed as:

$$\begin{aligned} \text{Minimize} \quad & \|\mathbf{w}_k - \mathbf{w}_0 - \Phi\mathbf{g}\|_{\ell_1} \\ \text{Subject to} \quad & \\ & \operatorname{trace}(GP) + \mathbf{p}^\top\mathbf{g} + r < 0 \\ & \begin{bmatrix} G & \mathbf{g} \\ \mathbf{g}^\top & 1 \end{bmatrix} \succeq 0. \end{aligned}$$

Equation 28

**[0085]** The semi-definite program above can be solved sufficiently fast using interior-point methods. As a way of simple summary, the overall resilient estimator algorithm is described in Algorithm 1.

**Algorithm 1** Boundary and Performance Constrained Resilient Estimator

**procedure** BPRE($\mathbf{y}_k, \{\mathbf{x}_i, q_i\}, K$)

*Normalize*:

$\mathbf{y}_{N_k} \leftarrow (5)$

*Local Approximation*:

$\mathbf{w}_k \leftarrow (10)$

$s_k \leftarrow (13)$

$\mathbf{q}^* \leftarrow H^{-1}\left(\gamma K^\top SD\mathbf{1} + \alpha^* s_k K_k\right)$ (24)

*Resilient Global Feature Vector*:

$P, \mathbf{p}, r \leftarrow (27)$

$\mathbf{g}_k^* \leftarrow \arg\min (28)$

*Output*:

$\mathbf{y}_{N_k}^* \leftarrow \mathbf{y}_b + \Omega\mathbf{g}_k^*$ (5)

**return** $\mathbf{y}_k \leftarrow \mathbf{y}_{\text{nom}} - \hat{\mathbf{y}}_{\text{nom}}\mathbf{y}_{N_k}^*$

**[0086]** The BPRE 233 forces the attacked sensor signals back into the normalcy boundary 304 which also translates to normal operation of the original system. As a result, neutralized signals 240 are output from the threat neutralizer 230. This inversion approach assumes that the feature transform algorithm is well defined and has only trivial kernels (at least injective). The MMMD feature discovery framework may be used to identify knowledge-based, shallow and/or deep learning features. Note that the MMMD feature discovery framework may be associated with feature engineering (*e.g.*, associated with analysis such as batch selection,

basis vector computation, feature extraction, dimensionality reduction, *etc.*) and engineered and dynamic system feature vectors. Knowledge-based feature engineering may use domain or engineering knowledge of the system to create features from different sensor measurements. These features might simply be statistical descriptors (*e.g.*, maximum, minimum, mean, variance, different orders of moments, *etc.*) calculated over a window of a time-series signal and its corresponding Fast Fourier Transformation ("FFT") spectrum as well. The knowledge-based features might also utilize analysis of the system, such as basis vector decomposition, state estimation, observability matrices, topology matrices, system plant matrices, frequency domain features and system poles and zeros. These analyses may represent a characterization of the system through steady-state, transient, and small signal behaviors. The feature extraction process may be further associated with a shallow feature learning technique, such as unsupervised learning, k-means clustering, manifold learning, non-linear embedding, an isomap method, LLE, PCA as described above in one example, Non-linear PCA, ICA (Independent Component Analysis), neural networks, a SOM (Self-Organizing Map) method, genetic programming, and/or sparse coding. A numerical inversion algorithm may be used to convert true features to their equivalent time series form when the features are not well defined as in the PCA case described above. Furthermore, smoothing of the time series signal from previous batches may be required while performing continuous updates.

**[0087]** FIG. 4 illustrates a method 400 for neutralizing cyber-based threats in accordance with example embodiments. For example, the method 400 may be performed by the neutralizer 230 shown in FIG. 2. Referring to FIG. 4, in 410 the method includes receiving input signals from an asset including time series data associated with the asset and transforming the input signals into feature values in a feature space. The asset may be a physical asset such as machinery or equipment in the field of healthcare, industry, manufacturing, transportation, energy, etc. The input signals may include signals related to a physical asset included within an Industrial Internet of Things (IIoT). Examples of input signals include measured parameters from the asset, control signals, reference signals, and the like. In response to receiving the input signals, the method may transform the input signals from the time domain into the frequency domain. For example, the input signals may be transformed into frequency values such as frequency vectors. The feature vectors may have

at least two dimensions (*e.g.*, magnitude, location, time, etc.) and be graphed in a multi-dimensional feature space.

**[0088]** In 420, the method includes detecting one or more abnormal feature values in the feature space based on a predetermined normalcy boundary associated with the asset. The normalcy boundary may be determined based on historical data (*i.e.*, normal and/or abnormal) data previously received from the asset. In some embodiments, the normalcy boundary may be continually updated based on newly received sensor data from the asset. The abnormal values may be identified based on the position of feature vector in the feature space with respect to the normalcy boundary, also referred to as an attack boundary. In this case, the values plotted inside the boundary may be determined as normal while the values plotted outside the boundary may be determined as abnormal or threats. Here, the detected abnormal feature values may correspond to an attack signal associated with the asset.

