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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 January 2019 (24.01.2019)

WIPO | PCT

(10) International Publication Number
WO 2019/016608 A1

(51) International Patent Classification:
G06F 15/18 (2006.01) *G06N 3/12* (2006.01)

(21) International Application Number:
PCT/IB2018/000929

(22) International Filing Date:
18 July 2018 (18.07.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/653,441 18 July 2017 (18.07.2017) US

(71) Applicant: **ANALYTICS FOR LIFE INC.** [CA/CA]; 661 University Avenue, Suite 1300, Toronto, ON M5G 0B7 (CA).

(72) Inventors: **GROUCHY, Paul**; 661 University Avenue, Suite 1300, Toronto, Ontario, M5G 0B7 (CA). **BURTON, Timothy**; 661 University Avenue, Suite 1300, Toronto, ON M5G 0B7 (CA). **KHOSOUSI, Ali**; 661 University Avenue, Suite 1300, Toronto, ON M5G 0B7 (CA). **DOOMRA, Abhinav**; 661 University Avenue, Suite 1300, Toronto, ON M5G 0B7 (CA). **GUPTA, Sunny**; 661 University Avenue, Suite 1300, Toronto, ON M5G 0B7 (CA).

(74) Agent: **MANNING, Gavin, N.** et al.; Oyen Wiggs Green & Mutala LLP, 480 - 601 West Cordova Street, Vancouver, British Columbia V6B 1G1 (CA).

(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,

(54) Title: DISCOVERING GENOMES TO USE IN MACHINE LEARNING TECHNIQUES

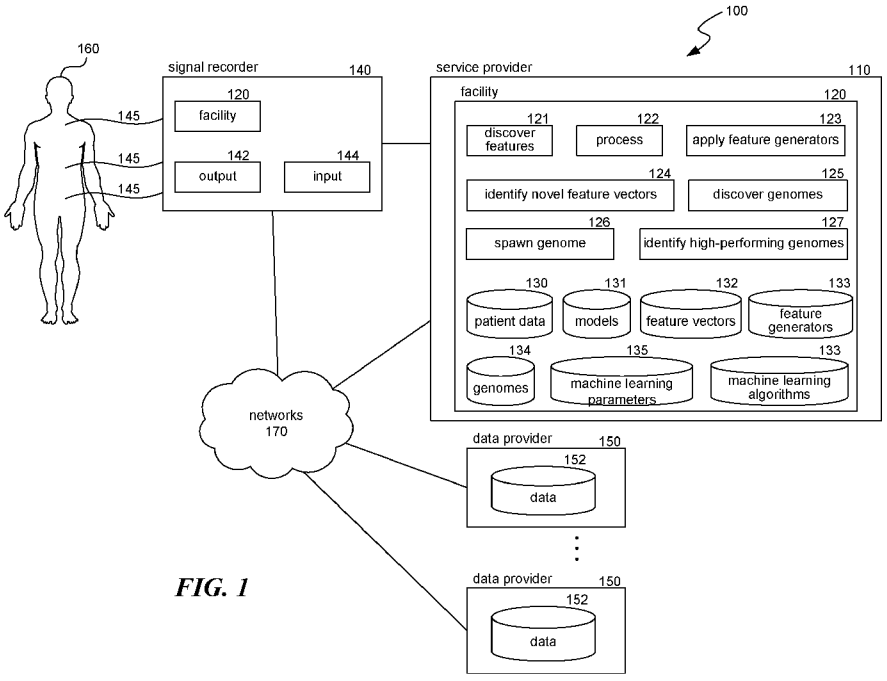

FIG. 1

(57) Abstract: A facility for identifying combinations of feature and machine learning algorithm parameters, where each combination can be combined with one or more machine learning algorithms to train a model, is disclosed. The facility evaluates each genome based on the ability of a model trained using that genome and a machine learning algorithm to produce accurate results when applied to a validation data set by, for example, generating a fitness or validation score for the trained model and the corresponding genome used to train the model. Genomes that produce fitness scores that exceed a fitness threshold are selected for mutation, mutated, and the process is repeated. These trained models can then be applied to new data to generate predictions for the underlying subject matter.

WO 2019/016608 A1

[Continued on next page]

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).

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))*

DISCOVERING GENOMES TO USE IN MACHINE LEARNING TECHNIQUES

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims priority to U.S. Patent Application 15/653,441, filed on July 18, 2018, which is hereby incorporated by reference in its entirety.

RELATED APPLICATIONS

[0002] This application is related to U.S. Patent Application No. 13/970,580, filed on August 19, 2013, entitled "NON-INVASIVE METHOD AND SYSTEM FOR CHARACTERIZING CARDIOVASCULAR SYSTEMS," now U.S. Patent No. 9,289,150; U.S. Patent Application No. 15/061,090, filed on March 4, 2016, entitled "NON-INVASIVE METHOD AND SYSTEM FOR CHARACTERIZING CARDIOVASCULAR SYSTEMS"; U.S. Patent Application No. 15/588,148, filed on May 5, 2017, entitled "NON-INVASIVE METHOD AND SYSTEM FOR CHARACTERIZING CARDIOVASCULAR SYSTEMS"; U.S. Patent Application No. 13/605,364, filed on September 6, 2012, entitled "SYSTEM AND METHOD FOR EVALUATING AN ELECTROPHYSIOLOGICAL SIGNAL," now U.S. Patent No. 8,923,958; U.S. Patent Application No. 13/970,582, filed on August 19, 2013, entitled "NON-INVASIVE METHOD AND SYSTEM FOR CHARACTERIZING CARDIOVASCULAR SYSTEMS FOR ALL-CAUSE MORTALITY AND SUDDEN CARDIAC DEATH RISK," now U.S. Patent No. 9,408,543; U.S. Patent Application No. 15/207,214, filed on July 11, 2016, entitled "NON-INVASIVE METHOD AND SYSTEM FOR CHARACTERIZING CARDIOVASCULAR SYSTEMS FOR ALL-CAUSE MORTALITY AND SUDDEN CARDIAC DEATH RISK"; U.S. Patent Application No. 14/295,615, filed on June 4, 2014, entitled "NONINVASIVE ELECTROCARDIOGRAPHIC METHOD FOR ESTIMATING MAMMALIAN CARDIAC CHAMBER SIZE AND MECHANICAL FUNCTION"; U.S. Patent Application No. 14/077,993, filed on November 12, 2013, entitled "NONINVASIVE ELECTROCARDIOGRAPHIC METHOD FOR ESTIMATING MAMMALIAN CARDIAC CHAMBER SIZE AND MECHANICAL FUNCTION"; U.S. Patent Application No. 14/596,541, filed on January 14, 2015, entitled "NONINVASIVE METHOD FOR ESTIMATING GLUCOSE, GLYCOSYLATED HEMOGLOBIN AND

OTHER BLOOD CONSTITUENTS," now U.S. Patent No. 9,597,021; U.S. Patent Application No. 15/460,341, filed on March 16, 2017, entitled "NONINVASIVE METHOD FOR ESTIMATING GLUCOSE, GLYCOSYLATED HEMOGLOBIN AND OTHER BLOOD CONSTITUENTS"; U.S. Patent Application No. 14/620,388, filed on February 12, 2015, entitled "METHOD AND SYSTEM FOR CHARACTERIZING CARDIOVASCULAR SYSTEMS FROM SINGLE CHANNEL DATA"; U.S. Patent Application No. 15/192,639, filed on June 24, 2016, entitled "METHODS AND SYSTEMS USING MATHEMATICAL ANALYSIS AND MACHINE LEARNING TO DIAGNOSE DISEASE"; U.S. Patent Application No. 15/248,838, filed on August 26, 2016, entitled "BIOSIGNAL ACQUISITION DEVICE"; U.S. Provisional Patent Application No. 62/397,895, filed on September 21, 2016, entitled "GRAPHICAL USER INTERFACE FOR CARDIAC PHASE-SPACE TOMOGRAPHY"; U.S. Patent Application No. 15/633,330, filed June 26, 2017, entitled "NON-INVASIVE METHOD AND SYSTEM FOR MEASURING MYOCARDIAL ISCHEMIA, STENOSIS IDENTIFICATION, LOCALIZATION AND FRACTIONAL FLOW RESERVE ESTIMATION"; and U.S. Patent Application No. 15/653,433, filed concurrently herewith, entitled "DISCOVERING NOVEL FEATURES TO USE IN MACHINE LEARNING TECHNIQUES, SUCH AS MACHINE LEARNING TECHNIQUES FOR DIAGNOSING MEDICAL CONDITIONS." Each of the above-identified applications and issued patents is hereby incorporated by reference in its entirety.

BACKGROUND

[0003] Machine learning techniques predict outcomes based on sets of input data. For example, machine learning techniques are being used to predict weather patterns, geological activity, provide medical diagnoses, and so on. Machine learning techniques rely on a set of features generated using a training set of data (i.e., a data set of observations, in each of which an outcome to be predicted is known), each of which represents some measurable aspect of observed data, to generate and tune one or more predictive models. For example, observed signals (e.g., heartbeat signals from a number of subjects) may be analyzed to collect frequency, average values, and other statistical information about these signals. A machine learning technique may use these features to generate and tune a model that relates these features to one or more conditions, such as some form of cardiovascular disease (CVD), including coronary artery disease (CAD), and then

apply that model to data sources with unknown outcomes, such as an undiagnosed patient or future weather patterns, and so on. Conventionally, these features are manually selected and combined by data scientists working with domain experts.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] Figure 1 is a block diagram illustrating an environment in which the facility operates in some embodiments.

[0005] Figure 2 is a flow diagram illustrating the processing of a discover features component in some embodiments.

[0006] Figure 3 is a flow diagram illustrating the processing of a process component in some embodiments.

[0007] Figure 4 is a flow diagram illustrating the processing of an apply feature generators component in some embodiments.

[0008] Figure 5 is a flow diagram illustrating the processing of an identify novel feature vectors component in some embodiments.

[0009] Figure 6 is a flow diagram illustrating the processing of a discover genomes component in accordance with some embodiments.

[0010] Figure 7 is a flow diagram illustrating the processing of a spawn genome component in accordance with some embodiments.

[0011] Figure 8 is a flow diagram illustrating the processing of an identify high-performing genomes component in accordance with some embodiments.

DETAILED DESCRIPTION

[0012] Because machine learning techniques rely on features and/or combinations of features, the process of feature selection and combination typically is an important part of a machine learning process. Moreover, because a large number of diverse machine learning algorithms exist (e.g., decision trees, artificial neural networks (ANNs), deep ANNs, genetic (and meta-genetic) algorithms, and so on), the choice of algorithm and any associated parameters can also be important. For example, different machine learning algorithms (or family of machine learn algorithms) may be best suited for different types of data and/or the types of predictions to be made. Furthermore, different machine learning algorithms may

present various tradeoffs with respect to resources (e.g., memory, processor utilization), speed, accuracy, and so on. Typically, models are trained using machine learning algorithms, features, and parameters selected by individuals based on the preferences of those individuals and/or criteria specified by those individuals. The inventors have recognized that it can be expensive and time-consuming manually to identify features, machine learning algorithms, and corresponding parameters and even more difficult to produce features, machine learning algorithms, and corresponding parameters that produce more accurate models and, therefore, more accurate predictions. Accordingly, the inventors have conceived and reduced to practice a facility that performs automatic discovery of combinations of features, machine learning algorithms, and/or machine learning parameters.

[0013] In some embodiments, the facility operates as part of a machine learning pipeline that constructs and evaluates predictive models, such as those for disease diagnosis, based on time-series and/or other signals, such as physiological signals. The machine learning process uses features to identify patterns within a training set of data and, based on these patterns, generates predictive models. These predictive models can be validated using validation data sets (i.e., data sets for which an outcome is known but that were not used to train the model) and applied to new input data in order to predict outcomes from the input data, such as providing a diagnosis for a medical condition, etc. As new data and new features are produced or acquired, the machine learning process improves upon the predictive capabilities of these models by incorporating new features and, in some cases, discarding others, such as those that are determined to be too similar to other features.

