



US 20240252048A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2024/0252048 A1
Tran (43) Pub. Date: Aug. 1, 2024

(54) AI ASSISTANCE SYSTEM

A61B 5/145 (2006.01)

(71) Applicant: Bao Tran, Saratoga, CA (US)

A61B 5/16 (2006.01)

(72) Inventor: Bao Tran, Saratoga, CA (US)

A61B 5/283 (2006.01)

(21) Appl. No.: 18/620,946

A61B 5/291 (2006.01)

(22) Filed: Mar. 28, 2024

A61B 5/378 (2006.01)*G05B 13/02* (2006.01)*G05B 13/04* (2006.01)*G06N 20/00* (2006.01)*H04R 25/00* (2006.01)**Related U.S. Application Data**

(63) Continuation of application No. 17/979,756, filed on Nov. 2, 2022, which is a continuation of application No. 17/110,087, filed on Dec. 2, 2020, now Pat. No. 11,596,316, which is a continuation of application No. 16/286,518, filed on Feb. 26, 2019, now Pat. No. 10,905,337.

(52) **U.S. Cl.**
CPC *A61B 5/02055* (2013.01); *A61B 1/00096* (2022.02); *A61B 1/00194* (2022.02); *A61B 1/05* (2013.01); *A61B 1/227* (2013.01); *A61B 5/026* (2013.01); *A61B 5/0538* (2013.01); *A61B 5/1032* (2013.01); *A61B 5/1117* (2013.01); *A61B 5/1118* (2013.01); *A61B 5/12* (2013.01); *A61B 5/14546* (2013.01); *A61B 5/165* (2013.01); *A61B 5/283* (2021.01); *A61B 5/291* (2021.01); *A61B 5/378* (2021.01); *A61B 5/4803* (2013.01); *A61B 5/6817* (2013.01); *A61B 5/7267* (2013.01); *G05B 13/027* (2013.01); *G05B 13/042* (2013.01); *G06N 20/00* (2019.01); *H04R 25/604* (2013.01); *H04R 25/652* (2013.01); *A61B 5/0816* (2013.01); *A61B 2560/0247* (2013.01); *A61B 2562/0219* (2013.01); *H04R 2225/025* (2013.01); *H04R 2225/77* (2013.01)**Publication Classification**(51) **Int. Cl.***A61B 5/0205* (2006.01)
A61B 1/00 (2006.01)
A61B 1/05 (2006.01)
A61B 1/227 (2006.01)
A61B 5/00 (2006.01)
A61B 5/026 (2006.01)
A61B 5/0538 (2006.01)
A61B 5/08 (2006.01)
A61B 5/103 (2006.01)
A61B 5/11 (2006.01)
A61B 5/12 (2006.01)**ABSTRACT**

(57) Systems and methods for assisting user with an ear-worn/earable device and a learning machine language model to assist the user.



FIG. 1

Scan ear (1)
Generate 3D model of ear structure (2)
3D Print a Negative of the ear structure with provisioned space for electronics (3)
Form a flexible ear insert from the ear structure negative (4)
Add amplifier to the ear insert (5)
During use, snugly push the ear insert toward the ear canal (6)
Render sound (7)
Capture medical data as biomarkers including ECG, EGG, physical condition, heart blood flow data, temperature, among others (8)