**[0089]** In 430, the method includes determining estimated true feature values (i.e., true feature vector) for each abnormal feature value (i.e., abnormal feature vector). For example, an estimated true feature values in a true feature vector may be determined by masking a portion of the feature vector associated with an attack signal such that the resulting feature vector with feature values is located within the predetermined normalcy boundary in the feature space. In this example, the true feature vector may be an estimate of the true features of normal signal with the attack signal removed. In some examples, prior to determining the true value for a feature vector, the method may include determining that an observability condition (Not shown in Figures) associated with the feature space is satisfied prior to estimating the true value. Examples of estimating the true value in the true feature vector are described with reference to FIGS. 2 and 3. In 440, the method includes performing an inverse transform of each estimated true feature vector to generate neutralized signals comprising time series data and outputting the neutralized signals, in 450. For example, the neutralized signals may be output to a control system or another device associated with the asset.

**[0090]** FIG. 5 illustrates an example 500 of the simulated performance of a boundary and performance constrained resilient estimator as described above. In this example, 45 seconds of a time series signal is shown for six sensor nodes (sensors 1 to 6). The top (red) signal corresponds to output of sensors during an attack when sensor 510 was subjected to a bias

attack. The middle (green) signal of each graph corresponds to the normal operational condition prior to the attack while the bottom (blue) signal of each graph corresponds to neutralized signals obtained after processing through techniques described in the example embodiments.

**[0091]** For this example simulation, the system identification in the feature space described by Equation 2 is not used. After performing optimization true global feature vectors are obtained. They were then transformed to time series using inverse feature transform. Clearly, FIG 5 shows, neutralized signals generated by BPRE algorithm are inside the attack boundary and is very close to the operational normalcy. FIG. 5 illustrates an example of results of the BPRE algorithm in the presence of a sensor attack.

**[0092]** FIG. 6 illustrates a computing device 600 for neutralizing cyber-based threats in accordance with an example embodiment. For example, the computing device 600 may correspond to the neutralizer 230 shown in FIG. 2, or another device. In some cases, the computing device 600 may be implemented within a control system, a cloud environment, and/or the like. Also, the computing device 600 may perform the method 400 of FIG. 4. Referring to FIG. 6, the computing device 600 includes a receiver 610, a processor 620, an output 630, and a storage device 640. Although not shown in FIG. 6, the device 600 may include other components such as a display, an input unit, and the like. The receiver 610 and the output 630 may include a signal receiver/transmitter, a transceiver, a network interface, and the like, and may transmit and receive data over a network such as the Internet, a private network, a public network, and the like, and may transmit and receive data via a wired connection. The receiver 610 and the transmitter 630 may be a wireless communicator, a wired communicator, or a combination thereof. The processor 620 may include one or more processing devices each including one or more processing cores. In some examples, the processor 620 is a multicore processor or a plurality of multicore processors. Also, the processor 620 may be fixed or it may be reconfigurable. The storage device 640 is not limited to any particular storage device and may include any known memory device such as RAM, ROM, hard disk, and the like.

**[0093]** According to various embodiments, the receiver 610 may receive input signals from an asset comprising time series data associated with the asset. The input signals may include time series data sensed by sensors associated with an Industrial Internet of Things

(IIoT). For example, the asset may be a physical asset such as a wind turbine, a gas turbine, an oil rig, mining equipment, an aircraft, a locomotive, a manufacturing system, a healthcare system, a chemical processing system, and the like. The time series data may be sensed by one or more sensors attached to or fixed about the asset. As another example, the time series data may include virtual data sensed from a digital twin representing the physical asset in virtual space. Examples of time series data include sensed minimums, maximums, ranges, magnitudes, direction, location, and the like. The processor 620 may transform the input signals into feature values in a feature space. For example, the processor 620 may convert the time series signals in a time domain into feature vectors in a feature domain. The feature vectors may represent multiple features of an object using a multi-dimensional vector. In some embodiments, the feature vector includes two dimensions, but is not limited thereto.

**[0094]** The processor 620 may detect one or more abnormal feature values in the feature space based on a predetermined normalcy boundary associated with the asset. For example, the processor 620 may define a normalcy boundary based on historical data received from the asset. The normalcy boundary may be generated based on a typical value or pattern of the input signal from the asset that is received. In some embodiments, the processor 620 may update the normalcy boundary based on currently sensed data associated with the asset. Accordingly, the computing device 600 may correspond to an intelligent system that continues to learn from incoming sensor data.

**[0095]** The abnormal feature values may be feature vectors that fall outside of the normalcy boundary. In other words, the abnormal feature values may have values that correspond to the normal behavior pattern previously established for the input signals. In this example, the processor 620 may determine that the abnormal feature values correspond to an attack or a threat that has infiltrated the asset. In order to neutralize the attack, the processor 620 may determine an estimated true value for each abnormal feature value. For example, the processor 620 may determine an estimated true value for a feature value by masking a portion of the feature value associated with an attack signal such that the resulting feature value is located within the predetermined normalcy boundary in the feature space. In some embodiments, the processor 620 may further determine that an observability condition associated with the abnormal feature values is satisfied prior to estimating the true value for the abnormal feature values. The processor 620 may perform an inverse transform of each

estimated true value to generate neutralized signals comprising time series data and control the output 630 to output the neutralized signals to a device or software such as a control system, a cloud, another component, and the like.