[0014] In particular, the facility seeks to identify combinations of features and machine learning algorithm parameters where each combination can be used to train one or more models. A combination of features and/or machine learning parameters is sometimes referred herein to as a "genome." The facility evaluates each genome based on the ability of a model trained using a machine learning algorithm and that genome to produce accurate results when applied to a validation data set by, for example, generating a fitness or validation score for the trained model and the corresponding genome used to train the model. In some cases, the facility uses the validation score as a fitness score while in other cases the validation score is an element of a fitness score (e.g., fitness score = training score + validation score). In

some cases, multiple models may be trained using a genome and the resulting fitness scores can be aggregated to generate an aggregated fitness score for the genome.

[0015] By way of example, the facility for identifying combinations of features and machine learning algorithm parameters can be used for a medical diagnosis predictive modeling task. In this example, the facility receives, for a number of patients or subjects, one or more sets of physiological data that relate to some type of physiological output or condition of the patient over a period of time (e.g., less than a second, on the order of a few seconds, about ten seconds, about 30 seconds and up to about five minutes, about an hour or more, etc.), such as electroencephalograms, and so on. These data may be received in real-time or near real-time, concurrent or nearly concurrent with the operation of the facility, or they may be received at an earlier time. In some cases, the facility discards certain portions of the signal to ensure that the signals from each patient commence at a stable and consistent initial condition. Furthermore, the data may be normalized to remove potentially misleading information. For example, the facility can normalize the amplitude of signal data (e.g., transforming to a z-score), to account for variations in signal strength caused by sensor contact or other non-physiological data. As another example, in the case of a cardiac signal, the facility can perform a peak search and discard any data before a first heartbeat identified in the signal and after a last heartbeat identified in the signal.

[0016] In some embodiments, the facility applies a set of feature generators to a set of signals to generate, for each combination of a signal and a feature generator, a feature value for the signal. Thus, each feature value is representative of some property of the underlying signal data. In one example, the facility receives patient data for each of 1000 patients and applies one or more feature generators to the data to generate, for each application of a feature generator to the data of a single patient, a feature value (or set of feature values). The facility collects the feature values generated by a single feature generator in a "feature vector," such that the feature vector stores one feature value per patient. Once the feature vectors are generated, they can be compared to determine how different each is relative to each of the other feature vectors. The facility computes a distance metric for each feature vector to assess the novelty of the corresponding feature generator. Based

on the assessed novelty, the facility (1) provides the feature generators that produced the novel feature vectors to the machine learning process for the purpose of basing new predictive models on the provided feature generators and (2) modifies these feature generators to create a new generation of feature generators. The facility repeats this evolutionary process to identify even more novel features for use by the machine learning process.

[0017] In some embodiments, for each received set of data, the facility computes or identifies separate sets of one or more values from the data. For example, in the case of data generated as part of an electrocardiogram, the facility identifies global and local maxima and minima within the data, computes frequency/period information from the data, calculates average values of the data over a certain period of time (e.g., the average duration and values generated during a QRS complex), and so on. In some cases, the facility transforms the received data and extracts sets of one or more values from the transformed data. The facility can transform received signal data in any number of ways, such as taking one or more (successive) derivatives of the data, taking one or more partial derivatives of the data, integrating the data, calculating the gradient of the data, applying a function to the data, applying a Fourier transform, applying linear or matrix transformations, generating topology metrics/features, generating computational geometry metrics/features, generating differential manifold metrics/features, and so on. In this manner, the facility generates multiple perspectives of the data in order to yield a diverse set of features. While these transformations are provided by way of example, one of ordinary skill will recognize that the data can be transformed in any number of ways.

[0018] In one example, the facility receives multiple input signals (e.g., input signals collected by different electrodes or leads connected to a patient, multimodal signals, such as signals from leads of wide-band biopotential measuring equipment and a channel of S_pO_2 (blood oxygen saturation), and so on) and/or transformed signals and extracts values from the signal data by computing, for each signal, an average value of the signal over the sampling period. In this example, four signals per patient are represented, although one of ordinary skill in the art will recognize that any number of signals may be monitored and/or received for processing and

further analysis by the facility. Thus, in this example, the extracted data of each patient can be represented as a set of these average values over time, such as:

Patient	A	B	C	D
1	0.24	0	0	30
2	0.2	0.6	4.2	5
...				
n	.32	2	4	.02

Table 1

Table 1 represents a set of average signal values (A, B, C, and D) for each of n patients. Although average values have been used here, one of ordinary skill in the art will recognize that any type of data can be extracted or computed from the underlying data signals, such as the amount of time that a signal exceeded a threshold value, the values for one signal while the value of another signal exceeded a threshold value, and so on.

[0019] In some embodiments, after data have been extracted from the received signal, the facility applies one or more feature generators to the received or generated data, such as the extracted data, the raw or preprocessed signal data, the transformed data, and so on. A feature generator receives as input at least a portion or representation of the signal data and produces a corresponding output value (or set of values) (i.e., a "feature"). One set of feature generators includes the following equations:

$$F1 = A + C - D, \quad \text{(Eq 1)}$$

$$F2 = \frac{A*S(4)*B}{D} + C + \sqrt{D}, \text{ and} \quad \text{(Eq 2)}$$

$$F3 = S(1) * D, \quad \text{(Eq 3)}$$

where each of A, B, C, and D represents a value extracted from a specific patient's data and $S(t)$ represents, for each signal, the value of the signal at time t. In Eq 1, for example, $F1$ represents the name of the feature while the equation $A + C - D$ represents the corresponding feature generator. In some cases, the facility employs

composite feature generators in which one feature generator serves as an input to another feature generator, such as:

$$F4 = \frac{F1 * F2}{\sqrt[3]{F3}} + .057 \quad \text{(Eq 4)}$$

In this example, the facility applies feature generators to the extracted data of each patient represented in Table 1 to generate, for each feature generator, a feature vector of three values (one for each patient), such as those represented in Table 2 below:

Patient	$F1$	$F2$	$F3$
1	-29.76	5.48	905.83
2	-0.6	6.67	9.57
…			
n	4.3	185.74	0.04

Table 2

[0020] In this example, the facility has applied each feature generator $F1$, $F2$, and $F3$ to the extracted data shown in Table 1 to generate, for each feature generator, a corresponding feature vector that includes a value for each patient. For example, the feature vector generated by applying feature generator $F1$ to the extracted data includes a value of -29.76 for Patient 1, a value of -0.6 for patient 2, and so on. Thus, each feature vector represents, for a specific feature generator, a signature (not necessarily unique) for the corresponding feature generator based on at least a portion of each patient's physiological data (i.e., the patients represented in the physiological data to which the feature generators were applied). In some examples, feature generators are expressed using different structures or models, such as expression trees, neural networks, etc. One of ordinary skill in the art will recognize that the facility may employ any number of feature generators and any number of sets of physiological data (or portions thereof) in the generation of feature vectors. In some embodiments, the facility randomly selects a number of previously-generated feature generators for use in generating feature vectors rather than employing each and every available feature generator. In some embodiments, the facility creates and/or modifies feature generators by, for example, randomly

generating expression trees, randomly assigning weights to connections within a neural network, and so on.

[0021] In some embodiments, after the facility generates a number of feature vectors, the facility employs some form of novelty search to identify the most "novel" feature vectors among the generated feature vectors. Novelty corresponds to how different a particular feature vector is from each of a comparison set of other feature vectors (made up of any feature vectors generated by the facility during a current iteration and feature vectors produced by feature generators selected in any earlier iteration); the greater the difference from the feature vectors of the comparison set, the greater the novelty. The facility uses a form of distance as a measure of novelty (i.e., how "far" each feature vector is from the other feature vectors). In this case, for each generated feature vector, the facility calculates the distance between that feature vector and each of the other generated feature vectors and performs an aggregation of the generated distance values, such as calculating an average or mean (e.g., arithmetic, geometric, harmonic, etc.) distance value for the feature vector, or a total (sum) distance between the feature vector and each of the other generated feature vectors, identifying a mode distance value, a median distance value, a maximum distance value for the feature vector, and so on. For example, using the feature vectors of Table 2 (for patients 1, 2, and n), the distances for each set of feature vectors could be calculated as such:

$F1$-$F2$ distance: $\sqrt{(-29.76-5.48)^2+(-0.6-6.67)^2+(4.3-185.74)^2}$ = 184.97.

$F1$-$F3$ distance: $\sqrt{(-29.76-905.83)^2+(-0.6-9.57)^2+(4.3-0.04)^2}$ = 936.23

$F2$-$F3$ distance: $\sqrt{(5.48-905.83)^2+(6.67-9.57)^2+(185.74-0.04)^2}$ = 919.70.

In this example, the total Euclidean distance between each of the feature vectors has been calculated as a means for calculating a difference between each of two vectors. In addition to the feature vectors generated by a current set (i.e., a current generation) of feature generators, the facility includes feature vectors produced by feature generators selected in an earlier generation. In some examples, the facility applies a weight, such as a randomly generated weight, to each of the feature vectors and/or normalizes each set of feature vectors prior to comparison. Thus, the distance measurements for each of the feature vectors in this example are as follows:

Feature Generator	Distance to $F1$	Distance to $F2$	Distance to $F3$	Average Distance	MAX Distance
$F1$	–	184.97	936.23	560.60	936.23
$F2$	184.97	–	919.70	552.34	919.70
$F3$	936.23	919.70	–	927.97	936.23

Table 3

[0022] In this example, the facility identifies the most "novel" feature vectors based on the calculated distances, which act as a "novelty score" or "fitness score" for each of the feature vectors. The facility identifies the feature vectors with the greatest average distance to other vectors (e.g., the feature vector generated by $F3$), the feature vectors with the greatest MAX distance (e.g., the feature vectors generated by $F1$ and $F3$), and so on. In some examples, the number of novel feature vectors identified is fixed (or capped) at a predetermined number, such as five, ten, 100, 500, etc. In other examples, the number of novel feature vectors to be identified is determined dynamically, such as the top 10% of analyzed feature vectors based on novelty scores, any feature vectors having a novelty scores that is more than a predetermined number of standard deviations beyond a mean novelty score for the analyzed feature vectors, and so on. The feature generators that produced each of these identified novel feature vectors can then be added to the set of features available for use as inputs to models constructed and evaluated by the machine learning pipeline. Those models can be applied to patient data for, e.g., diagnostic, predictive, therapeutic, or other analytic, scientific, health-related or other purposes.

[0023] In some embodiments, in addition to providing the feature generators used to generate the identified novel feature vectors for use by the machine learning process, the facility randomly mutates or modifies the feature generators used to generate the identified novel feature vectors. Each mutation effects some change in the corresponding feature generator and creates a new version of the feature generator that can be used to contribute to a new generation of feature generators. The facility uses this new feature generator to generate new feature vectors, and then assesses the novelty of the new feature vectors. Moreover, the corresponding

feature generator can be further mutated to continue this process of feature vector and feature generation creation. For example, a feature generator expressed in the form of an equation, such as $F1_0 = A + C - D$, can be mutated by randomly selecting one or more element(s) of the equation and replacing the selected element(s) with other elements (e.g., randomly selected elements). In this example, the equation can be changed by replacing A with B to create $F1_1 = B + C - D$ or replacing $C - D$ with $\sqrt[3]{C - B^2}$ to create $F1_1 = B + \sqrt[3]{C - B^2}$. In this case, the subscripted 0 and 1 have been included to represent a generational marker or count for each of the feature generators. In other words, $F1_0$ represents $F1$ above (Eq 1) at generation 0 (i.e., the first generation), $F1_1$ represents a mutated version of $F1$ at generation 1 (i.e., the second generation), and so on. In some cases, an earlier generation (or a transformation thereof) is included as an element in subsequent generations, such as $F2_1 = \sqrt{F2_0} + C^2$ or $F2_n = \sqrt{F2_{n-1}} + C^2$ $(n \neq 0)$.