FIG. 2A

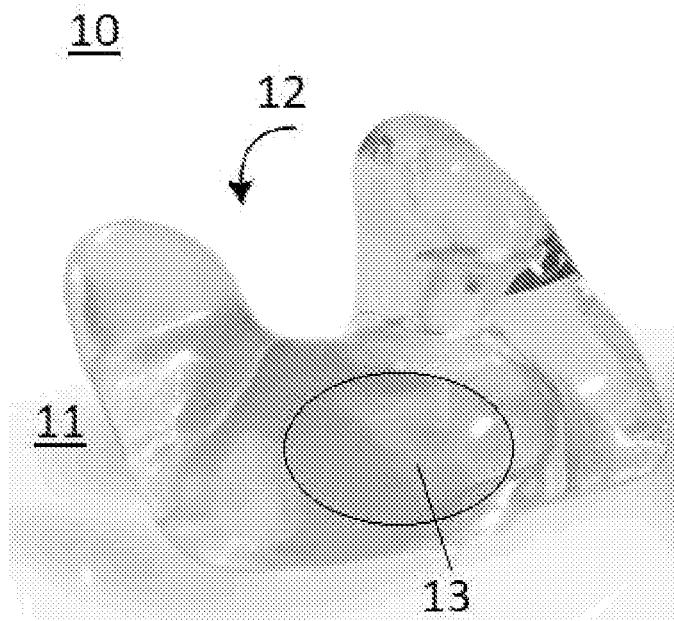


FIG. 2B



FIG. 3A

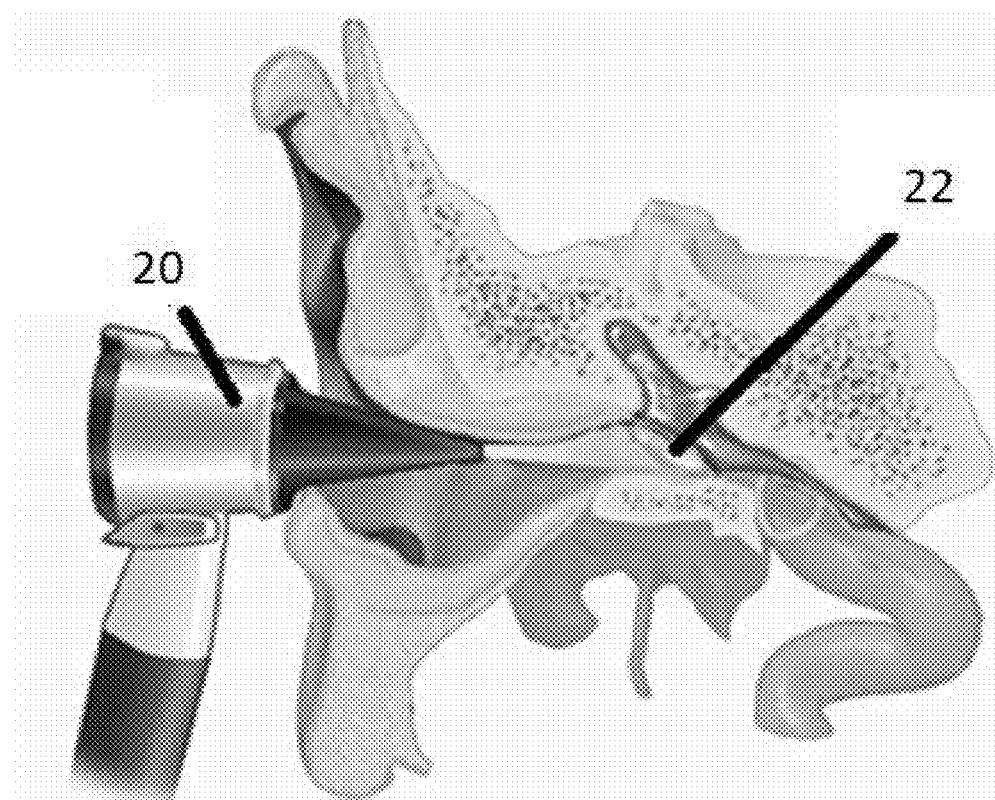


FIG. 3B



FIG. 3C

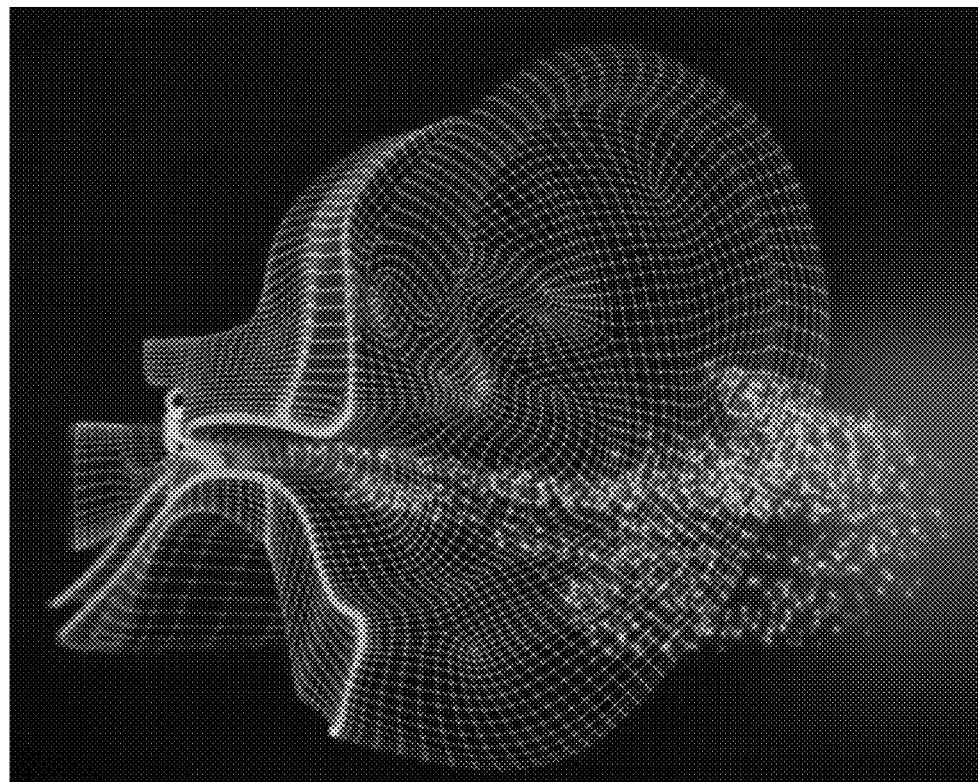


FIG. 3D

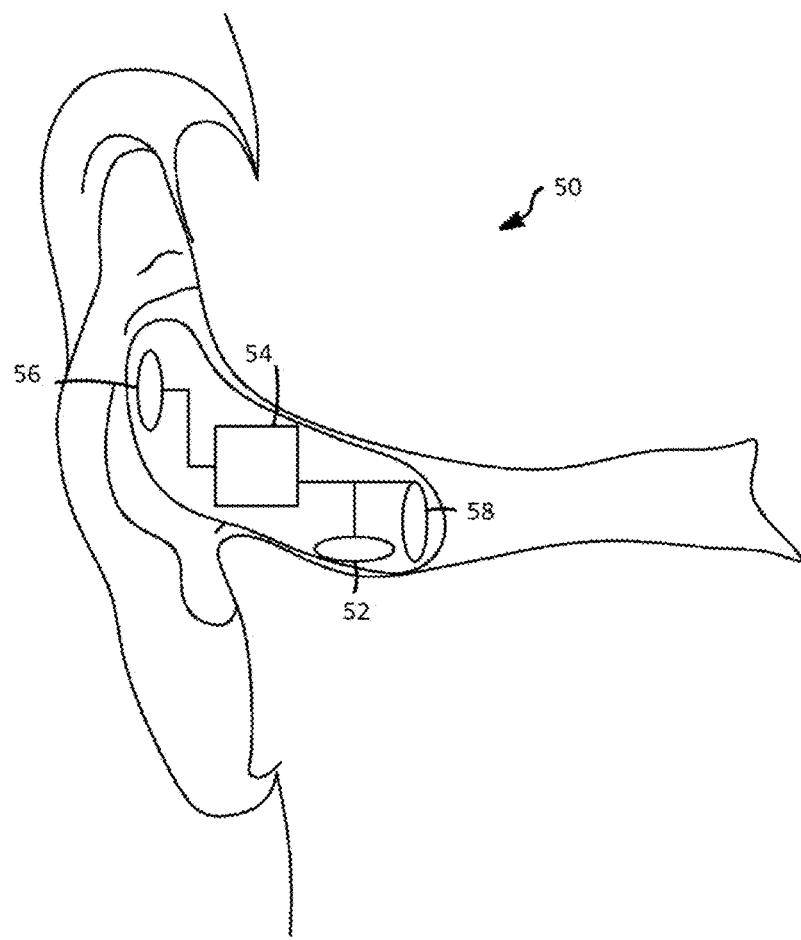
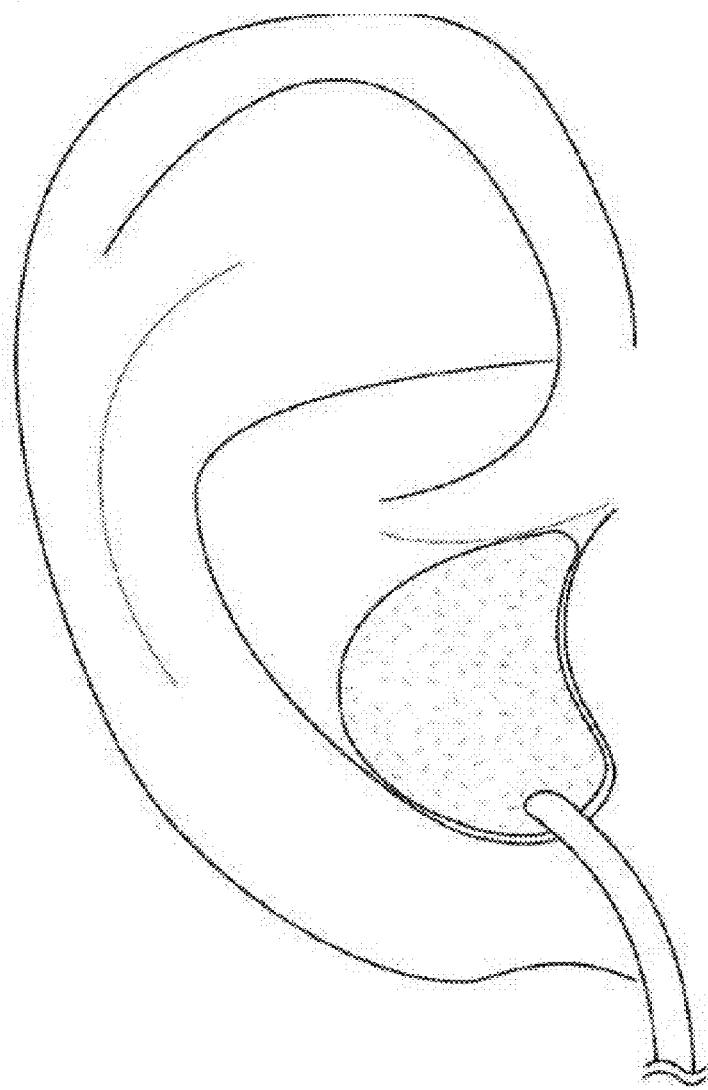
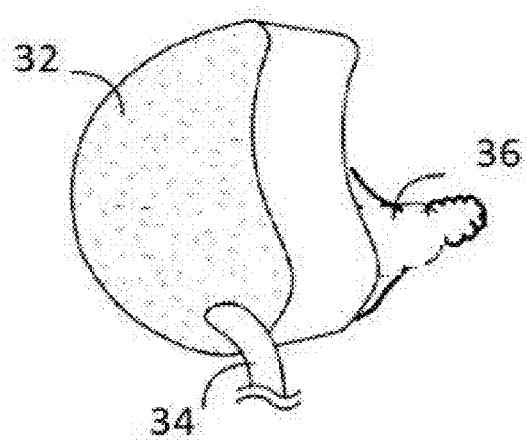