**[0096]** The example embodiments are aimed at going beyond detecting attacks/anomalies. The threat neutralizer described herein provides an automated accommodation when cyber disruptions, such as stealthy attacks occur to an industrial system. Attacks have become very sophisticated. According to cybersecurity experts of control systems, sophisticated malware and viruses may sitting inside the industrial control system software remained undetected for as many as 443 days in 2016, whereas the same for IT systems 146 days in 2015. They are challenging to detect because they are multi-prong and current fault detection systems are unable to find them. Also, fault detection systems do not estimate true states for operational normalcy. The boundary and performance constrained resilient estimator included within the threat neutralizer is able to automatically find true estimates of signals and hence provide adjustments to system operations in response to ongoing attacks while maintaining the operations expected for performance. The threat neutralizer is able to self-detect attacks and then nullify the attack signatures by finding true value of signals, mechanistically the same way as human immune system does to destroy pathogens. Thus, learning from the human body is accomplished here.

**[0097]** As will be appreciated based on the foregoing specification, the above-described examples of the disclosure may be implemented using computer programming or engineering techniques including computer software, firmware, hardware or any combination or subset thereof. Any such resulting program, having computer-readable code, may be embodied or provided within one or more non transitory computer-readable media, thereby making a computer program product, i.e., an article of manufacture, according to the discussed examples of the disclosure. For example, the non-transitory computer-readable media may be, but is not limited to, a fixed drive, diskette, optical disk, magnetic tape, flash memory, semiconductor memory such as read-only memory (ROM), and/or any transmitting/receiving medium such as the Internet, cloud storage, the internet of things, or other communication network or link. The article of manufacture containing the computer code may be made and/or used by executing the code directly from one medium, by copying the code from one medium to another medium, or by transmitting the code over a network.

[0098] The computer programs (also referred to as programs, software, software applications, "apps", or code) may include machine instructions for a programmable processor, and may be implemented in a high-level procedural and/or object-oriented programming language, and/or in assembly/machine language. As used herein, the terms "machine-readable medium" and "computer-readable medium" refer to any computer program product, apparatus, cloud storage, internet of things, and/or device (*e.g.*, magnetic discs, optical disks, memory, programmable logic devices (PLDs)) used to provide machine instructions and/or data to a programmable processor, including a machine-readable medium that receives machine instructions as a machine-readable signal. The "machine-readable medium" and "computer-readable medium," however, do not include transitory signals. The term "machine-readable signal" refers to any signal that may be used to provide machine instructions and/or any other kind of data to a programmable processor.

[0099] The above descriptions and illustrations of processes herein should not be considered to imply a fixed order for performing the process steps. Rather, the process steps may be performed in any order that is practicable, including simultaneous performance of at least some steps. Although the disclosure has been described in connection with specific examples, it should be understood that various changes, substitutions, and alterations apparent to those skilled in the art can be made to the disclosed embodiments without departing from the spirit and scope of the disclosure as set forth in the appended claims.

WHAT IS CLAIMED IS:

1. A method comprising:

receiving input signals from an asset comprising time series data associated with the asset and transforming the input signals into feature values in a feature space;

detecting one or more abnormal feature values in the feature space based on at least one predetermined normalcy boundary associated with the asset, and determining an estimated true value for each abnormal feature value; and

performing an inverse transform of each estimated true value to generate neutralized signals comprising time series data and outputting the neutralized signals.

2. The method of claim 1, wherein the input signals correspond to time series data sensed by sensors associated with an Industrial Internet of Things (IIoT).

3. The method of claim 1, wherein the transforming comprises transforming the time series data from the asset into feature vectors.

4. The method of claim 1, wherein transforming the signals into feature values and performing an inverse transform comprises an identity transform.

5. The method of claim 1, further comprising determining the normalcy boundary based on one or more of a historical data associated with the asset, and a data set obtained from input-output design of experiments on a digital twin of the asset.

6. The method of claim 5, further comprising updating the normalcy boundary based on currently sensed data associated with the asset.

7. The method of claim 1, wherein the detected abnormal feature values correspond to an attack signal associated with the asset.

8. The method of claim 1, wherein the determining of an estimated true value for a feature value comprises masking a portion of the feature value associated with an attack signal such that the resulting feature value is located within the predetermined normalcy boundary in the feature space.

9. The method of claim 1, wherein the input signals are received from the asset and the neutralized signals are output to a control system of the asset.

10. The method of claim 1, wherein the determining comprises determining a model associated with the feature values during real-time prior to estimating the true value for the abnormal value.

11. A computing system comprising:

a receiver configured to receive input signals from an asset comprising time series data associated with the asset;

a processor configured to transform the input signals into feature values in a feature space, detect one or more abnormal feature values in the feature space based on at least one predetermined normalcy boundary associated with the asset, determine an estimated true value for each abnormal feature value, and perform an inverse transform of each estimated true value to generate neutralized signals comprising time series data; and

an output configured to output the neutralized signals.

12. The computing system of claim 11, wherein the input signals correspond to time series data sensed by sensors associated with an Industrial Internet of Things (IIoT).

13. The computing system of claim 11, wherein the processor is configured to transform the time series data from the asset into feature vectors.

14. The computing system of claim 11, wherein the transforming the signals into feature values and performing an inverse transform comprises an identity transform.

15. The computing system of claim 11, wherein the processor is further configured to determine the normalcy boundary based on one or more of a historical data associated with the asset, and a data set obtained from input-output design of experiments on a digital twin of the asset.

16. The computing system of claim 15, wherein the processor is further configured to update the normalcy boundary based on currently sensed data associated with the asset.