[0024] In some embodiments, the facility obtains features in different ways. For example, the facility may receive from a user, such as a domain expert, a set of features (and corresponding feature generators) that the user has identified as being optimal and/or that the user desires to be tested. As another example, the features may be editorially selected from one or more feature stores. In some cases, features automatically generated by the facility can be combined with other features to create various hybrid features. Even features of unknown provenance may be used.

[0025] In some embodiments, the facility identifies genomes to train models, identifies, from among these genomes, the "best" (highest rated) genomes, and mutates the identified genomes to produce even more genomes that can be used to train models. After using a genome to train one or more models, the facility applies each trained model to a validation data set so that the trained model can be scored (e.g., how well does the trained model correctly identify and/or classify subjects in the underlying validation data set). The facility mutates the genomes that produce the best results (e.g., have the highest validation or fitness scores), trains new models using these mutated genomes, and repeats this process until one or more termination criteria are met (e.g., a predetermined number of generations, no additional high scoring (higher than a predetermined or dynamically generated threshold) genomes are generated during a predetermined or dynamically number (e.g., 1, 5, 8, 17, etc.) of previous generations, a combination thereof, etc.).

[0026] In some embodiments, the facility uses previously identified or generated genomes as a first set of genomes (i.e., a first generation) from which to discover genomes for machine learning algorithms. In other examples, the facility automatically generates a first generation of genomes by, for each genome, randomly (with or without replacement) selecting one or more feature vectors from one or more previously generated sets of feature vectors (e.g., a feature vector produced by applying a feature generator to a set of training data). A genome may also include one or more machine learning algorithm parameters to the machine learning algorithm, such as the number of predictors (e.g., regressors, classifiers, the number and/or the maximum number of decision trees to use for a machine learning algorithm, etc.) to use for an underlying ensemble method associated with the algorithm, a maximum depth for a machine learning algorithm (e.g., maximum depth for decision trees), and so on. In the event that the genome is configured to be used with one specific machine learning algorithm, the genome can be configured to define a value for each machine learning parameter associated with that machine learning algorithm. In other cases, one of the elements of the genome selects among different machine learning algorithms and may be mutated so that the genome and its corresponding parameter values are used with different machine learning algorithms to train models over the evolutionary process. For example, during a first generation, a genome may identify a machine learning algorithm that relies on decision trees while a mutated version of that same genome identifies a machine learning algorithm that uses one or more support vector machines, linear models, etc. In these cases, the genome may specify a modeling parameter for each and every machine learning algorithm that may be combined with the genome to train a model. Thus, a single genome may include machine learning parameters for multiple machine learning algorithms. However, a genome need not include each and every modeling parameter for a corresponding machine learning algorithm. In the event that a model is to be trained using a particular machine learning algorithm and a genome that does not include a value for a machine learning parameter of that machine learning algorithm, the facility can retrieve a default value for these parameters from, for example, a machine learning parameter store.

[0027] For example, a set of genomes may be represented as:

$G1_1$	MLA=4	F23	F78798	F32	F55	F453	F234

$G2_1$	MLA=9	F9701	F223	F1	F63	F349	P9:1=7
$G3_1$	MLA=2	F823	F525	F732	F525	F125	
$G4_1$	MLA=6	F597	F135	F404	F31	P6:1=5	P6:2=150
…							
$G20_1$	MLA=1	F43	F65	P1:1=8	P1:2=218	P1:3=0.3	

Table 4

where each row corresponds to a different genome (named in the first column from the left) from among a first generation of selected or generated genomes and identifies a machine learning algorithm ("MLA"; second column from the left) to use to train a model using the genome, such as an index into a machine learning algorithm store. For example, genome $G3_1$ specifies a machine learning algorithm corresponding to index 2 in a machine learning algorithm store (MLA=2). In this example, each non-shaded region (to the right of the second column) identifies a different feature. A genome can also include a corresponding feature generator or a reference to the corresponding feature generator, such as a link to feature generator store. As discussed above, these features may be generated automatically by the facility and/or retrieved from another source.

[0028] Furthermore, each shaded region in Table 4 represents a value for a particular machine learning parameter. In this example set of genomes, machine learning parameters are represented by an indicator or reference (e.g., P6:1) followed by an equals sign and a corresponding value. For example, machine learning algorithm parameter P6:1 has a corresponding value of 8 in genome $G20_1$. In this example set of genomes, each machine learning parameter is presented as an index into a two-dimensional array, such that "P6:1" represents the "first" machine learning parameter of the "sixth" machine learning algorithm (i.e., the machine learning parameter with an index of 1 for the machine learning algorithm with an index of 6). As discussed above, a genome may specify values for any or all machine learning parameters that may be used to training a model using the genome (or a mutated version of that genome). Moreover, as is clear from Table 4, genomes may be of varying length. For example, genome $G1_1$ includes values for six features and zero machine learning parameters while gnome $G2_1$ includes values

for two features and three machine learning parameters. Accordingly, the facility may employ variable-length genomes in the machine learning processes.

[0029] In some embodiments, the facility may filter features from within genomes and/or filters genomes themselves to avoid redundancy among each. In order to filter features and/or genomes, the facility generates correlation values for each pair and discards one item of the pair. To identify and filter correlated features from a genome, the facility generates, for each of the features, a feature vector by applying a feature generator associated with the feature to a training set of data to produce a set of values. The facility compares each of the generated feature vectors to the other generated feature vectors to determine whether any of the feature vectors are "highly" correlated (i.e., not "novel" within the selected set of feature vectors). For example, the component may calculate a distance value for each of the generated feature vectors relative to the other feature vectors (as discussed above with respect to identifying novel feature generators) and, if the distance between any pair (set of two) is less than or equal to a distance threshold (i.e., "highly" correlated or not "novel"), discard a feature corresponding to one of the pair of feature vectors. Moreover, the facility may replace the discarded feature with a new feature, such as a randomly-selected feature. Similarly, the facility may identify and discard redundant genomes by generating, for each feature of the genome, a feature vector, calculating distance metrics for each pair (set of two) of genomes based on the generated feature vectors, and identifying pairs of genomes whose calculated distances do not exceed a genome distance threshold. For each identified pair of genomes, the facility may discard or mutate one or both of the genomes to reduce correlation and redundancy among a group of genomes. Although distance is used in this example as a metric for determining a correlation between two vectors or sets of vectors, one of ordinary skill in the art will recognize that correlations between two or sets of vectors can be calculated in other ways, such as normalized cross-correlation, and so on. In some embodiments, the facility may employ additional or other techniques to filter genomes, such as generating a graph where features represent vertices in the graph which are connected via edges in the graph. An edge between two features is generated if, for example, a correlation value between the two features exceeds a predetermined correlation threshold and/or the distance between the two features is less than a predetermined

distance threshold. Once the graph is generated, the facility removes connected vertices (features) from the graph until no edges remain in the graph (an edge being removed when a connected vertex is removed) and selects the remaining non-connected vertices (features) for inclusion in the "filtered" genome. In some cases, the facility may randomly select connected vertices for removal. Moreover, the facility may perform this process multiple times for a set of vertices (features) and then select a preferred "filtered" genome, such as the genome with the most or least vertices (features) removed.

[0030] In order to test the fitness or validity of each genome, the facility trains at least one model using the features, machine learning parameters, and/or machine learning algorithm(s) of that genome. For example, the facility can use AdaBoost ("Adaptive Boosting") techniques to train a model using the corresponding features, machine learning parameters, machine learning algorithm, and a training set of data. However, one of ordinary skill in the art will recognize that many different techniques can be used to train one or more models given a genome or a set of genomes. After the model is trained, the facility applies the trained model to one or more sets of validation data to assess how well the trained model identifies and/or classifies previously-identified or classified subjects within the validation data set. For example, a genome may be generated to train models to identify patients represented in a data set who are likely to have diabetes. Once a model is trained using one of these genomes, the trained model can be applied to a validation set of data to determine a validation score that reflects how well the trained model identifies patients from the validation set that are known to have or now have diabetes; scoring (adding) one "point" for every correct determination (e.g. true positives and true negatives) and losing (subtracting) one "point" for every incorrect determination (e.g., false positives and false negatives). Thus, an overall score for the trained model can be determined based on how many "points" the trained model scores when applied to one or more sets of validation data. One of ordinary skill in the art will recognize that several techniques may be used to generate a fitness score for a trained model, such as calculating the area under a corresponding receiver operating characteristic (ROC) curve, calculating a mean squared prediction error, f scores, sensitivity, specificity, negative and positive predictive values, diagnostic odds ratios, and so on. In this example, where a single machine learning

algorithm is trained using the genome, the generated fitness score may be similarly attributed to the genome. In other case, the genome may be used to train multiple machine learning algorithms and each of those trained machine learning algorithms may be applied to multiple validation sets to produce, for each genome used to train machine algorithms, multiple fitness scores. In these cases, the facility generates a fitness score for the corresponding genome by aggregating each of the fitness scores generated for the machine learning algorithms trained using the genome. In some cases, the generated fitness scores may be aggregated and/or filtered prior to aggregation.

[0031] In some embodiments, after the facility has produced fitness scores for each of the genomes, the facility identifies the "best" genomes based on these fitness scores. For example, the facility can establish a fitness threshold based on the produced fitness scores and identify the "best" genomes as those genomes whose resulting fitness scores exceed the fitness threshold. The fitness threshold may be generated or determined in any number of ways, such as receiving a fitness threshold from a user, calculating a fitness threshold based on the set of fitness scores (e.g., average, average plus 15%, top fifteen, top n-th percentile (where n is provided by a user or generated automatically by the facility), and so on. The facility then stores each of the genomes in association with their corresponding fitness scores and selects the genomes identified as "best" for mutation (i.e., the genomes having a fitness score that exceeds a fitness threshold).

[0032] In some embodiments, the facility mutates a genome by adding, removing, or changing any one or more of the feature vectors or machine learning parameters of the genome. For example, Table 5 below represents a number of mutations to the genomes represented above in Table 4.

$G1_2$	MLA=5	F23	F78798	F32	~~F55~~	~~F453~~	~~F234~~		
$G2_2$	MLA=9	~~F9701~~	F223	F1	F63	F349	F584	P9:1=12	
$G4_2$	MLA=6	F597	F135	F404	F31	F24	F982	P6:1=5	P6:2=150
...									
$G20_2$	MLA=1	F43	F65*F14	P1:1=8		P1:2=218	P1:3=0.3		

Table 5

In this example, each row corresponds to a different genome (named in the first column from the left) from among a second generation of genomes selected for mutation. In this example, based on its low fitness score, the facility did not select genome $G3_1$ for mutation and, therefore, Table 5 does not include a corresponding entry for a mutated version of this genome. Moreover, genome $G1_1$ has been mutated (represented as $G1_2$) by removing three feature vectors (represented with a strikethrough) and changing the references machine learning algorithm index from 4 to 5. Furthermore, the facility has mutated genome $G2_1$ by 1) removing feature vector F9701, 2) adding feature vector F584, and 3) adjusting machine learning parameter $P9_1$ from 7 to 12; genome $G4_1$ by adding features F24 and F982; and genome Gn_1 by multiplying values generated by F65 by values generated by F14. These mutated genomes can then be used to train one or more machine learning algorithms, scored by applying the trained machine learning algorithm to one or more validation data sets, selected for mutation, mutated, and so on. The facility performs this process until a termination point is reached, such as when a predetermined number of generations has been produced (e.g., six, 30, 100,000, etc.), and so on.