FIG. 4A



30

FIG. 4B



32

36

34

FIG. 5A

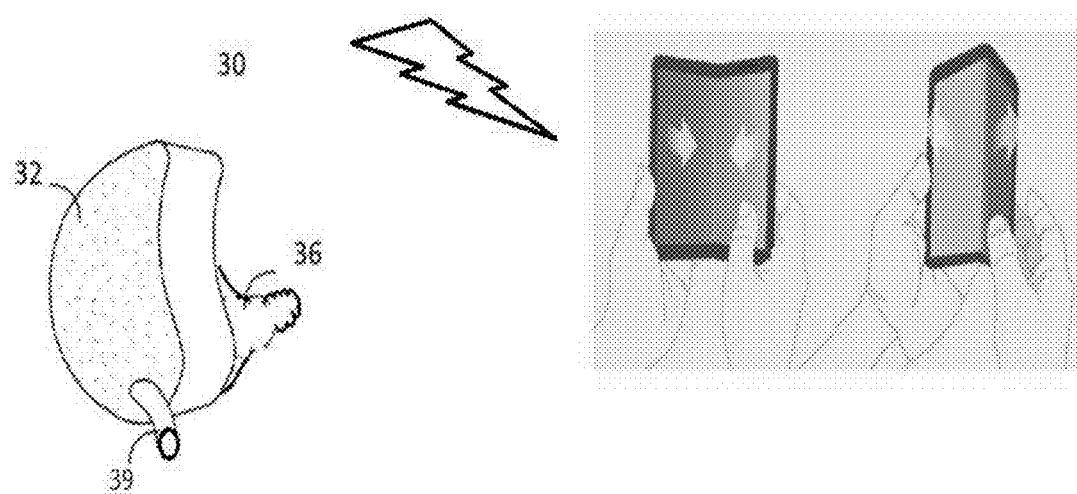
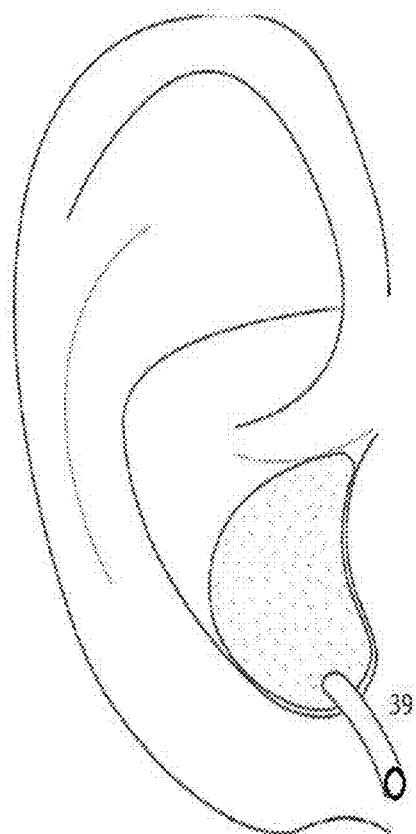
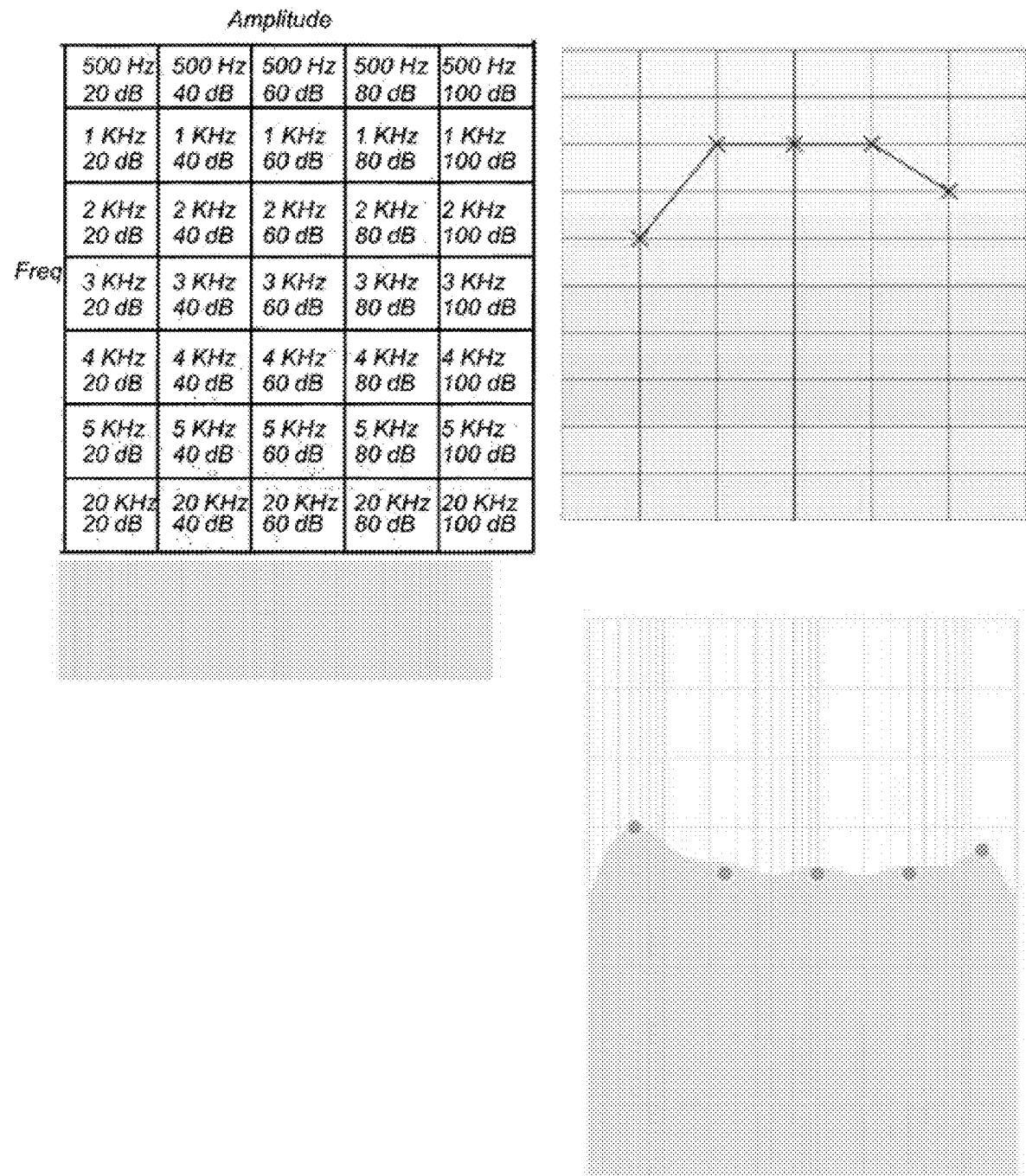