17. The computing system of claim 11, wherein the detected abnormal feature values correspond to an attack signal associated with the asset.

18. The computing system of claim 11, wherein the processor is configured to determine an estimated true value for a feature value by masking a portion of the feature

value associated with an attack signal such that the resulting feature value is located within the predetermined normalcy boundary in the feature space.

19. The computing system of claim 11, wherein the receiver receives the input signals from the asset and the output outputs the neutralized signals to a control system of the asset.

20. A non-transitory computer readable storage medium having stored therein instructions that when executed cause a processor to perform a method comprising:

receiving input signals from an asset comprising time series data associated with the asset and transforming the input signals into feature values in a feature space;

detecting one or more abnormal feature values in the feature space based on at least one predetermined normalcy boundary associated with the asset, and determining an estimated true value for each abnormal feature value; and

performing an inverse transform of each estimated true value to generate neutralized signals comprising time series data and outputting the neutralized signals.

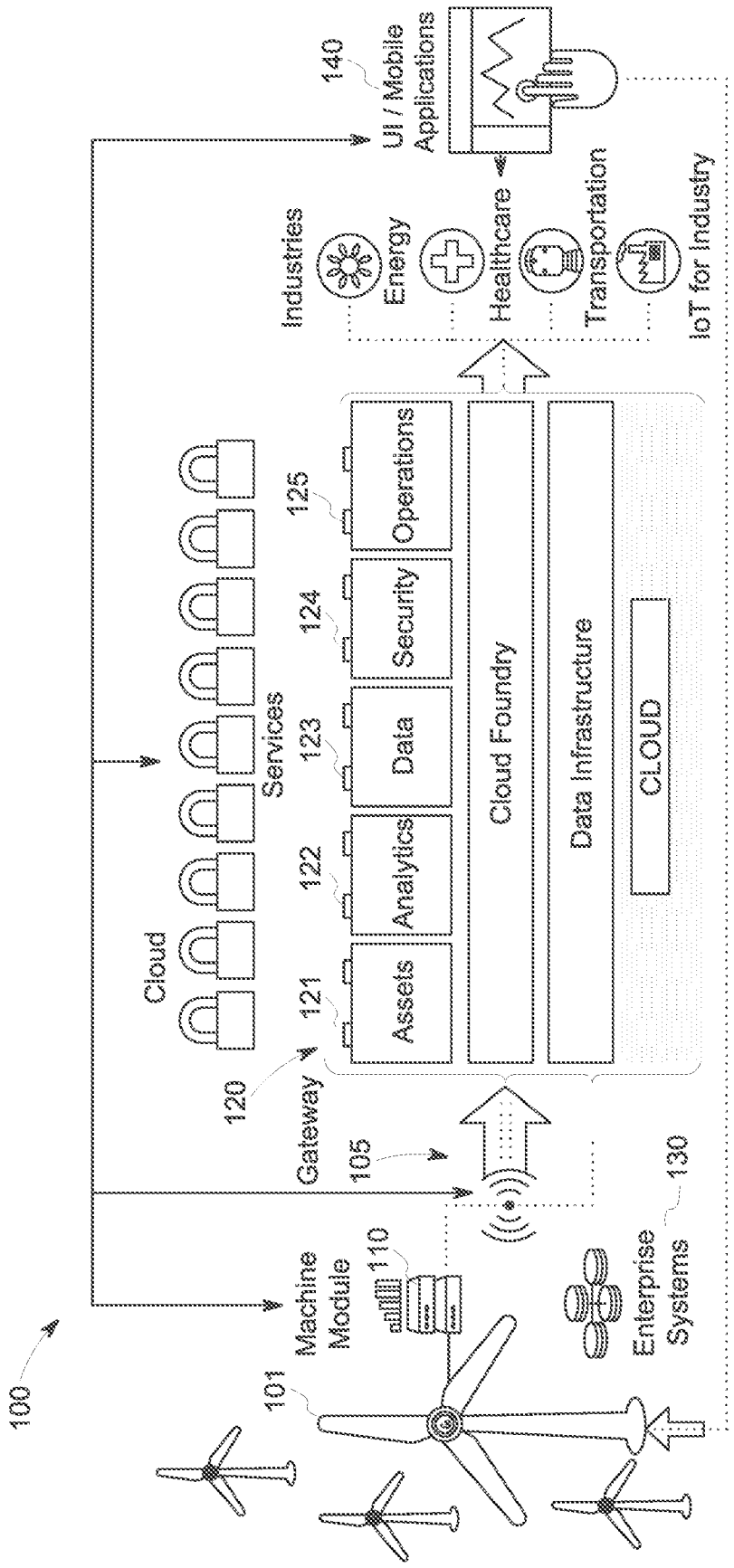

100
101
Machine Module
110
Enterprise Systems
130
105
Gateway
120
Cloud
Services
121
Assets
122
Analytics
123
Data
124
Security
125
Operations
Cloud Foundry
Data Infrastructure
CLOUD
Asset Optimization
145
140
UI / Mobile Applications
Industries
Energy
Healthcare
Transportation
IoT for Industry