[0033] Figure 1 is a block diagram illustrating an environment 100 in which the facility operates in accordance with some embodiments of the disclosed technology. In this example, environment 100 includes service provider 110, signal recorder 140 (e.g., a wide-band biopotential measuring equipment), data providers 150, patient 160, and network 160. In this example, service provider includes facility 120, which includes discover features component 121, process component 122, apply feature generators component 123, identify novel feature vectors component 124, discover genomes component 125, spawn genome component 126, identify high-performing genomes component 127, patient data store 130, model store 131, feature vector store 132, and feature generator store 133. Discover features component 121 is invoked by the facility to identify and mutate feature generators based on received data. Process component 122 is invoked by the discover features component 121 to process and transform patient signal data, such as raw signal data from signal recorder 140 (e.g., one or more measurement devices and/or systems used to collect the underlying data such as wide-band biopotential measuring equipment, etc.), 3-D image data, etc. Apply feature generators component 123 is invoked by the discover features component to apply a set of one

or more feature generators to the processed and transformed patient signal data. Identify novel feature vectors component 124 is invoked by the discover features component to identify the most novel feature vectors from among a group of feature vectors generated by, for example, one or more feature generators. Discover genomes component 125 is invoked by the facility 120 to generate, analyze, and mutate genomes for use by machine learning algorithms. Spawn genome component 126 is invoked by the discover genomes component to generate a genome comprising any number of feature vectors and/or machine learning parameters. Identify high-performing genomes component 127 is invoked by the discover genomes component to identify, from among a group of genomes, the genomes that have a corresponding fitness score that exceeds a fitness threshold. Patient data store 130 includes physiological patient data, such as raw physiological data (including but not limited to data obtained via, e.g., signal recorder 140), transformed physiological data, biographical information, demographic information, etc. These data may be stored anonymously to protect the privacy of each of the corresponding patients and may be processed and encrypted to ensure that its transmission and storage complies with any governing laws and their implementing regulations, such as the U.S. Health Insurance Portability and Accountability Act of 1996 (as amended), the European Data Protection Directive, the Canadian Personal Information Protection and Electronics Documents Act, the Australian Privacy Act of 1988, Japan's Personal Information Protection Act of 2015 (as amended), state and provincial laws and regulations, and so on. Model store 131 stores information about models generated by applying machine learning techniques to training data, such as the machine learning techniques described in Christopher M. Bishop, *Pattern Recognition and Machine Learning* (2006) (Library of Congress Control Number: 2006922522; ISBN-10: 0-387-31073-8), which is herein incorporated by reference in its entirety. Feature vector store 132 stores sets of feature vectors generated by applying one or more feature generators to a set of physiological data. Feature generator store 133 stores sets of feature generators that can be applied to patient physiological data and can include multiple generations of feature generators. Genome store 134 stores generated and/or mutated genomes created by the facility and/or other sources. Machine learning parameters store 135 stores, for each of a number of machine learning algorithms, a set of parameters that can serve as inputs to that machine learning algorithm and additional information related to the

parameter, such as a maximum value for a corresponding parameter, a minimum value for a corresponding parameter, a default value for a corresponding parameter, etc. Machine learning algorithm store 133 stores the logic for each of a number of machine learning algorithms, each of which can be selectively trained and validated by the facility. In this example, a signal recorder 140 is connected to patient 160 via electrodes 145 and includes facility 120, one or more output devices 142, such as a monitor, printer, speaker, etc., and one or more input devices 144, such as settings controls, keyboard, biometric data reader, etc. Thus, as in this example, the facility can be configured to operate remotely from a patient and other diagnostics equipment and/or in conjunction with or part of the diagnostics equipment such as a wide-band biopotential measuring equipment (i.e., any device configured to capture unfiltered electrophysiological signals, including those with spectral components that are not altered). Accordingly, the facility can be configured to operate in real-time with the reading of physiological data and/or can be applied to previously recorded physiological data. Data providers 150, each of which includes a data store 152, can provide information for analysis or use by the facility such as, physiological patient data recorded off-site (e.g., at a hospital or clinic without access to a facility on premises, third-party data providers, etc.), feature vectors and/or feature generators produced or generated elsewhere, and so on. Network 170 represents communications links over which the various elements of environment 100 may communicate, such as the internet, a local area network, and so on.

[0034] In various examples, these computer systems and other devices can include server computer systems, desktop computer systems, laptop computer systems, netbooks, tablets, mobile phones, personal digital assistants, televisions, cameras, automobile computers, electronic media players, appliances, wearable devices, other hardware, and/or the like. In some embodiments, the facility 120 may operate on specific-purpose computing systems, such as wide-band biopotential measuring equipment (or any device configured to capture unfiltered electrophysiological signals, including electrophysiological signals with unaltered spectral components), electroencephalogram equipment, radiology equipment, sound recording equipment, and so on. In various examples, the computer systems and devices include one or more of each of the following: a central processing unit ("CPU") configured to execute computer programs; a computer memory configured

to store programs and data while they are being used, including a multithreaded program being tested, a debugger, the facility, an operating system including a kernel, and device drivers; a persistent storage device, such as a hard drive or flash drive configured to persistently store programs and data (e.g., firmware and the like); a computer-readable storage media drive, such as a floppy, flash, CD-ROM, or DVD drive, configured to read programs and data stored on a computer-readable storage medium, such as a floppy disk, flash memory device, CD-ROM, or DVD; and a network connection configured to connect the computer system to other computer systems to send and/or receive data, such as via the internet, a Local Area Network (LAN), a Wide Area Network (WAN), a point-to-point dial-up connection, a cell phone network, or another network and its networking hardware in various examples including routers, switches, and various types of transmitters, receivers, or computer-readable transmission media. While computer systems configured as described above may be used to support the operation of the facility, those skilled in the art will readily appreciate that the facility may be implemented using devices of various types and configurations, and having various components. Elements of the facility may be described in the general context of computer-executable instructions, such as program modules, executed by one or more computers or other devices. Generally, program modules include routines, programs, objects, components, data structures, and/or the like configured to perform particular tasks or implement particular abstract data types and may be encrypted. Furthermore, the functionality of the program modules may be combined or distributed as desired in various examples. Moreover, display pages may be implemented in any of various ways, such as in C++ or as web pages in XML (Extensible Markup Language), HTML (HyperText Markup Language), JavaScript, AJAX (Asynchronous JavaScript and XML) techniques, or any other scripts or methods of creating displayable data, such as the Wireless Access Protocol (WAP). Typically, the functionality of the program modules may be combined or distributed as desired in various embodiments, including cloud-based implementations, web applications, mobile applications for mobile devices, and so on.

[0035] The following discussion provides a brief, general description of a suitable computing environment in which the disclosed technology can be implemented. Although not required, aspects of the disclosed technology are

described in the general context of computer-executable instructions, such as routines executed by a general-purpose data processing device, e.g., a server computer, wireless device, or personal computer. Those skilled in the relevant art will appreciate that aspects of the disclosed technology can be practiced with other communications, data processing, or computer system configurations, including: internet or otherwise network-capable appliances, hand-held devices (including personal digital assistants (PDAs)), wearable computers (e.g., fitness-oriented wearable computing devices), all manner of cellular or mobile phones (including Voice over IP (VoIP) phones), dumb terminals, media players, gaming devices, multi-processor systems, microprocessor-based or programmable consumer electronics, set-top boxes, network PCs, mini-computers, mainframe computers, and the like. Indeed, the terms "computer," "server," "host," "host system," and the like are generally used interchangeably herein, and refer to any of the above devices and systems, as well as any data processor.

[0036] Aspects of the disclosed technology can be embodied in a special purpose computer or data processor, such as application-specific integrated circuits (ASIC), field-programmable gate arrays (FPGA), graphics processing units (GPU), manycore processors, and so on, that is specifically programmed, configured, or constructed to perform one or more of the computer-executable instructions explained in detail herein . While aspects of the disclosed technology, such as certain functions, are described as being performed exclusively on a single device, the disclosed technology can also be practiced in distributed computing environments where functions or modules are shared among disparate processing devices, which are linked through a communications network such as a Local Area Network (LAN), Wide Area Network (WAN), or the internet. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

[0037] Aspects of the disclosed technology may be stored or distributed on tangible computer-readable media, including magnetically or optically readable computer discs, hard-wired or preprogrammed chips (e.g., EEPROM semiconductor chips), nanotechnology memory, biological memory, or other computer-readable storage media. Alternatively, computer-implemented instructions, data structures, screen displays, and other data under aspects of the disclosed technology may be

distributed over the internet or over other networks (including wireless networks), on a propagated signal on a propagation medium (e.g., electromagnetic wave(s), sound wave, etc.) over a period of time, or they may be provided on any analog or digital network (packet switched, circuit switched, or other scheme). Furthermore, the term computer-readable storage medium does not encompass signals (e.g., propagating signals) or transitory media.

[0038] Figure 2 is a flow diagram illustrating the processing of a discover features component 121 in accordance with some embodiments of the disclosed technology. The discover features component is invoked by the facility to identify novel feature vectors based on selected patient data. In block 205, the component receives physiological signal data, such as raw signal data received directly from signal recorder, previously-generated physiological signal from another device or site, etc. Several techniques exist for collecting and analyzing physiological signals (e.g., electrophysiological signals, biosignals) from patients for diagnostic and other purposes including, for example, activity trackers, echocardiograms, wide-band biopotential measuring equipment, electroencephalograms, electromyograms, electrooculography, galvanic skin response, heart rate monitors, magnetic resonance imaging, magnetoencephalograms, mechanomyograms, wearable technology devices (e.g., FITBITs), and so on. While data provided by these systems can be helpful in identifying medical concerns and diagnosing medical conditions, they are often only a starting point for the diagnosis process. Moreover, given the specific nature of most of these systems, the data they analyze is often over-filtered to reduce complexity for the system itself or for, e.g., a technician, physician, or other health care provider (in such cases, to reduce visual complexity, etc.) thereby eliminating data that potentially have untapped diagnostic value. In block 210, the component invokes a process signal data component to process and transform the received signal data, which can produce multiple sets of data and transformed data. In block 215, the component sets a generation value equal to 0. In block 220, the component generates one or more feature generators by, for example, randomly generating an expression tree, randomly generating a set of weights for a neural network, randomly mutating one or more of a set of previously-generated feature generators, and so on. In block 225, the component invokes an apply feature generators component to apply the generated feature generators to

one or more sets of the processed signal data to produce a set of feature vectors. In block 230, the component invokes an identify novel feature vectors component to identify the most novel feature vectors from among the group of feature vectors generated by the feature generators. In block 235, the component stores the feature generators that produced the identified feature vectors in, for example, a feature generator store. In block 240, the component increments the generation variable. In decision block 245, if the generation variable is greater than or equal to a generation threshold, then the component completes, else the component continues at block 250. The component may also use other stopping conditions, such as a number of generations of feature generators that do not produce at least a threshold number of novel feature vectors. In block 250, the component copies and mutates the identified feature generators and then loops back to block 225 to apply the mutated feature generators to one or more sets of the processed signal data. As discussed above, the component may apply any type or types of mutations to a feature generator, such as applying multiple point mutations and/or a random recombination to one or more expression trees, randomly generating a set of connection weights for a neural network, and so on.

[0039] Figure 3 is a flow diagram illustrating the processing of a process component 122 in accordance with some embodiments of the disclosed technology. The process component is invoked by the discover features component to process and transform patient signal data. In blocks 305 through 365, the component loops through each signal (or data set) of a set of received signals (or set of data sets), each signal representing physiological data received from a patient. In block 310, the component pre-processes the received signal, such as applying one or more signal filters to the signal, performing a peak search on the data and discarding extraneous information, down-sampling the received signal, up-sampling the received signal, sub-sampling the received signal, converting an analog signal to a digital signal, converting image data to signal data, and so on. In block 315, the component stores the pre-processed signal in, for example, a patient data store. The signal data may be stored anonymously (i.e., without explicitly or implicitly identifying the corresponding patient, etc.). However, different instances of signal data associated with the same patient may be associated with an anonymized unique identifier so that multiple signals from a single patient can be used in

conjunction for training and diagnostic purposes. In block 320, the component extracts one or more values from the stored signal data. In block 325, the component stores the one or more extracted values. In block 330, the component identifies any transformations to be applied to the signal. For example, the facility may store an indication of a set of transformations or transformation functions (e.g., Fourier transforms, functions to apply to the signal, derivatives, partial derivatives, and so on) to apply to a particular signal. As another example, the facility may randomly select, from among a catalog of transformations, one or more transformations to apply to the signal data. In blocks 335 through 360, the component loops through each of the transformations and applies the transformation to the signal. In block 340, the component applies the transformation to the signal (e.g., calculating the third derivative with respect to a particular variable, calculating the result of a composite function generated by applying one function to the signal data (i.e., a function representative of the signal data), etc.). In block 345, the component stores the transformed signal data in, for example, a patient data store. In block 350, the component extracts one or more values from the transformed signal data. In block 355, the component stores the one or more extracted values. In block 360, if there are any identified transformations yet to be applied, then the component selects the next transformation and loops back to block 335 to apply the transformation to the signal data, else the component continues at block 365. In block 365, if there are any signals yet to be analyzed, then the component selects the next signal and loops back to block 305 to process the next signal, else the component completes.