FIG. 5B

FIG. 6



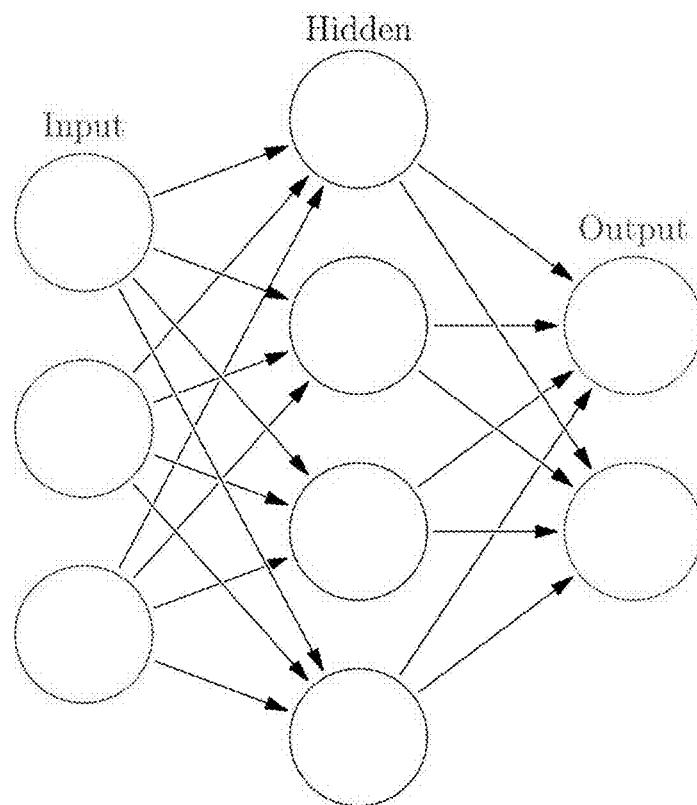


FIG. 7A

FIG. 7B

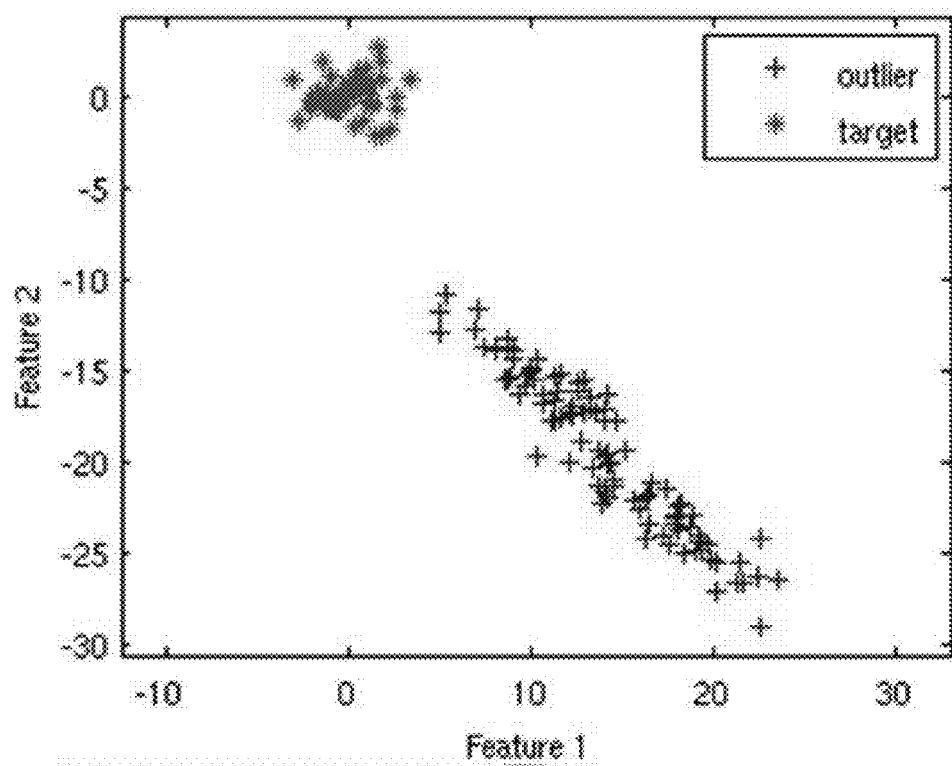


FIG. 7C

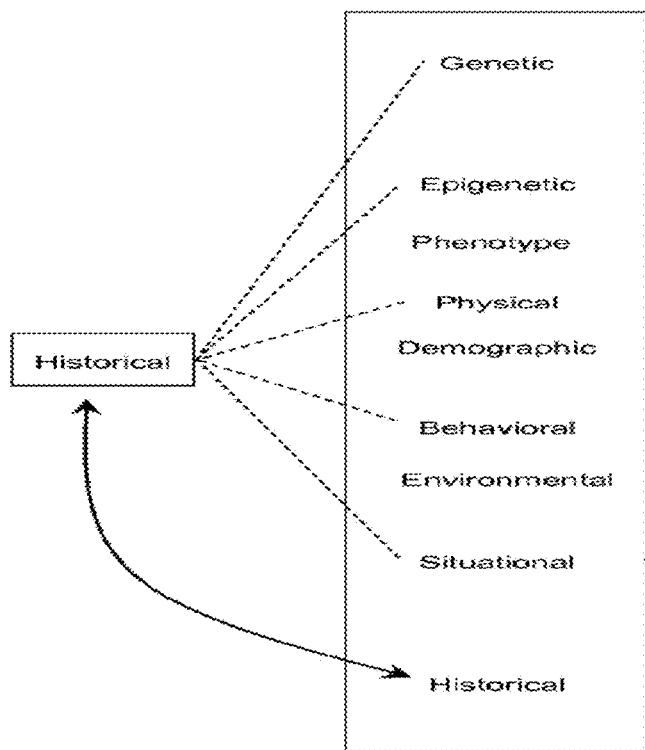
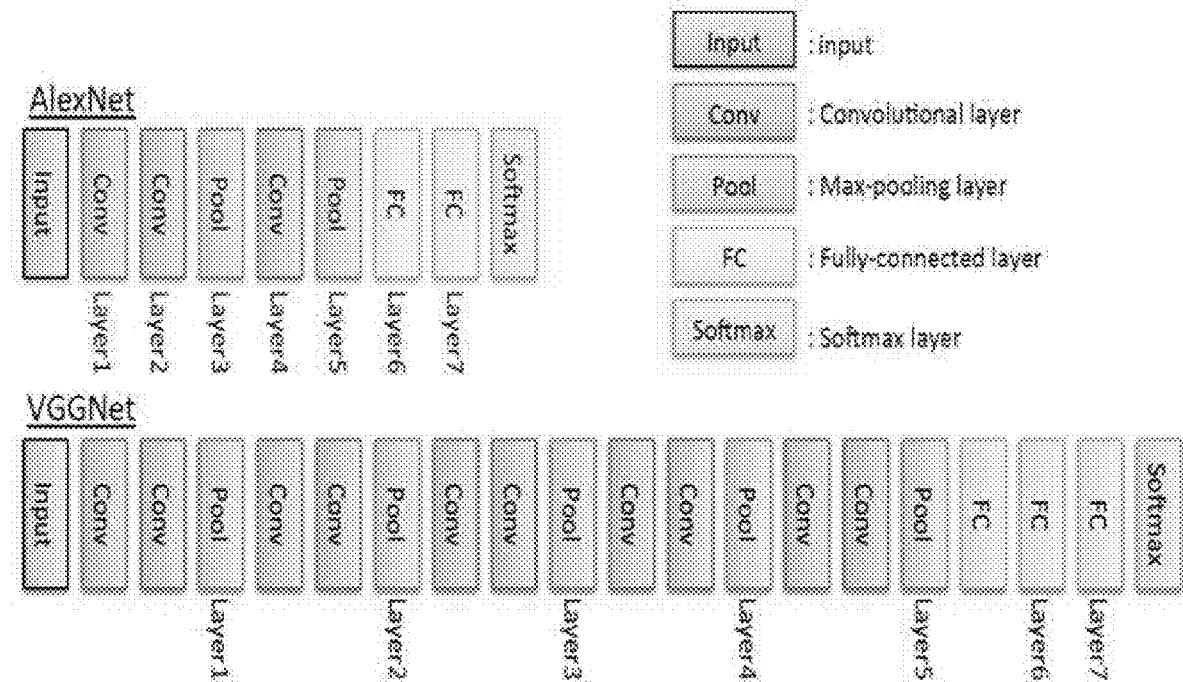


FIG. 7D

AI consultant is trained on all known literature and expert methodology (499). Treating physician/doctor logs into a consultation system and “Create New Case” (500). Initiates “New Case Wizard” with guided “Initial Assessment” form (501).

- enter Patient Information (502)
- enter Initial Assessment of patient/case (504)
- upload Test Results, Patient Photographs and X-Rays (506)
- accept Payment and Service Terms and Conditions (508)
- review Summary of Case (510)
- submit Forms to a consultant such as an Oncology Service Provider (512).

After the case has been submitted, the Service Provider Consultant can log into the system 1 and consult/process the case (520).

Using the Treating Doctors Initial Assessment and Photos/X-Rays, the Consultant will click on “Case Consultation” to initiate the “Case Consultation Wizard” (522).

- AI Consultant can fill out the “Consultant Record Analysis” form (524).
- AI Consultant can also complete the “Prescription Form” (526)
- Submit completed forms to the original Treating Doctor (528).

Treating Doctor accesses completed forms and either accept the consultation results (i.e. a pre-filled Prescription form) or communicate with the Consultant (530).

Treating Doctor can log into the system (532)

Click on Patient Name to review (534)

Review the Consultation Results (Summary Letter and pre-filled Prescription Form) (536).

If satisfied, the Treating Doctor can click “Approve Treatment” (538)

Mark the case as having being approved (540).

Treating Doctor can print a copy of the Prescription Form and the Summary Letter for submission to hearing aid provider (542).

If not satisfied, Treating Doctor can initiate a computer dialog with the Consultant by clicking “Send a Message” (544).

Treating Doctor will be presented with the “Send a Message” screen where a message about the case under consultation can be written (546).

Treating Doctor would click “Submit” to send the message to the appropriate Consultant (548).

Consultant can reply to the Treating Doctor's Message (550)

FIG. 7E

FIG. 8A

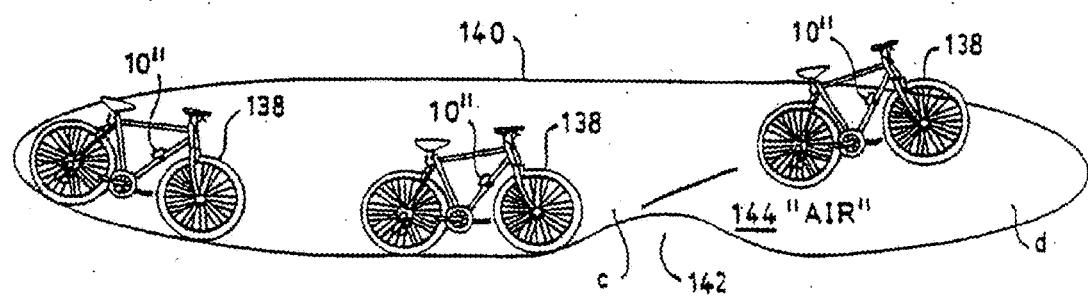
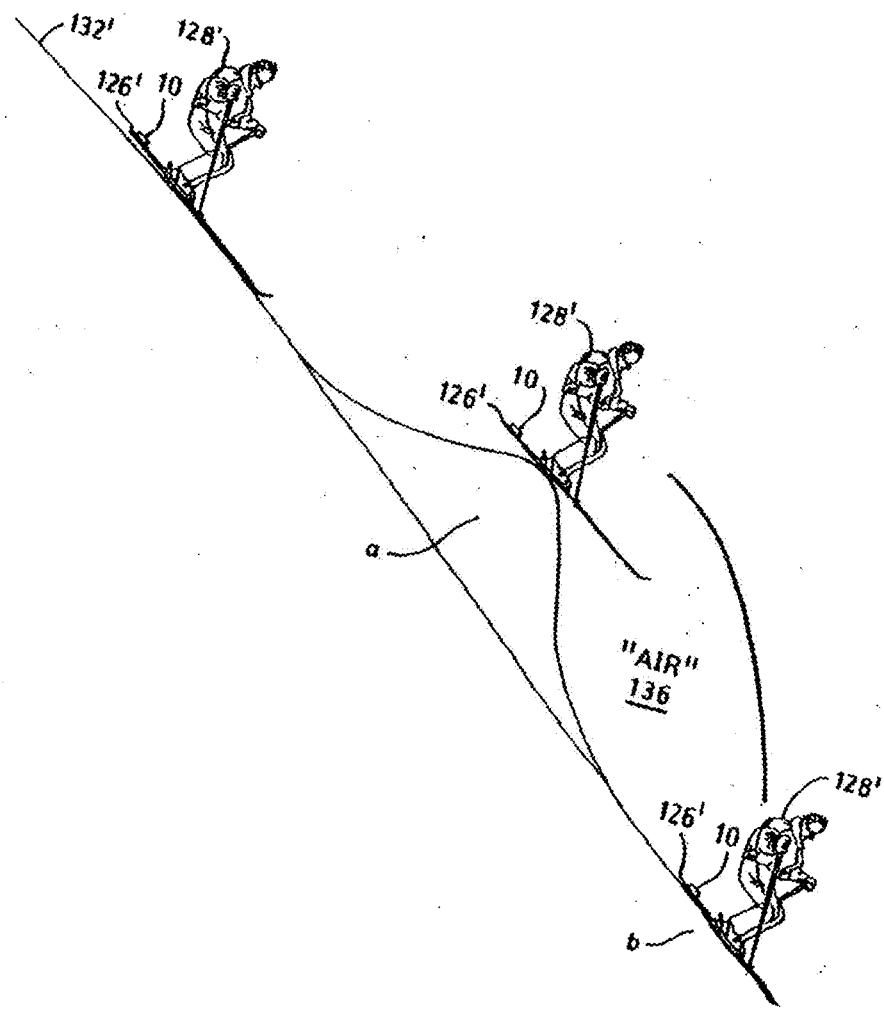