FIG. 1

FIG. 2

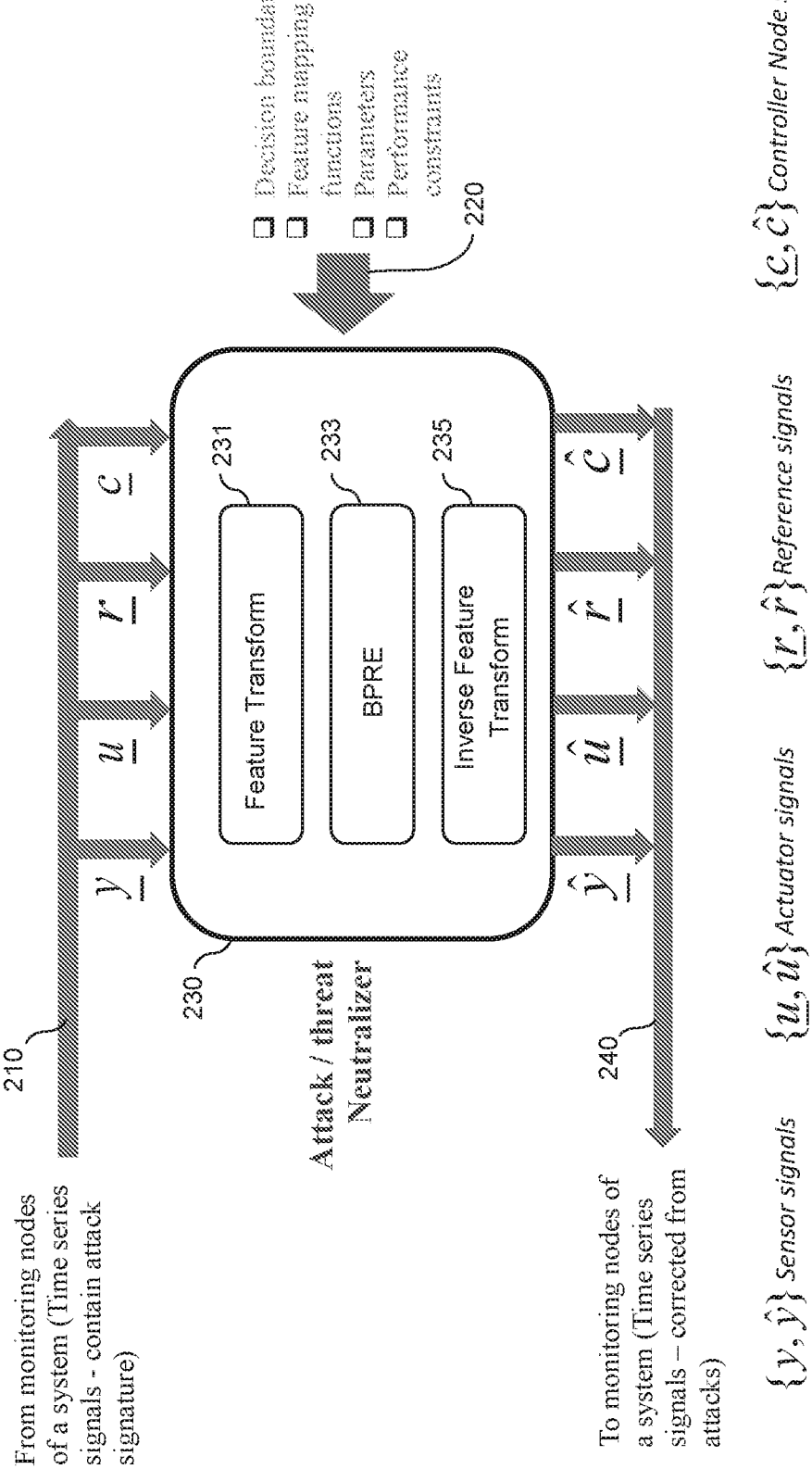

200
Decision boundary
Feature mapping functions
Parameters
Performance constraints
220
231
233
235
Feature Transform
BPRE
Inverse Feature Transform
230
Attack / threat Neutralizer
210
240
From monitoring nodes of a system (Time series signals - contain attack signature)
To monitoring nodes of a system (Time series signals – corrected from attacks)
$\{y, \hat{y}\}$ Sensor signals
$\{u, \hat{u}\}$ Actuator signals
$\{r, \hat{r}\}$ Reference signals
$\{c, \hat{c}\}$ Controller Node signals