[0040] Figure 4 is a flow diagram illustrating the processing of an apply feature generators component 123 in accordance with some embodiments of the disclosed technology. The apply feature generators component is invoked by the discover features component 121 to apply a set of one or more feature generators to signal data, such as pre-processed and transformed signal data, modeled signal data, etc. In blocks 410 through 470, the component loops through each of a received set of feature generators and applies the feature generator to each signal in a received set of signal data. For example, the received signal data can include multiple sets of signal data for each of multiple patients, multiple transformations of that data, and so on. In blocks 420 through 450, the component loops through each of the signals to

apply the feature generators to the signal data. In block 430, the component applies the currently-selected feature generator to the currently-selected signal data. For example, the component may apply the feature generator to each of a pre-processed version of the currently-selected data signal and any transformed version of that data. As another example, the component "plugs in" or substitutes coefficients generated by modeled signal data into a feature generator with a set of variables to produce an output feature value. As another example, the component can apply one or more elements of modeled signal data to a neural network to produce an output feature value. In block 440, the component stores the output value. In block 450, if there are any signals yet to be analyzed, then the component selects the next signal and loops back to block 420 to process the next signal, else the component continues at block 460. In block 460, the component generates a feature vector that includes each of the generated feature values and stores the feature vector in association with the feature generator in, for example, a feature vector store. For example, the feature vector may comprise an array of features and a link to, or identifier of, the corresponding feature generator. The component may also associate the feature vector with the signal data used to generate the feature vector. In block 470, if there are any feature generators yet to be processed, then the component selects the next feature generator and loops back to block 410 to process the feature generator, else the component completes.

[0041] Figure 5 is a flow diagram illustrating the processing of an identify novel feature vectors component 124 in accordance with some embodiments of the disclosed technology. In this example, the facility receives a set of feature vectors and, for each feature vector, information related to the corresponding feature generator, such as an identifier for the feature generator. In block 505, the component collects a comparison set of feature vectors that includes, for example, feature vectors generated by feature generators of earlier generations that were found to be novel and the feature vectors generated by a current generation of feature vectors. For example, the component can randomly select a set of novel feature vectors from a feature store. In some cases, a request to retrieve feature vectors includes upper and lower limits on the number of features values for each feature vector to be retrieved, such as no less than 50 (lower threshold) and no more than 5000 (upper threshold). In blocks 510 through 540, the component loops

through each feature vector of a current generation of feature generators to determine how different each of their corresponding feature vectors is to each of the feature vectors of the comparison set of feature vectors. In blocks 515 through 530, the component loops through each feature vector of the comparison set of feature vectors to compare each feature vector to the feature vector of the currently-selected feature generator. In block 520, the component calculates a difference value between the currently-selected feature vector of the comparison set and the feature vector of the currently-selected feature generator. For example, the component can calculate a distance value between each of the feature vectors. In block 525, the component stores the calculated difference value. In block 530, if there are any feature vectors yet to be compared, then the component selects the next feature vector and loops back to block 515 to process the feature vector, else the component continues at block 535. In block 535, the component calculates a novelty score for the currently-selected feature generator based on the stored difference values, such as an average or maximum distance, and stores the novelty score in association with the feature generator (e.g., in a feature generator store). In block 540, if there are any feature generators yet to be assessed, then the component selects the next feature generator and loops back to block 515 to process the feature generator, else the component continues at block 545. In blocks 545 through 560, the component tests whether each of the feature vectors is novel, based on the calculated novelty scores, and identifies any corresponding feature generators. In decision block 550, if the novelty score for the currently-selected feature generator is greater than a novelty threshold, then the component continues at block 555, else the component continues at block 560. The novelty threshold may be generated or determined in any number of ways, such as receiving a novelty threshold from a user, calculating a novelty threshold based on the set of novelty scores (e.g., average, average plus 25%, top n (where n is provided by a user or generated automatically by the facility), top tenth percentile), and so on. In this manner, the novelty threshold may change dynamically (e.g., from generation to generation) based on, for example, the number of generations without a new feature generator that exceeds the current novelty threshold to ensure that the facility is producing and testing new feature generators and corresponding features. In block 555, the component identifies the currently-selected feature vector as novel. In block 560, if there are any feature vectors yet to be processed, then the component

selects the next feature vector and loops back to block 545 to process the feature vector, else the component completes.

[0042] Figure 6 is a flow diagram illustrating the processing of a discover genomes component 126 in accordance with some embodiments of the disclosed technology. The facility invokes the discover genomes component to generate and analyze genomes for use by machine learning algorithms. In block 610, the facility initializes a generation variable equal to 0. In block 620, the component determines a number (*n*) of genomes to generate based on, for example, user input, system parameters, or randomly. In block 630, the component invokes a spawn genome component *n* time(s) to spawn the appropriate number of genomes. In block 640, the component invokes an identify high-performing genomes component to identify, from among the spawned genomes, the genomes that have a fitness score that exceeds a fitness threshold. In block 650, the component increments the generation variable. In decision block 660, if the generation variable is greater than or equal to a generation threshold, then processing of the component completes, else the component continues at block 670. In block 670, the component mutates the high-performing genomes and then loops back to block 640 to identify high-performing genomes from among the mutated genomes (e.g., the mutated genomes having fitness scores that exceed a fitness threshold). The component may mutate the genome by adding, changing, or removing (or any combination thereof) one or more elements of the variable-length genome. For example, the component may mutate one genome by replacing one feature with another feature and adding a new feature to the mutated genome. In another example, the component may select a new machine learning algorithm to associate with the genome. In this case, the component may also remove or mutate any irrelevant machine learning algorithm parameters and/or replace them with machine learning parameter values for the newly selected machine learning algorithm. As another example, genomes may use sexual reproduction techniques as a form of mutation by randomly selecting elements of multiple genomes and combining those elements to form a new genome. Moreover, one or more elements of a genome may be configured so that they remain fixed (i.e., are not changed) during the evolutionary process described herein.

[0043] Figure 7 is a flow diagram illustrating the processing of a spawn genome component 126 in accordance with some embodiments of the disclosed technology. The discover genomes component 125 invokes the spawn genome component to generate a genome identifying any number of features, machine learning parameters, and/or machine learning algorithms. In block 710, the component identifies a set of available features, such as features referenced in one or more feature generator stores. In block 720, the component determines the number of features to include in the genome to be generated. For example, the component may determine the number of features to include in the genome to be generated based on user input, system parameters, or randomly. In block 730, the component randomly selects, from among the identified features, the determined number of features. In block 740, the component replaces correlated features from among the selected features with randomly selected features. In block 750, the component identifies a set of available machine learning parameters. For example, the component may identify, for each machine learning algorithm available to the facility, a set of parameters associated with that machine learning algorithm, which may be stored in a list or other data structure available to the component (e.g., a machine learning parameter store). In some cases, a genome may be generated for a single machine learning algorithm (or fixed set of machine learning algorithms). In this case, the component may identify only those machine learning parameters (or a proper subset thereof) that are associated with the single or fixed set of machine learning algorithms. In other cases, a genome may include an element that identifies a machine learning algorithm and that can be mutated. In this case, the component may identify any or all machine learning parameters of machine learning algorithms that are within the scope of this mutation (i.e., the parameters of the machine learning algorithms that the genome and its descendants may be associated with to train models during the evolutionary process described herein). In block 760, the component determines the number of machine learning parameters to include in the genome to be generated. For example, the component may determine the number of machine learning parameters to include in the genome to be generated based on user input, system parameters, or randomly. For example, one genome may include each and every machine learning parameter associated with a particular machine learning algorithm or a set of machine learning algorithms while another genome includes only a proper subset of machine learning parameters associated with a

particular machine learning algorithm. In block 770, the component randomly selects, from among the identified machine learning parameters, the determined number of machine learning parameters and assigns a value to the parameter based on any associated constraints, such as randomly selecting a value between a minimum value and a maximum value associated with the parameter. In block 780, the component stores each of the selected features and machine learning parameters in a genome data structure and then returns the genome data structure.

[0044] Figure 8 is a flow diagram illustrating the processing of an identify high-performing genomes component 127 in accordance with some embodiments of the disclosed technology. The discover genomes component invokes the identify high-performing genomes component to identify, from among a group of genomes, the genomes that have a corresponding fitness score that exceeds a fitness threshold (i.e., the genomes that are "high-performing"). In blocks 810 through 850, the component loops through a set of genomes provided to the component, such a first generation set of genomes, a mutated set of genomes, or some combination thereof. In block 820, the component trains one or more models using the currently-selected genome, including its features, machine learning parameters, and any specified machine learning algorithm. In the event that a machine learning parameter of the genome is not associated with the machine learning algorithm used to train the model, that machine learning parameter may be ignored. Similarly, if a particular machine learning algorithm requires as input a particular machine learning parameter that is not included in the currently-selected genome, the facility (or the machine learning algorithm itself) may provide a default value retrieved from, for example, a machine learning parameter store. In block 830, the component generates a validation or fitness score for the currently-selected genome by, for example, applying the trained model to a set of validation data and assessing the ability of the trained model to correctly identify or classify subjects from among the validation data. In block 840, the component stores the generated score(s) and/or an aggregation thereof produced for the trained models in association with the currently-selected genome. In block 850, if there are any genomes yet to be scored, then the component selects the next genome and loops back to block 810, else the component continues at block 860. In blocks 860 through 890, the component assesses the score generated for each genome and selects the "best" genomes for

mutation. In this example, the "best" genomes are those that produce validation or fitness scores that exceed a fitness threshold. In decision block 870, if the score generated for the currently-selected genome exceeds a fitness threshold, then the component continues at block 880, else the component continues at block 890. In block 880, the component flags the currently-selected genome for mutation. In some embodiments, the component may select genomes for mutation based on criteria other than, or in addition to, fitness scores. For example, the component may use novelty scores or other scores to select genomes for mutation. In some cases, the component may employ a tournament selection process in which a number of genomes are randomly selected from the population and the genome with the highest score among this "tournament" is selected for reproduction. In this example, if only low scoring genomes appear in a tournament, a low-scoring genome will be selected for reproduction. In block 890, if there are any genomes yet to be processed, then the component selects the next genome and loops back to block 860, else the component returns the flagged genomes and processing of the component completes.

[0045] From the foregoing, it will be appreciated that specific embodiments of the disclosed technology have been described herein for purposes of illustration, but that various modifications may be made without deviating from the scope of the disclosed technology. For example, the disclosed techniques can be applied to fields outside of the medical field, such as predicting weather patterns, geological activity, or any other field in which predictions are made based on sampled input data. To reduce the number of claims, certain aspects of the disclosed technology are presented below in certain claim forms, but applicants contemplate the various aspects of the disclosed technology in any number of claim forms. Accordingly, the disclosed technology is not limited except as by the appended claims.