FIG. 8B

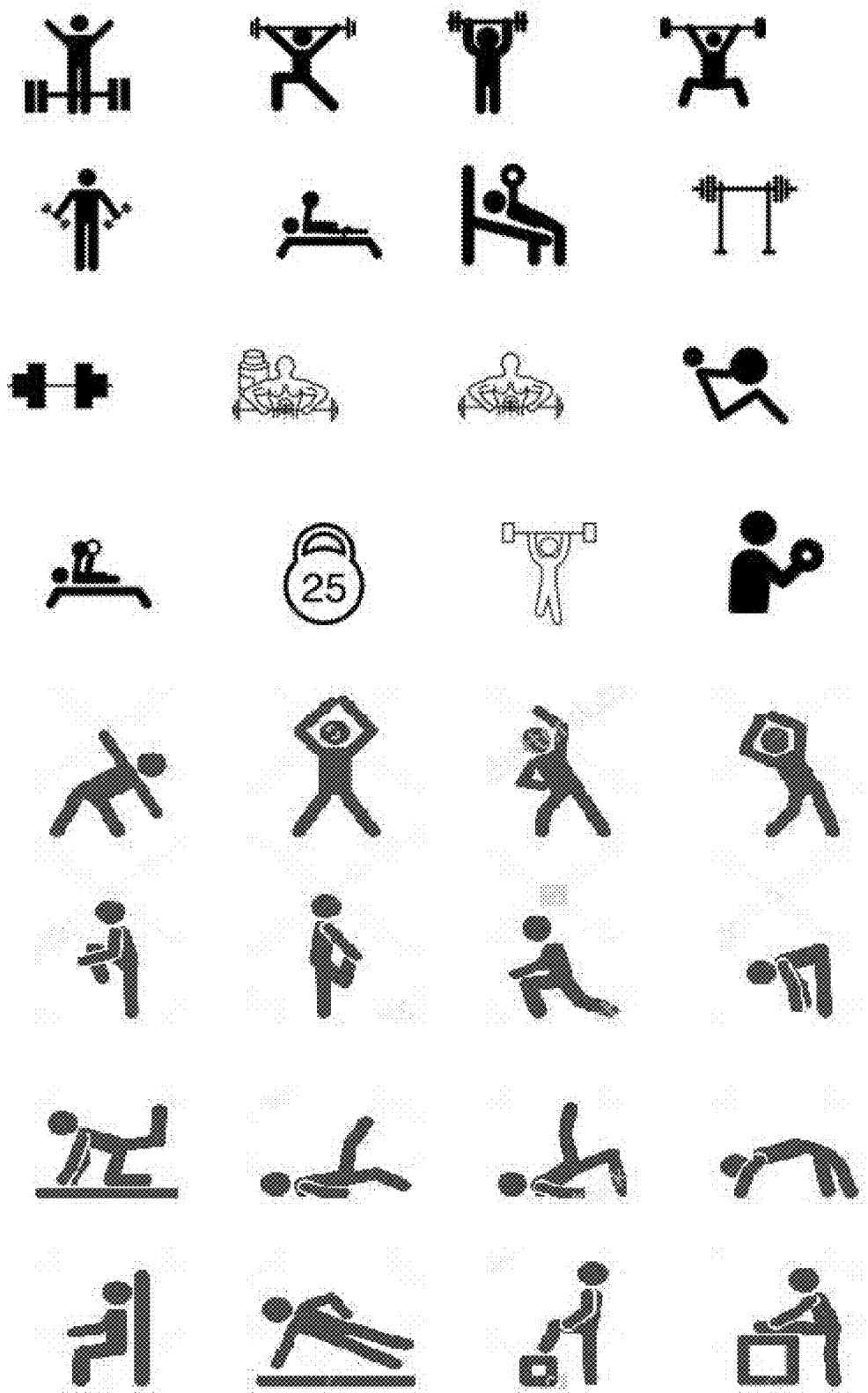


FIG. 8C

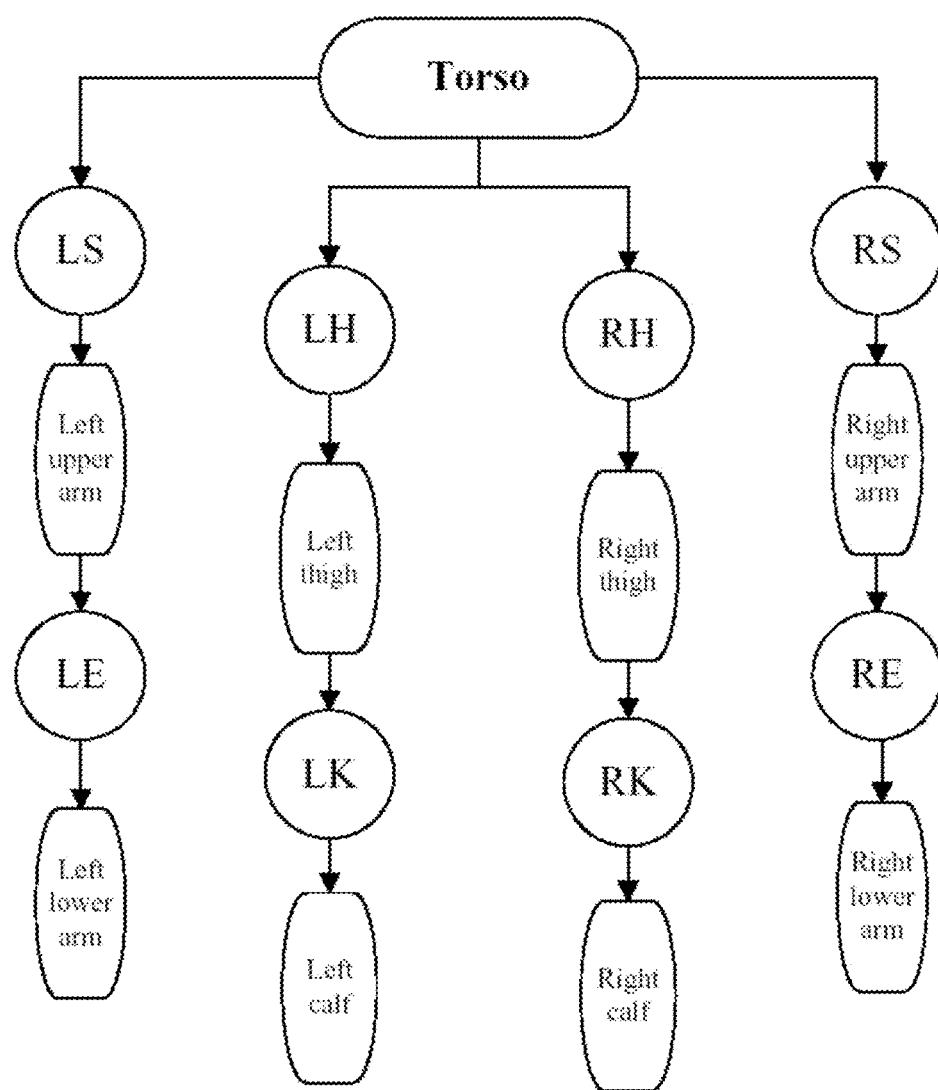


FIG. 8D

FIG. 9A

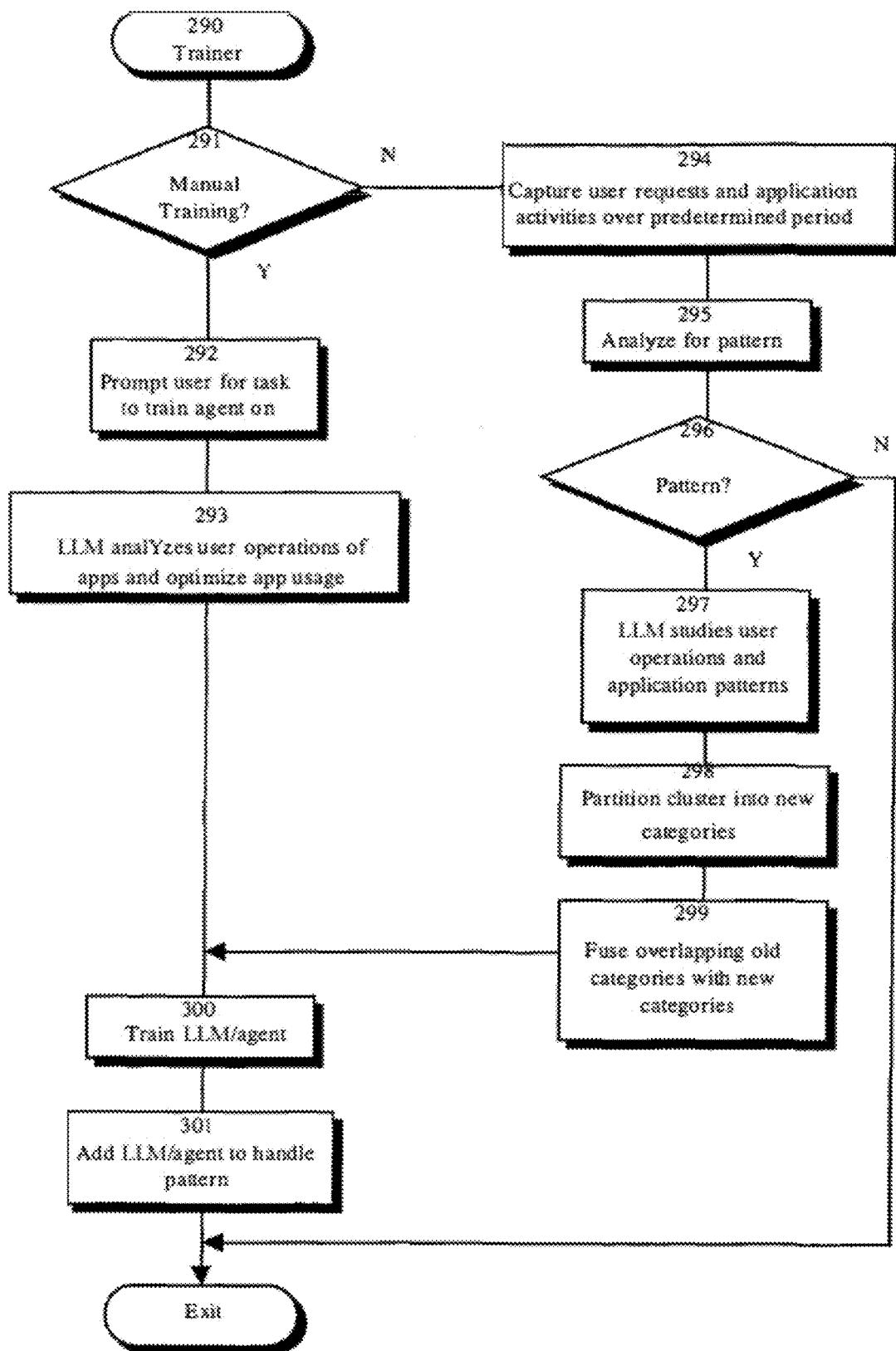
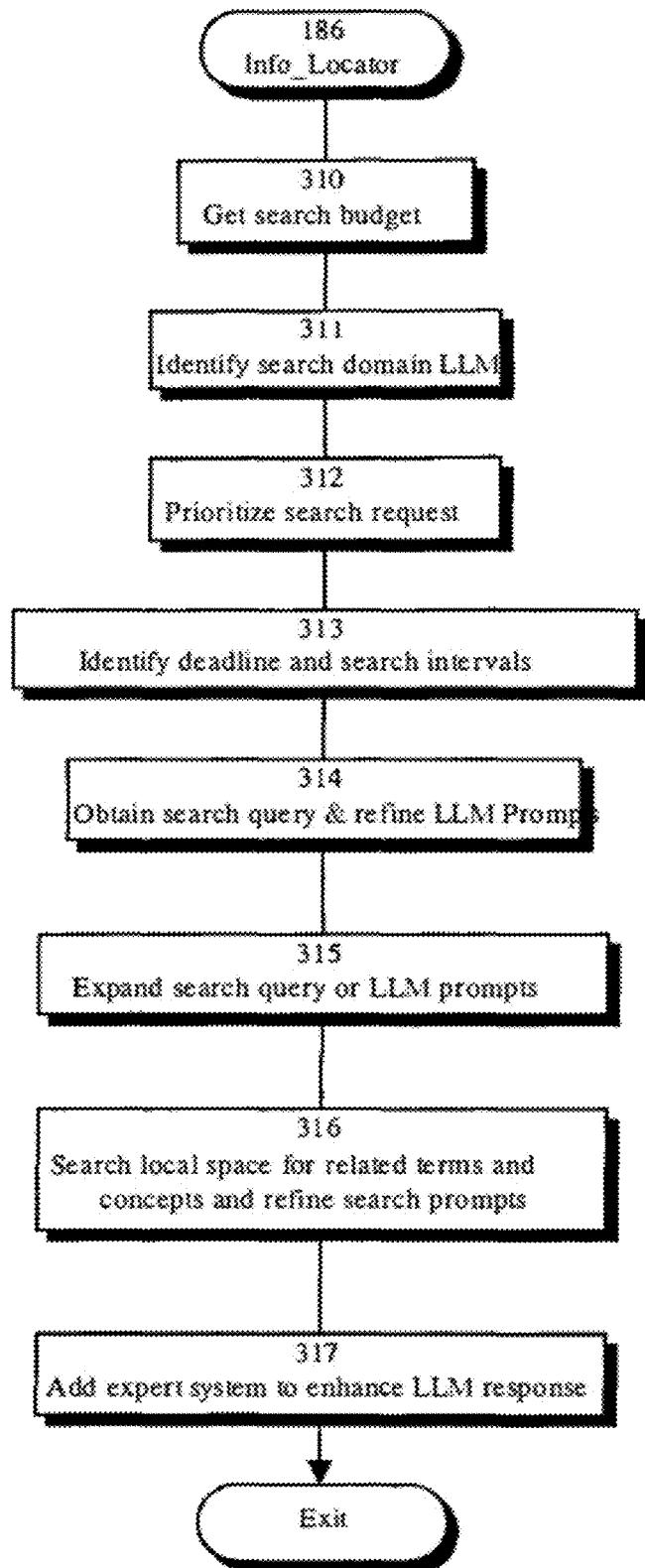


FIG. 9B



AI ASSISTANCE SYSTEM

BACKGROUND

[0001] The present system relates to ear system that applies AI to improve the sound quality, and also applies to provide a natural speech user interface, and in one embodiment acts as a gateway to a plurality of apps (super-apps).

SUMMARY

[0002] In one aspect, systems and methods for assisting a user include a housing custom fitted to a user anatomy; a microphone to capture sound coupled to a processor to deliver enhanced sound to the user anatomy; an amplifier with gain and amplitude controls for each hearing frequency; and a learning machine (such as a neural network) to identify an aural environment (such as party, movie, office, or home environment) and adjusting amplifier controls to optimize hearing based on the identified aural environment. In one embodiment, the environment can be identified by the background noise or inferred through GPS location, for example.

[0003] In another aspect, a method for assisting a user includes receiving a verbal request from the user with a device coupled to an ear, wherein the device includes biometric sensors coupled to the ear; selecting a learning machine language model or an intelligent agent based on the user request, the selected model or agent applying user environmental information including positioning information or nearby images in responding to the requested information or decision; and providing a response to the verbal request. In implementations of this aspect, the position/image data can be used to customize the response. For example, if the image data of people nearby is captured, the system can do face recognition and look up profiles (via LinkedIn or other social networks) and provide real time assistance with the bios. The biometric data can be used to custom the response as well. For example, if in meeting a person with a suitable bio and the heart rate goes up, the model/agent can assist with helpful text to guide the user in navigating through the discussion with relevant data. In a further example, based on location, the user can be assisted with suitable restaurants/cafes nearby for further discussion, and the system can also call Lyft/Uber for transportation at the end of the conversation, for example. Multiple apps can be controlled by the model/agent to offload as many tasks as possible.

[0004] In another aspect, a method for assisting a user includes customizing an in-ear device to a user anatomy; capturing sound using the in-ear device; enhancing sound based on predetermined profiles and transmitting the sound to an ear drum.