FIG. 3

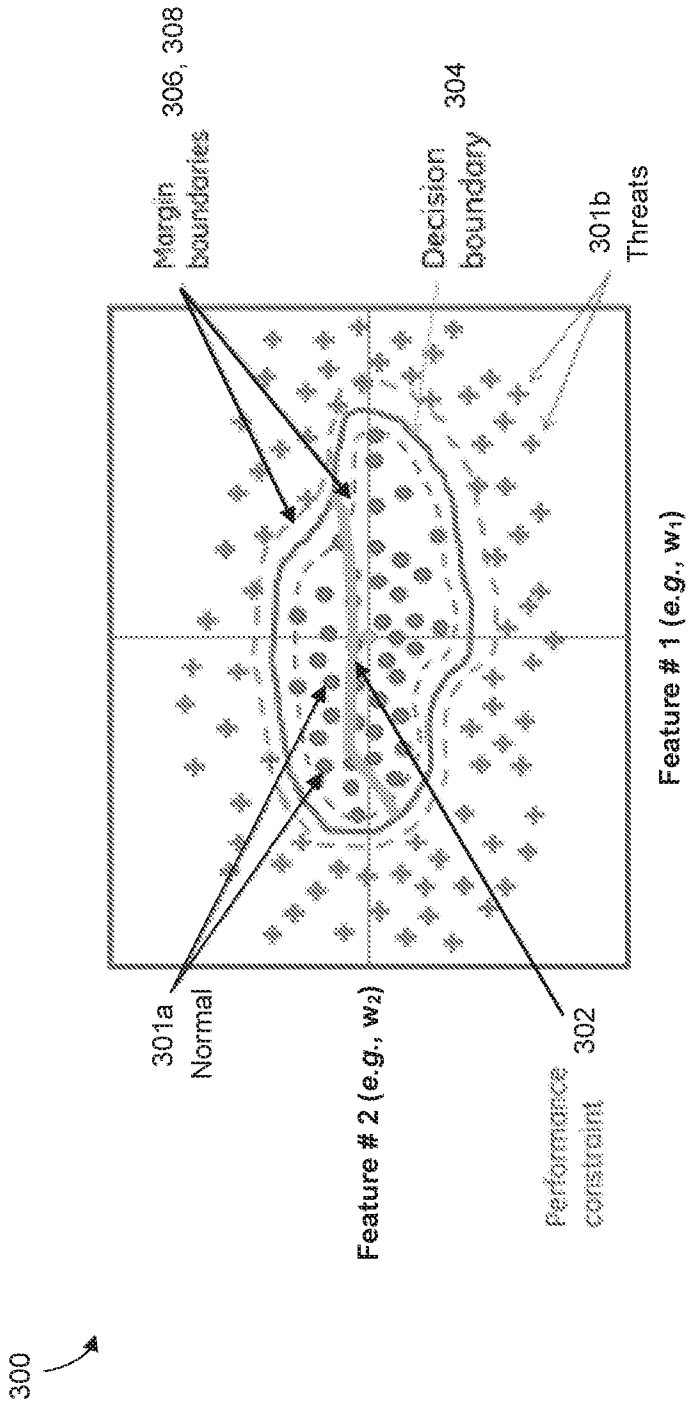
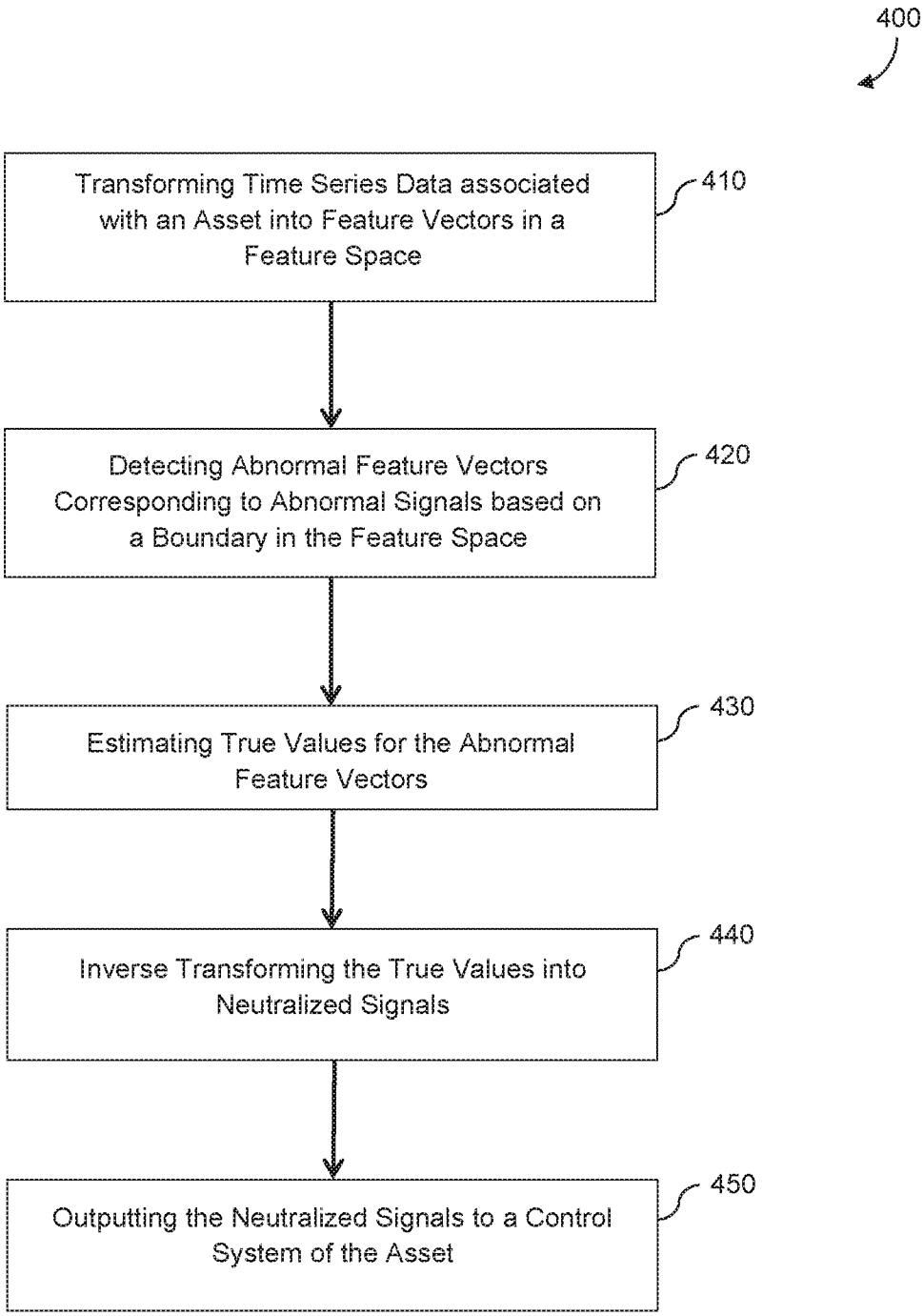