CLAIMS

We claim:

1. A system, having a memory and a processor, for discovering machine learning genomes, the system comprising:

a first component configured to generate a plurality of genomes, wherein each genome identifies at least one feature and at least one parameter for at least one machine learning algorithm, wherein generating a first genome of the plurality of genomes comprises:

randomly selecting, from among a set of features, one or more of the features,

randomly selecting, from among a set of parameters for at least one machine learning algorithm, one or more of the parameters, and

assigning at least one random value to each of the selected parameters;

a second component configured to, for each generated genome,

train a one or more models using the generated genome, and

for each model trained using the generated genome,

calculate a fitness score for the trained model at least in part by applying the trained model to a validation data set, and

produce a fitness score for the generated genome based at least in part on the fitness scores generated for the models trained using the generated genome;

a third component configured to identify, from among the generated genomes, a plurality of genomes having a fitness score that exceeds a fitness threshold; and

a fourth component configured to, for each of the identified genomes, mutate the identified genome,

wherein at least one of the components comprises computer-executable instructions stored in the memory for execution by the system.

2. The system of claim 1, further comprising:

a fifth component configured to, for the first genome comprising a first set of features, identify correlated features from among the first set of features at least in part by:

for each feature of the first set of features,

applying a feature generator associated with the feature to a training set of data to generate a feature vector for the feature,

for at least one pair of feature vectors,

calculating a distance between each feature vector of the pair of feature vectors,

determining that the calculated distance is less than a distance threshold,

in response to determining that the calculated distance is less than a distance threshold, removing, from the first genome, a feature corresponding to at least one feature vector of the pair of feature vectors,

wherein each feature vector includes, for each of a plurality of patients, a single value generated by applying a first feature generator to at least one representation of physiological data representative of the patient.

3. The system of claim 2, wherein the removing, from the first genome, of at least one feature corresponding to at least one feature vector of a first pair of feature vectors comprises:

randomly selecting one of feature vector of the first pair of feature vectors,

identifying, from among the features of the first genome, a feature corresponding to the randomly selected feature vector; and

removing, from the first genome, the identified feature.

4. The system of claim 1, further comprising:

a fifth component configured to, for the first genome comprising a first set of features, generate a graph comprising a vertex for each feature of the first set of features;

a sixth component configure to generate an edge between vertices whose corresponding features have a correlation value that exceeds a correlation threshold or a distance value that is less than a distance threshold; and

a seventh component configured to remove vertices from the graph until no connected vertices remain in the graph.

5. The system of claim 1, further comprising:

a machine configured to receive physiological signal data from at least one patient;

a fifth component configured to, for each patient,

apply at least one of the trained models to at least a portion of the physiological signal data received for the patient by the machine, and

generate a prediction for the patient based at least in part on the application of the at least one of the trained models to at least a portion of the received physiological signal.

6. A method, performed by a computing system having a memory and a processor, for discovering machine learning genomes, the method comprising:

generating, with the processor, a plurality of genomes, wherein each genome identifies at least one feature and at least one parameter for at least one machine learning algorithm;

for each generated genome,

training at least one model using the generated genome, and

producing a fitness score for the genome based at least in part on the trained at least one model;

identifying, from among the generated genomes, at least one genome having a fitness score that exceeds a fitness threshold; and

mutating each identified genome.

7. The method of claim 6, wherein generating a first genome of the plurality of genomes comprises:

randomly selecting, from among a set of features, one or more of the features;

randomly selecting, from among a set of parameters for at least one machine learning algorithm, one or more of the parameters; and

assigning at least one value to each of the selected parameters.

8. The method of claim 7, wherein generating the first genome further comprises:

for each feature of the randomly selected features,

retrieving a feature vector for the feature based at least in part on a feature generator associated with the feature and a training set of data;

identifying pairs of correlated feature vectors from among the generated feature vectors; and

for each identified pair of correlated feature vectors,

identifying one feature vector of the pair of correlated feature vectors,

removing, from the first genome, the feature associated with the feature generator used to generate the identified feature vector;

randomly selecting, from among the set of features, a feature to add to the first genome, and

adding the randomly selected feature to the first genome.

9. The method of claim 8, wherein identifying pairs of correlated feature vectors comprises:

for each pair of feature vectors,

calculating a distance metric for the pair of feature vectors, and

determining whether the distance metric calculated for the pair of feature vectors is less than a distance threshold,

wherein the distance threshold is determined based at least in part on the calculated distance metrics determined for each pair of feature vectors.

10. The method of claim 6, wherein producing a fitness score for a first genome comprises:

identifying a number of false positives generated by applying, to two or more validation data sets, a model trained using the first genome; and

identifying a number of false negatives generated by applying, to two or more validation data sets, a model trained using the first genome.

11. The method of claim 6, wherein producing a fitness score for a first genome comprises:

generating, for at least one model trained using the first genome, a receiver operating characteristic curve; and

calculating an area under the generated receiver operating characteristic curve.

12. The method of claim 6, wherein producing a fitness score for a first genome comprises calculating, for at least one model trained using the first genome, one or more of the errors selected from the group comprising: mean squared prediction error, mean absolute error, interquartile error, and log loss error, receiver-operator characteristic curve error, and f-score error.

13. The method of claim 6, wherein mutating a first identified genome comprises:

selecting at least one feature of the first identified genome; and

removing, from the first identified genome, each of the selected features of the first identified genome.

14. The method of claim 6, wherein mutating the first identified genome further comprises:

randomly selecting, from among the set of features, a plurality of the features; and

adding, to the first identified genome, each of the randomly selected plurality of features.

15. The method of claim 6, wherein mutating a first identified genome comprises:

modifying at least one feature of the first identified genome.

16. The method of claim 6, wherein mutating a first identified genome comprises:

modifying at least one machine learning algorithm parameter of the first identified genome.

17. A computer-readable medium storing instructions that, if executed by a computing system having a memory and a processor, cause the computing system to perform a method for discovering machine learning genomes, the method comprising:

generating a plurality of genomes, wherein each genome identifies at least one feature;

for each generated genome,

training at least one model using the generated genome, and

producing a fitness score for the genome based at least in part on the trained at least one model; and

identifying, from among the generated genomes, one or more genomes having a fitness score that exceeds a fitness threshold.

18. The computer-readable medium of claim 17, wherein each genome further identifies at least one parameter for at least one machine learning algorithm.

19. The computer-readable medium of claim 17, the method further comprising:

mutating each identified genome having a fitness score that exceeds the fitness threshold.

20. The computer-readable medium of claim 17, the method further comprising:

computing the fitness threshold at least in part by, determining an overall fitness score based on the fitness scores produced for each of the generated genomes.

21. The computer-readable medium of claim 17, the method further comprising:

computing the fitness threshold at least in part by, determining a n-th percentile of fitness scores based on the fitness scores produced for each of the generated genomes.

22. The computer-readable medium of claim 17, the method further comprising:

computing the fitness threshold at least in part by, determining an n-th highest fitness score from among the fitness scores produced for each of the generated genomes.

23. The computer-readable medium of claim 17, further comprising:

for each model trained using the first genome,

calculating a fitness score for the model trained using the first genome; and

aggregating the fitness scores calculated for the models trained using the first genome.

24. The computer-readable medium of claim 23, wherein aggregating the fitness scores calculated for the models trained using the first genome comprises calculating an average value of the fitness scores calculated for the models trained using the first genome.

25. The computer-readable medium of claim 17, the method further comprising:

for each of the plurality of genomes identified from among the generated genomes,

mutating the identified genome,

training at least one model using the mutated genome, and

producing a fitness score for the mutated genome based at least in part on the at least one model trained using the mutated genome.

26. One or more computer memories collectively storing a genome data structure, wherein the genome data structure comprises:

a plurality of features, each feature identifying at least one feature generator, and

a plurality of parameters for at least one machine learning algorithm,

wherein the genome data structure is configured to be used to train at least one model in accordance with the plurality of feature and at least one machine learning algorithm.

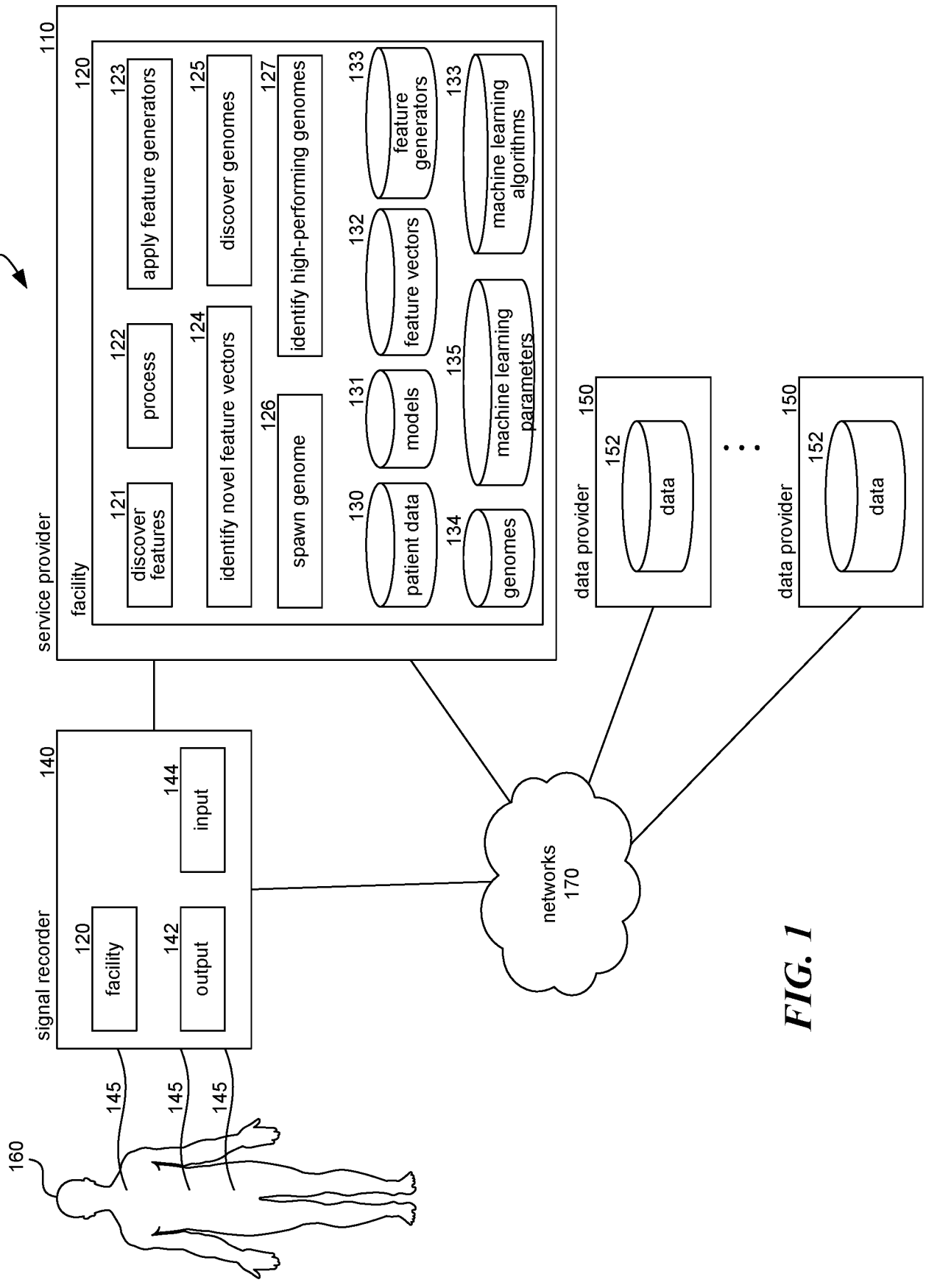
100
service provider
110
facility
120
discover features
121
process
122
apply feature generators
123
identify novel feature vectors
124
discover genomes
125
spawn genome
126
identify high-performing genomes
127
patient data
130
models
131
feature vectors
132
feature generators
133
genomes
134
machine learning parameters
135
machine learning algorithms
133
signal recorder
140
facility
120
output
142
input
144
networks
170
data provider
150
data
152
data provider
150
data
152
145
145
145
160