[0005] In yet another aspect, a method for assisting a user includes customizing an in-ear device to a user anatomy; capturing sound using the in-ear device; capturing vital signs with the in-ear device; and learning health signals from the sound and the vital signs from the in-ear device.

[0006] In a further aspect, a method includes customizing an in-ear device to a user anatomy; capturing vital signs with the in-ear device; and learning health signals from the vital signs from the in-ear device.

[0007] In another aspect, a method includes customizing an in-ear device to a user anatomy; capturing vital signs to detect biomarkers with the in-ear device; correlating genomic disease markers with the detected biomarkers to predict health with the in-ear device.

genomic disease markers with the detected biomarkers to predict health with the in-ear device.

[0008] In another aspect, a method includes customizing an in-ear device to a user anatomy; identifying genomic disease markers; capturing vital signs to detect biomarkers with the in-ear device; correlating genomic disease markers with the detected biomarkers to predict health with the in-ear device.

[0009] In another aspect, a method includes customizing an in-ear device to a user anatomy; capturing accelerometer data and vital signs; controlling a virtual reality device or augmented reality device with acceleration or vital sign data from the in-ear device.

[0010] In another aspect, a method includes customizing an in-ear device to a user anatomy; capturing heart rate, EEG or ECG signal with the in-ear device; and determining user intent with the in-ear device. The determined user intent can be used to control an appliance, or can be used to indicate interest for advertisers.

[0011] In another aspect, a method includes customizing an in-ear device to a user anatomy; capturing heart rate, EEG/ECG signal or temperature data to detect biomarkers with the in-ear device; and predict health with the in-ear device data.

[0012] In another aspect, a method includes customizing an in-ear device to a user anatomy; capturing sounds from an advertisement, capturing vital signs associated with the advertisement; and customizing the advertisement to attract the user.

[0013] In another aspect, a method includes customizing an in-ear device to a user anatomy; capturing vital signs associated with a situation; detecting user emotion from the vital signs; and customizing an action based on user emotion. In one embodiment, such detected user emotion is provided to a robot to be more responsive to the user.

[0014] In another aspect, a method includes customizing an in-ear device to a user anatomy; capturing a command from a user, detecting user emotion based on vital signs; and performing an action in response to the command and the detected user emotion.

[0015] In another aspect, a method includes customizing an in-ear device to a user anatomy; capturing a command from a user, authenticating the user based on a voiceprint or user vital signs; and performing an action in response to the command.

[0016] In one aspect, a method for assisting a user includes customizing an in-ear device to a user anatomy; capturing sound using the in-ear device; enhancing sound based on predetermined profiles and transmitting the sound to an ear drum.

[0017] In one aspect, a method for assisting a user includes providing an in-ear device to a user anatomy; capturing sound using the in-ear device; capturing vital signs with the in-ear device; and learning health signals from the sound and the vital signs from the in-ear device.

[0018] In another aspect, a method includes providing an in-ear device to a user anatomy; capturing vital signs with the in-ear device; and learning health signals from the vital signs from the in-ear device.

[0019] In another aspect, a method includes providing an in-ear device to a user anatomy; capturing vital signs to detect biomarkers with the in-ear device; correlating genomic disease markers with the detected biomarkers to predict health with the in-ear device.

[0020] In another aspect, a method includes providing an in-ear device to a user anatomy; identifying genomic disease markers; capturing vital signs to detect biomarkers with the in-ear device; correlating genomic disease markers with the detected biomarkers to predict health with the in-ear device.

[0021] In another aspect, a method includes providing an in-ear device to a user anatomy; capturing accelerometer data and vital signs; controlling a virtual reality device or augmented reality device with acceleration or vital sign data from the in-ear device.

[0022] In another aspect, a method includes providing an in-ear device to a user anatomy; capturing heart rate, EEG or ECG signal with the in-ear device; and determining user intent with the in-ear device. The determined user intent can be used to control an appliance, or can be used to indicate interest for advertisers.

[0023] In another aspect, a method includes providing an in-ear device to a user anatomy; capturing heart rate, EEG/ECG signal or temperature data to detect biomarkers with the in-ear device; and predict health with the in-ear device data.

[0024] In another aspect, a method includes providing an in-ear device to a user anatomy; capturing sounds from an advertisement, capturing vital signs associated with the advertisement; and customizing the advertisement to attract the user.

[0025] In another aspect, a method includes providing an in-ear device to a user anatomy; capturing vital signs associated with a situation; detecting user emotion from the vital signs; and customizing an action based on user emotion. In one embodiment, such detected user emotion is provided to a robot to be more responsive to the user.

[0026] In another aspect, a method includes providing an in-ear device to a user anatomy; capturing a command from a user, detecting user emotion based on vital signs; and performing an action in response to the command and the detected user emotion.

[0027] In another aspect, a method includes providing an in-ear device to a user anatomy; capturing a command from a user, authenticating the user based on a voiceprint or user vital signs; and performing an action in response to the command.

[0028] In another aspect, a method includes providing an in-ear device to a user anatomy; determine an audio response chart for a user based on a plurality of environments (restaurant, office, home, theater, party, concert, among others), determining a current environment, and updating the hearing aid parameters to optimize the amplifier response to the specific environment. The environment can be auto detected based on GPS position data or external data such as calendaring data or can be user selected using voice command, for example. In another embodiment, a learning machine automatically selects an optimal set of hearing aid parameters based on ambient sound and other confirmatory data.

[0029] Implementations of any of the above aspects may include one or more of the following:

- [0030] detecting electrical potentials encephalography (EEG) or electrocardiogram (ECG) in the ear;
- [0031] using a camera in the ear to detect ear health;
- [0032] detecting blood flow with an in-ear sensor;
- [0033] detecting with an in-ear sensor blood parameters including carboxyhemoglobin (HbCO), methemoglobin (HbMet) and total hemoglobin (Hbt);

[0034] detecting pressure based on a curvature of an ear drum;

[0035] detecting body temperature in the ear;

[0036] detecting one or more of: alpha rhythm, auditory steady-state response (ASSR), steady-state visual evoked potentials (SSVEP), visually evoked potential (VEP), visually evoked response (VER) and visually evoked cortical potential (VECP), cardiac activity, speech and breathing;

[0037] detecting alpha rhythm, auditory steady-state response (ASSR), steady-state visual evoked potentials (SSVEP), and visually evoked potential (VEP);

[0038] correlating EEG, ECG, speech and breathing to determine health;

[0039] correlating cardiac activity, speech and breathing;

[0040] determining user health by detecting fluid in an ear structure, change in ear color, curvature of the ear structure;

[0041] determining one or more bio-markers from the vital signs and indicating user health;

[0042] performing a 3D scan inside an ear canal;

[0043] matching predetermined points on the 3D scan to key points on a template and morphing the key points on the template to the predetermined points;

[0044] 3D printing a model from the 3D scan and fabricating the in-ear device;

[0045] correlating genomic biomarkers for diseases to the vital signs from the in-ear device and applying a learning machine to use the vital signs from the in-ear device to predict disease conditions;

[0046] determining a fall based on accelerometer data, vital signs and sound;

[0047] determining loneliness or mental condition based on activity of life data; or

[0048] providing a user dashboard showing health data over a period of time and matching research data on the health signals.

[0049] Advantages of the preferred embodiments may include one or more of the following. By using the disclosed data analysis method, a health system can add a new method of collecting and analyzing patient information. Patients can be assured that there is quantitative, objective information. For most medical conditions, biomarker pattern can be obtained and compared against known conditions. The system supplies adjunctive information by which the health practitioner may make a better decision regarding treatment alternatives. Appropriate measurement yields tremendous information to the doctors and users. This information improves diagnosis, aids in better treatment, and allows for earlier detection of problems.

BRIEF DESCRIPTION OF THE DRAWINGS

[0050] Other systems, methods, features, and advantages of the present system will be or will become apparent to one of ordinary skill in the art upon examination of the following figures and detailed description. It is intended that all such additional systems, methods, features, and advantages be included within this description, be within the scope of the present invention, and be protected by the accompanying claims. Component parts shown in the drawings are not necessarily to scale, and may be exaggerated to better illustrate the important features of the present invention. In

the drawings, like reference numerals designate like parts throughout the different views, wherein:

[0051] FIG. 1 shows an exemplary process to render sound and to monitor a user.

[0052] FIGS. 2A-2B show a custom monitoring device.

[0053] FIG. 3A shows an exemplary scanner to scan ear structures.

[0054] FIG. 3B shows a smart-phone scanner.

[0055] FIG. 3C shows exemplary point clouds from the camera(s).

[0056] FIG. 3D shows another custom hearing aid and monitoring device.

[0057] FIGS. 4A-4B show another custom hearing aid and monitoring device.