300
306, 308
Margin boundaries
304
Decision boundary
301b Threats
301a Normal
302
Performance constraint
Feature # 2 (e.g., w2)
Feature # 1 (e.g., w1)

FIG. 4


400
Transforming Time Series Data associated with an Asset into Feature Vectors in a Feature Space
410
Detecting Abnormal Feature Vectors Corresponding to Abnormal Signals based on a Boundary in the Feature Space
420
Estimating True Values for the Abnormal Feature Vectors
430
Inverse Transforming the True Values into Neutralized Signals
440
Outputting the Neutralized Signals to a Control System of the Asset
450

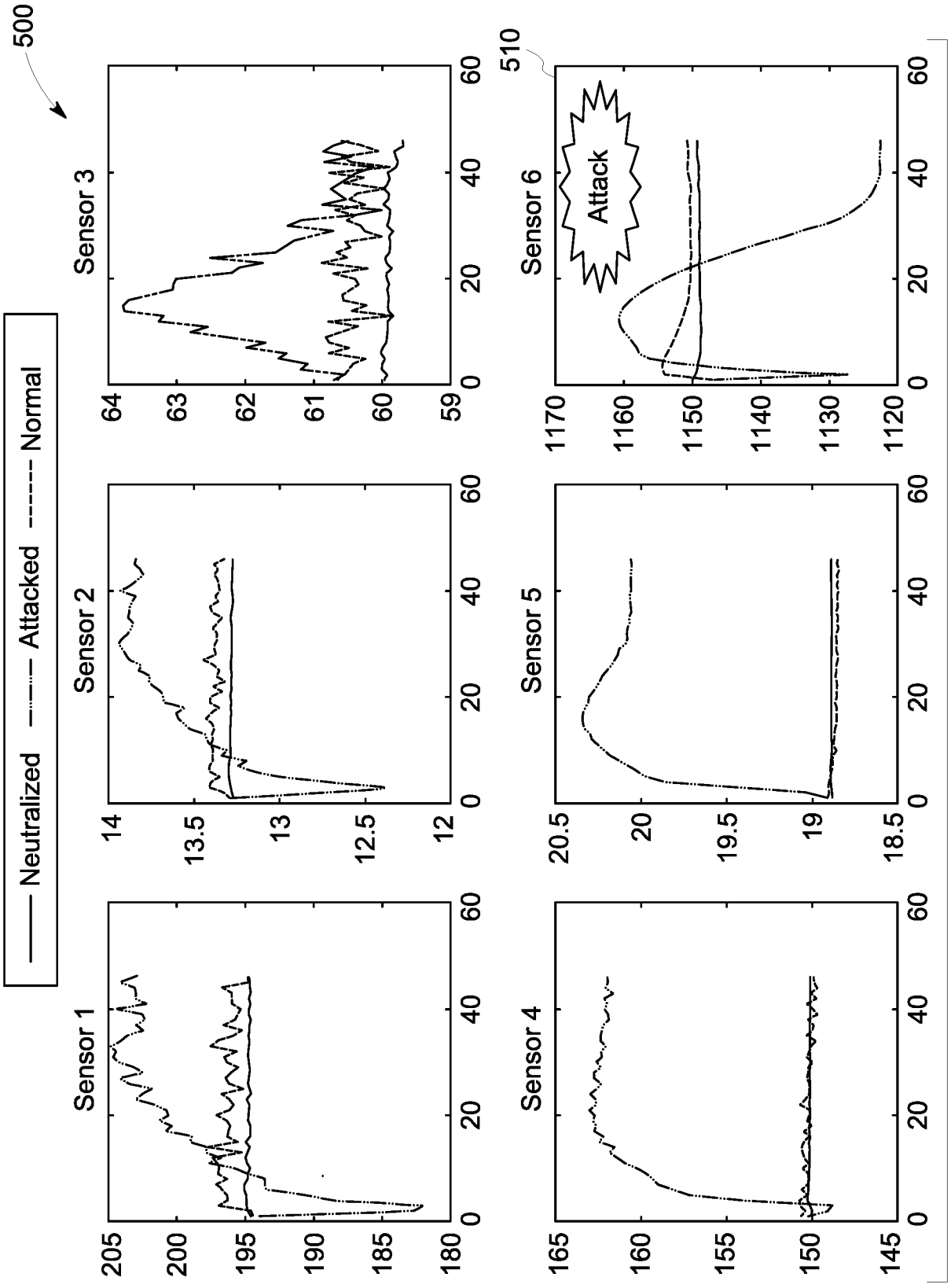
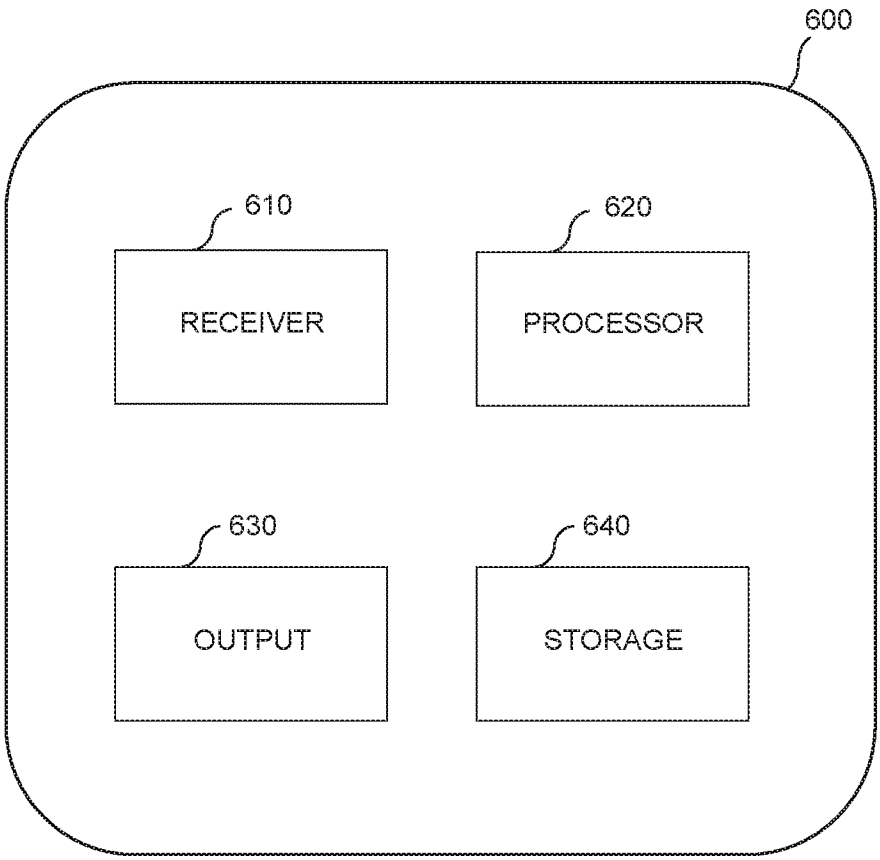