FIG. 1

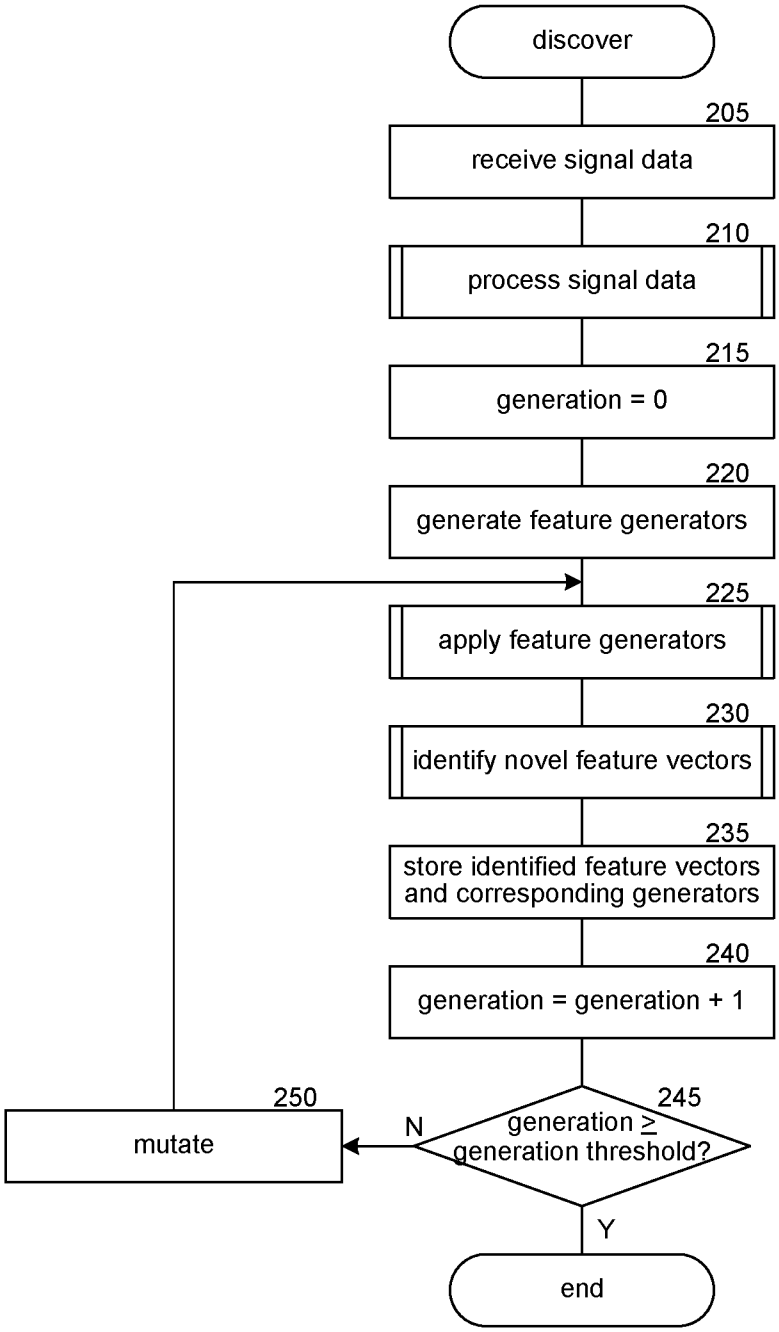
discover
205
receive signal data
210
process signal data
215
generation = 0
220
generate feature generators
225
apply feature generators
230
identify novel feature vectors
235
store identified feature vectors and corresponding generators
240
generation = generation + 1
245
generation ≥ generation threshold?
N
250
mutate
Y
end

FIG. 2

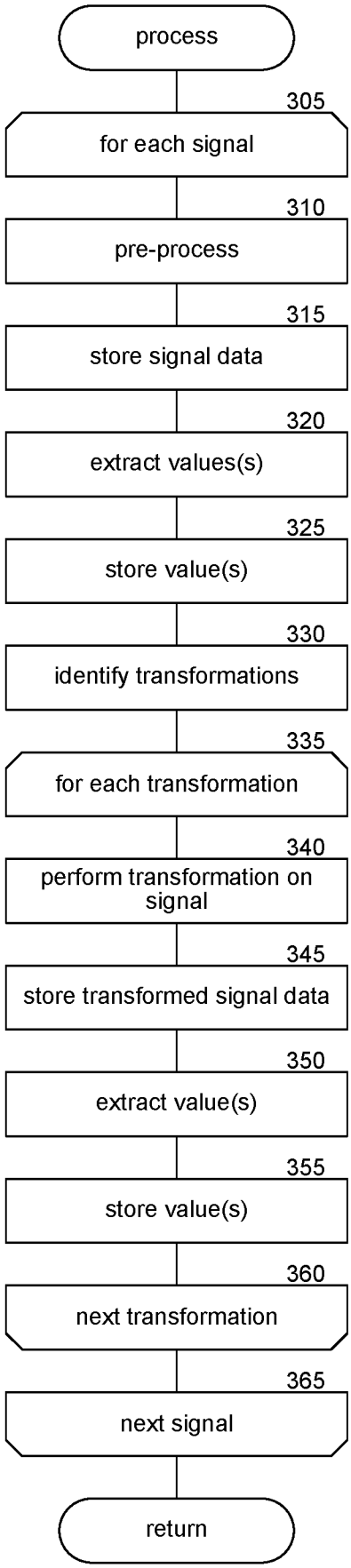
process
305
for each signal
310
pre-process
315
store signal data
320
extract values(s)
325
store value(s)
330
identify transformations
335
for each transformation
340
perform transformation on signal
345
store transformed signal data
350
extract value(s)
355
store value(s)
360
next transformation
365
next signal
return

FIG. 3

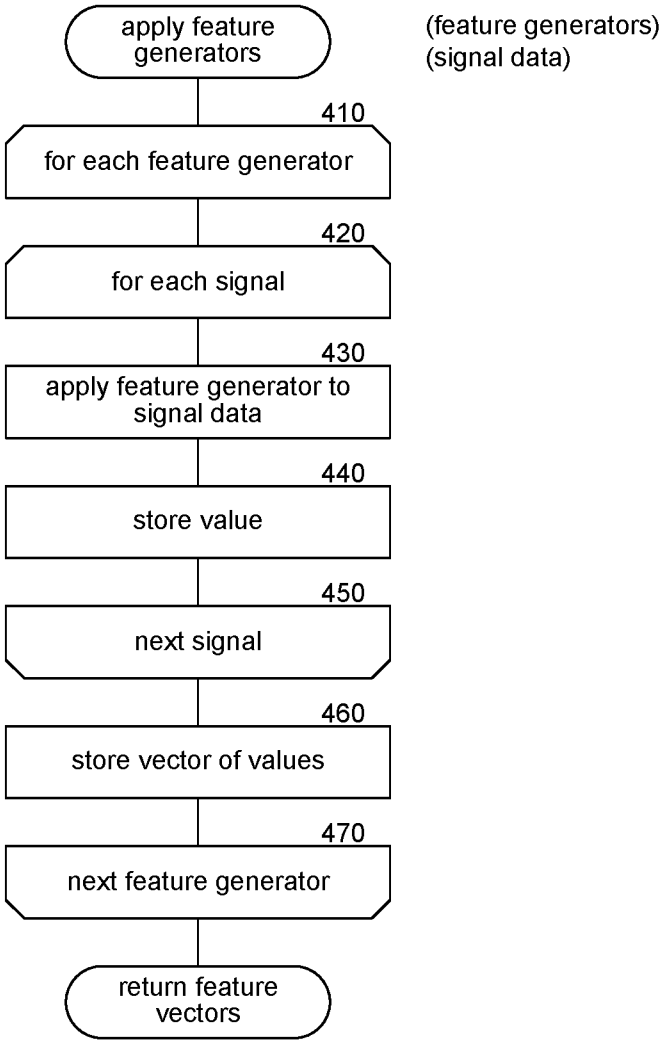
apply feature generators
(feature generators)
(signal data)
410
for each feature generator
420
for each signal
430
apply feature generator to signal data
440
store value
450
next signal
460
store vector of values
470
next feature generator
return feature vectors

FIG. 4

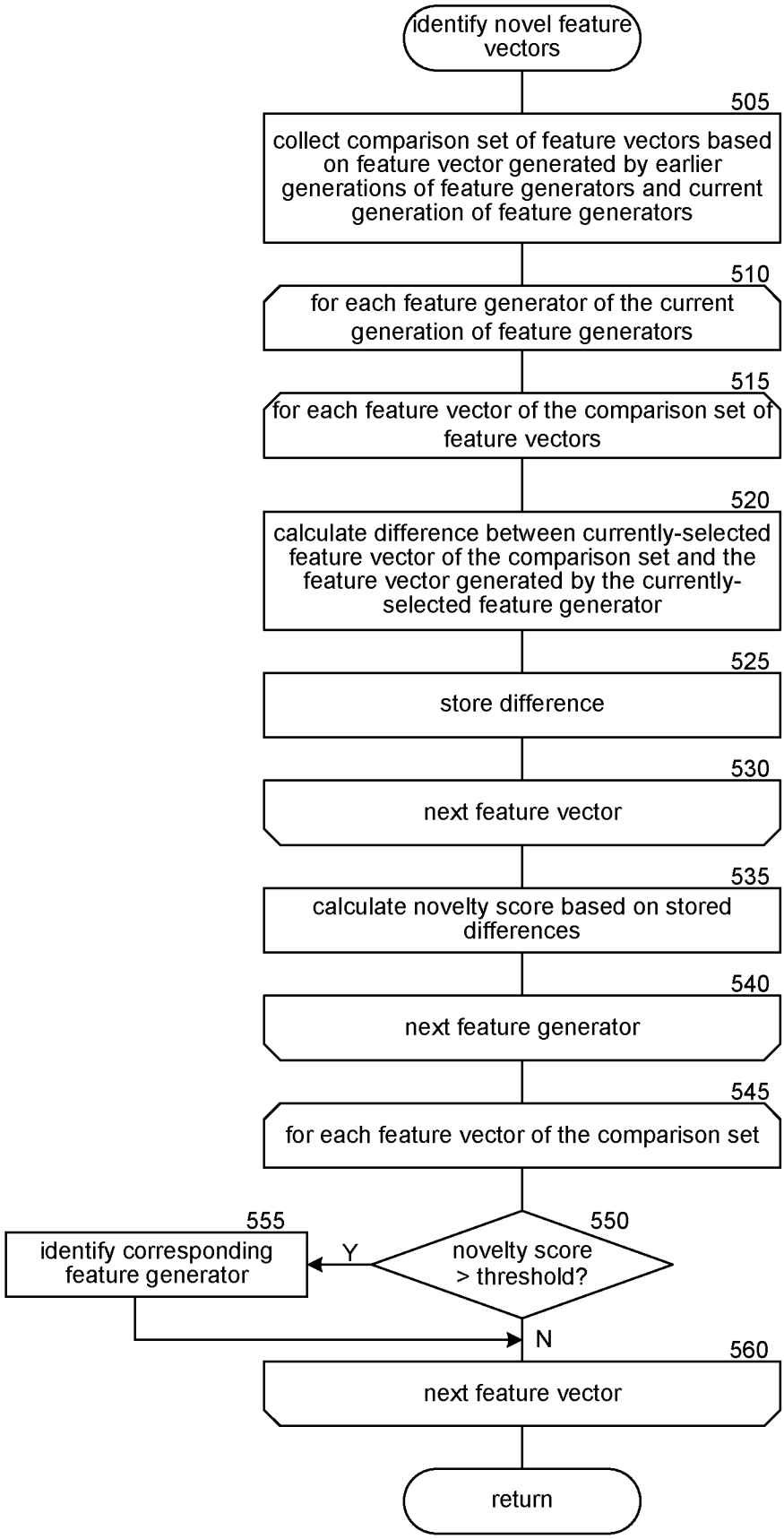
identify novel feature vectors
505
collect comparison set of feature vectors based on feature vector generated by earlier generations of feature generators and current generation of feature generators
510
for each feature generator of the current generation of feature generators
515
for each feature vector of the comparison set of feature vectors
520
calculate difference between currently-selected feature vector of the comparison set and the feature vector generated by the currently-selected feature generator
525
store difference
530
next feature vector
535
calculate novelty score based on stored differences
540
next feature generator
545
for each feature vector of the comparison set
550
novelty score > threshold?
Y
555
identify corresponding feature generator
N
560
next feature vector
return

FIG. 5

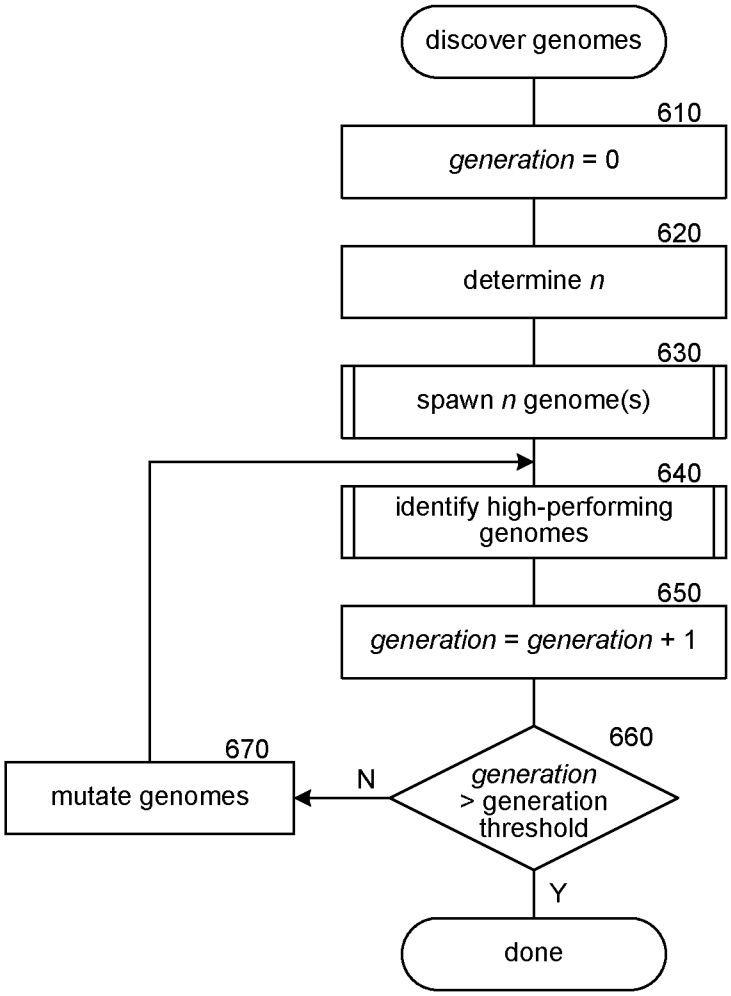
discover genomes
610
generation = 0
620
determine n
630
spawn n genome(s)
640
identify high-performing genomes
650
generation = generation + 1
660
generation > generation threshold
N
670
mutate genomes
Y
done

FIG. 6

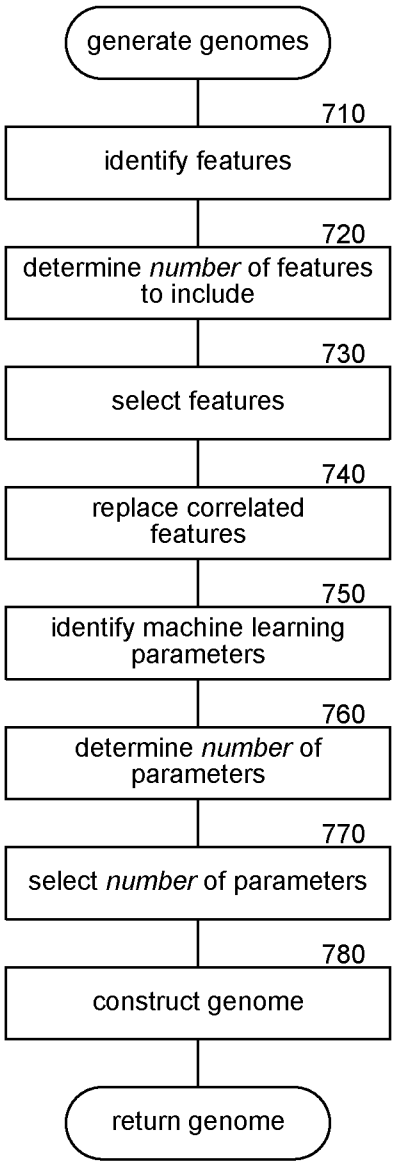
generate genomes
710
identify features
720
determine *number* of features to include
730
select features
740
replace correlated features
750
identify machine learning parameters
760
determine *number* of parameters
770
select *number* of parameters
780
construct genome
return genome

FIG. 7

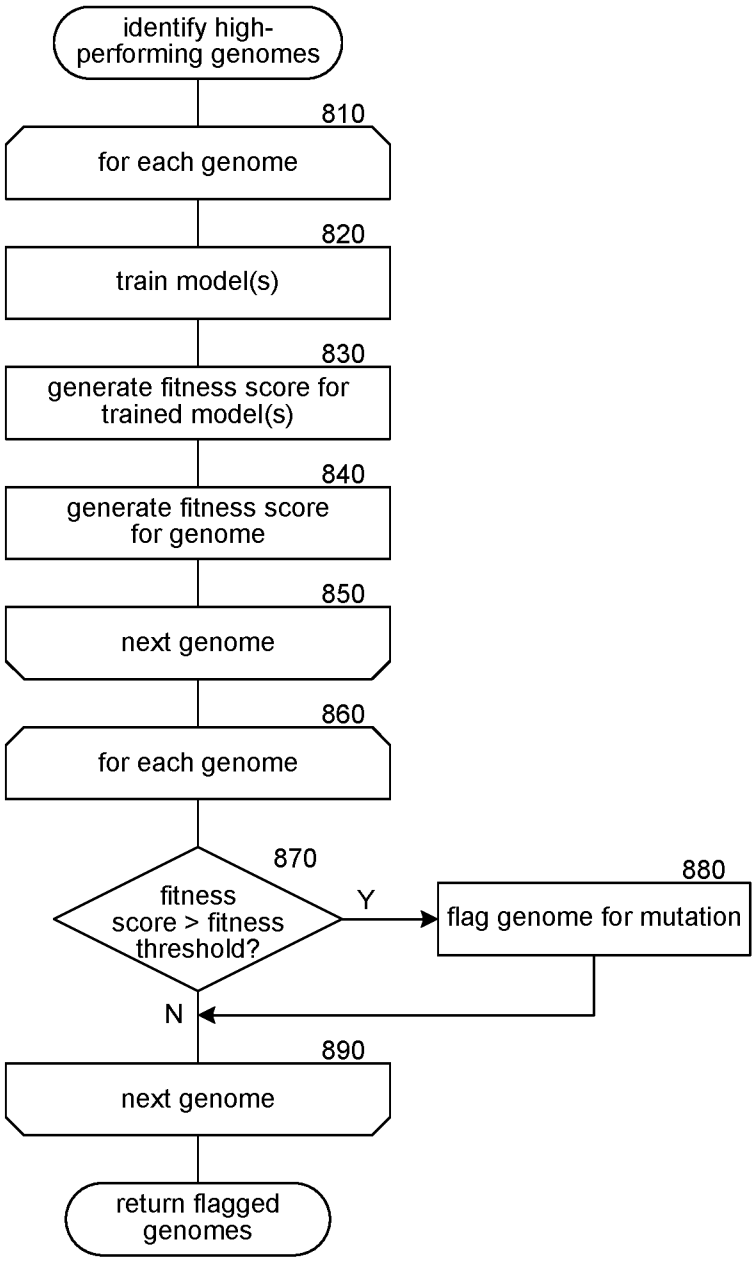
identify high-performing genomes
810
for each genome
820
train model(s)
830
generate fitness score for trained model(s)
840
generate fitness score for genome
850
next genome
860
for each genome
870
fitness score > fitness threshold?
Y
880
flag genome for mutation
N
890
next genome
return flagged genomes

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IB2018/000929

A. CLASSIFICATION OF SUBJECT MATTER
IPC: ***G06F 15/18*** (2006.01) , ***G06N 3/12*** (2006.01)

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)
IPC: G06F 15/18 (2006.01) , G06N 3/12 (2006.01)

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

Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used)
Databases: Questel Orbit, Google Patents
Keywords: machine; learning; train+; model+; genom+; graph; mutat+; train+; fitness; vert+; edge; score+; algorithm; false positives; false negatives

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	US20060230006 A1, Buscema, 12 October 2006 (12-10-2006) * Fig. 5; claim 25; paragraphs [0066-0101]	1, 5-7 and 13-26 2-4 and 8-12
Y	US20130103620 A1, Yoon et al., 25 April 2013 (25-04-2013) * paragraphs [0006, 0058-0062]	2, 3 and 8
Y	US20040204957 A1, Afeyan et al., 14 October 2004 (14-10-2004) * paragraphs [0163-0169]	4 and 9
Y	US20100063948 A1, Virkar et al., 11 March 2010 (11-03-2010) * paragraphs [0048, 0131-0133, 0142, 0165]	10-12

☐ 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 06 December 2018 (06-12-2018)	Date of mailing of the international search report 07 January 2019 (07-01-2019)
Name and mailing address of the ISA/CA Canadian Intellectual Property Office Place du Portage I, C114 - 1st Floor, Box PCT 50 Victoria Street Gatineau, Quebec K1A 0C9 Facsimile No.: 819-953-2476	Authorized officer Leslie Yeow (819) 639-8372

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/IB2018/000929

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US2006230006A1	12 October 2006 (12-10-2006)	US2006230006A1	12 October 2006 (12-10-2006)
		US7711662B2	04 May 2010 (04-05-2010)
		EP1586076A2	19 October 2005 (19-10-2005)
		JP2006518062A	03 August 2006 (03-08-2006)
		WO2004063831A2	29 July 2004 (29-07-2004)
US2013103620A1	25 April 2013 (25-04-2013)	US2013103620A1	25 April 2013 (25-04-2013)
		US9275304B2	01 March 2016 (01-03-2016)
		KR20130042783A	29 April 2013 (29-04-2013)
		KR101725121B1	12 April 2017 (12-04-2017)
US2004204957A1	14 October 2004 (14-10-2004)	US2004204957A1	14 October 2004 (14-10-2004)
		US7016882B2	21 March 2006 (21-03-2006)
		AU2002246919B2	30 August 2007 (30-08-2007)
		CA2428079A1	25 July 2002 (25-07-2002)
		EP1334458A2	13 August 2003 (13-08-2003)
		EP2631856A2	28 August 2013 (28-08-2013)
		JP2011238284A	24 November 2011 (24-11-2011)
		JP5639544B2	10 December 2014 (10-12-2014)
		JP2004529406A	24 September 2004 (24-09-2004)
		JP2008097637A	24 April 2008 (24-04-2008)
		JP2014059899A	03 April 2014 (03-04-2014)
		USRE46178E	11 October 2016 (11-10-2016)
		US2003088458A1	08 May 2003 (08-05-2003)
		US7177851B2	13 February 2007 (13-02-2007)
		US2006080268A1	13 April 2006 (13-04-2006)
		US7610249B2	27 October 2009 (27-10-2009)
		US2007282666A1	06 December 2007 (06-12-2007)
		US7730002B2	01 June 2010 (01-06-2010)
		WO02057986A2	25 July 2002 (25-07-2002)
US2010063948A1	11 March 2010 (11-03-2010)	US2010063948A1	11 March 2010 (11-03-2010)
		US8386401B2	26 February 2013 (26-02-2013)
		US2013238533A1	12 September 2013 (12-09-2013)
		US9082083B2	14 July 2015 (14-07-2015)
		US2015286955A1	08 October 2015 (08-10-2015)
		WO2010030794A1	18 March 2010 (18-03-2010)