500
510
Neutralized
Attacked
Normal
Sensor 1
Sensor 2
Sensor 3
Sensor 4
Sensor 5
Sensor 6
Attack

FIG. 5

FIG. 6



**INTERNATIONAL SEARCH REPORT**

International application No.
**PCT/US2018/012858**

**A. CLASSIFICATION OF SUBJECT MATTER**

**H04L 29/06(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)
H04L 29/06; G06F 11/34; H04L 12/26; G06F 21/57; H04L 12/22

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: feature transform, feature value, abnormal feature value, normalcy boundary, true value, inverse transform, neutralized

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | US 2014-0298098 A1 (VIVIWARE, INC.) 02 October 2014<br>See paragraphs [0004], [0016], [0052]-[0055]; and figures 1-2. | 1-20 |
| A | KR 10-2013-0020265 A (KOREA UNIVERSITY RESEARCH AND BUSINESS FOUNDATION) 27 February 2013<br>See paragraphs [0017]-[0035]; and figure 1. | 1-20 |
| A | US 2014-0082730 A1 (KDDI CORP. et al.) 20 March 2014<br>See paragraphs [0056]-[0068]; claim 1; and figure 4. | 1-20 |
| A | US 2015-0295943 A1 (CYBER SENSE LTD.) 15 October 2015<br>See paragraphs [0001], [0067]-[0073]; claim 1; and figure 2. | 1-20 |
| A | WO 2016-022705 A1 (ATTACKIQ, INC.) 11 February 2016<br>See paragraphs [0007]-[0032]; and figure 7. | 1-20 |

☐ 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 | Date of mailing of the international search report |
|---|---|
| 04 May 2018 (04.05.2018) | **04 May 2018 (04.05.2018)** |

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
KIM, Seong Woo
Telephone No. +82-42-481-3348

**INTERNATIONAL SEARCH REPORT**

Information on patent family members

International application No.

**PCT/US2018/012858**

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| US 2014-0298098 A1 | 02/10/2014 | None | |
| KR 10-2013-0020265 A | 27/02/2013 | KR 10-1281456 B1 | 08/07/2013 |
| | | KR 10-2013-0020862 A | 04/03/2013 |
| | | US 2015-0304346 A1 | 22/10/2015 |
| | | WO 2013-027970 A1 | 28/02/2013 |
| US 2014-0082730 A1 | 20/03/2014 | JP 06184270 B2 | 23/08/2017 |
| | | JP 2014-060722 A | 03/04/2014 |
| | | US 9386030 B2 | 05/07/2016 |
| US 2015-0295943 A1 | 15/10/2015 | EP 3132372 A1 | 22/02/2017 |
| | | US 9516054 B2 | 06/12/2016 |
| | | WO 2015-159287 A1 | 22/10/2015 |
| WO 2016-022705 A1 | 11/02/2016 | US 2016-0044057 A1 | 11/02/2016 |