[0058] FIGS. 5A-5B show another custom hearing aid and monitoring device with extended camera, microphone and antenna.

[0059] FIG. 6 shows exemplary audiogram test flow and user interface designs.

[0060] FIG. 7A shows a neural network, FIG. 7B shows an exemplary analysis with normal targets and outliers as warning labels, FIG. 7C show an exemplary learning system, FIG. 7D shows a medical AI system with the learning machines, and FIG. 7E show an exemplary AI based expert consultation system with human healthcare personnel.

[0061] FIG. 8A-8C shows exemplary coaching system for skiing, bicycling, and weightlifting/free style exercise, respectively, while FIG. 8D shows a kinematic modeling for detecting exercise motion which in turn allows precision coaching suggestions, all using data from ear sensors.

[0062] FIGS. 9A-9B show exemplary agents used to assist the user.

[0063] Similar reference characters denote corresponding features consistently throughout the attached drawings.

DETAILED DESCRIPTION OF THE INVENTION

[0064] FIG. 1 shows an exemplary process to form and use a custom ear insert as a hearing aid with vital sign monitoring. The process is as follows: Scan ear (1) Generate 3D model of ear structure (2) 3D Print a Negative of the ear structure with provisioned space for electronic module (3) Form a flexible ear insert from the ear structure negative (4) Insert or add amplifier to the provisioned space to the ear insert (5) During use, snugly push the ear insert toward the ear canal (6) Render sound (7) Capture medical data as biomarkers including ECG, EGG, physical condition, heart rate, blood flow data, temperature (8), and apply a language model assistance to assist the user (9)

[0065] FIGS. 2A-2B show an exemplary ear insert that is injection molded with a pliant material. In one embodiment, the material is a medical grade thermoplastic elastomer. The device can render sound for the user, and can also collect medical data such as oxygenation or other blood constituents or temperature, for example. Turning now to FIG. 2A, an ear monitoring device **10** includes a customized shell **12** for reception in the ear canal of a person. The shell **12** has an opening **13** to receive the hearing aid **10** when mounted. In general, the shell **12** has the frontally facing part pointing outward from the person's ear canal when the shell is positioned therein and the frontally facing part has an opening **13** to receive an electronic module therein. The part of the shell **12** actually engaging the user's ear canal is denoted **11**.

[0066] FIG. 3A shows an exemplary ear scanner **30** with a camera mounted on a tip that is inserted into the ear canal that terminates at an ear drum **32**. The camera captures images that are converted into point cloud, as shown in FIG. 3B. In one embodiment, as the user or operator passes the camera of the scanner in the ear, a point cloud is generated and a 3D model of the ear is constructed. The 3D model is used to construct the device of FIG. 2, and further used in the sound modeling process to optimize electronic performance for hearing. The scanner **20** has camera(s) that pick up a plurality of images which are then used to reconstruct a 3D model of the ear structure. The tip of the scanner **20** is made sufficiently large that it cannot accidentally be inserted too far into the ear canal as to puncture the ear drum or otherwise harm the ear. The optics of the scanner **20** has a lamp and a lens and the camera view angle provides at least one eccentric observation point and/or at least one light source (preferably both) into an ear canal of a subject's outer ear and capturing imaged from the eccentric position. In particular, an optical axis does not have to correspond to the longitudinal axis of the head portion. Instead, an optical axis of the electronic imaging unit may be arranged radially offset.

[0067] FIG. 3B shows a smart phone scanner embodiment. In this embodiment, scanned ear structures can be detected on any ARKit. When first run, the app displays a box that roughly estimates the size of whatever real-world objects appear centered in the camera view. Before scanning, the system determines a region of the world contains the ear structures. The ear is scanned, and the camera is moved to scan from different angles. The app highlights parts of the bounding box to indicate sufficient scanning to recognize the object from the corresponding direction.

[0068] One aspect of an ear sensor optically measures physiological parameters related to blood constituents by transmitting multiple wavelengths of light into a concha site and receiving the light after attenuation by pulsatile blood flow within the concha site. The ear sensor comprises a sensor body, a sensor connector and a sensor cable interconnecting the sensor body and the sensor connector. The sensor body comprises a base, legs and an optical assembly. The legs extend from the base to detector and emitter housings. An optical assembly has an emitter and a detector. The emitter is disposed in the emitter housing and the detector is disposed in the detector housing. The legs have an unflexed position with the emitter housing proximate the detector housing and a flexed position with the emitter housing distal the detector housing. The legs are moved to the flexed position so as to position the detector housing and emitter housing over opposite sides of a concha site. The legs are released to the unflexed position so that the concha site is grasped between the detector housing and emitter housing.

[0069] Pulse oximetry systems for measuring constituents of circulating blood can be used in many monitoring scenarios. The sensor transmits the detector signal to the monitor, which processes the signal to provide a numerical readout of physiological parameters such as oxygen saturation (SpO_2) and pulse rate. Advanced physiological monitoring systems may incorporate pulse oximetry in addition to advanced features for the calculation and display of other blood parameters, such as carboxyhemoglobin (HbCO), methemoglobin (HbMet) and total hemoglobin (Hbt), as a few examples. In other embodiments, the device has physi-

ological monitors and corresponding multiple wavelength optical sensors capable of measuring parameters in addition to SpO₂, such as HbCO, HbMet and Hbt are described in at least U.S. patent application Ser. No. 12/056,179, filed Mar. 26, 2008, titled Multiple Wavelength Optical Sensor and U.S. patent application Ser. No. 11/366,208, filed Mar. 1, 2006, titled Noninvasive Multi-Parameter Patient Monitor, both incorporated by reference herein. Further, noninvasive blood parameter monitors and corresponding multiple wavelength optical sensors to sense SpO₂, pulse rate, perfusion index (PI), signal quality (SiQ), pulse variability index (PVI), HbCO and HbMet among other parameters.

[0070] Heart pulse can be detected by measuring the dilation and constriction of tiny blood vessels in the ear canal. In one embodiment, the dilation measurement is done optically and in another embodiment, a micromechanical MEMS sensor is used. ECG sensor can be used where the electrode can detect a full and clinically valid electrocardiogram, which records the electrical activity of the heart.

[0071] Impact sensors, or accelerometers, measure in real time the force and even the number of impacts that players sustain. Data collected is sent wirelessly via Bluetooth to a dedicated monitor on the sidelines, while the impact prompts a visual light or audio alert to signal players, coaches, officials, and the training or medical staff of the team. One such sensor example is the ADXL377 from Analog Devices, a small, thin and low-power 3-axis accelerometer that measures acceleration from motion, shock, or vibration. It features a full-scale range of +200 g, which would encompass the full range of impact acceleration in sports, which typically does not exceed 150 g's. When a post-impact individual is removed from a game and not allowed to return until cleared by a concussion-savvy healthcare professional, most will recover quickly. If the injury is undetected, however, and an athlete continues playing, concussion recovery often takes much longer. Thus, the system avoids problems from delayed or unidentified injury can include: Early dementia, Depression, Rapid brain aging, and Death. The cumulative effects of repetitive head impacts (RHI) increases the risk of long-term neuro-degenerative diseases, such as Parkinson's disease, Alzheimer's, Mild Cognitive Impairment, and ALS or Lou Gehrig's disease. The sensors can warn of dangerous concussions.

[0072] The device can use optical sensors for heart rate (HR) as a biomarker in heart failure (HF) both of diagnostic and prognostic values. HR is a determinant of myocardial oxygen demand, coronary blood flow, and myocardial performance and is central to the adaptation of cardiac output to metabolic needs. Increased HR can predict adverse outcome in the general population and in patients with chronic HF. Part of the ability of HR to predict risk is related to the forces driving it, namely, neurohormonal activation. HR relates to emotional arousal and reflects both sympathetic and parasympathetic nervous system activity. When measured at rest, HR relates to autonomic activity during a relaxing condition. HR reactivity is expressed as a change from resting or baseline that results after exposure to stimuli. These stress-regulating mechanisms prepare the body for fight or flight responses, and as such can explain individual differences to psychopathology. Thus, the device monitors HR as a biomarker of both diagnostic and prognostic values.

[0073] The HR output can be used to analyze heart-rate variability (HRV) (the time differences between one beat and the next) and HRV can be used to indicate the potential

health benefits of food items. Reduced HRV is associated with the development of numerous conditions for example, diabetes, cardiovascular disease, inflammation, obesity and psychiatric disorders. Aspects of diet that are viewed as undesirable, for example high intakes of saturated or trans-fat and high glycaemic carbohydrates, have been found to reduce HRV. The consistent relationship between HRV, health and morbidity allows the system to use HRV as a biomarker when considering the influence of diet on mental and physical health. Further HRV can be used as a biomarker for aging. HRV can also act as biomarkers for:

[0074] Overtraining: "Cumulative or too intensive sporting activity (e.g. competition series, overtraining syndrome), however, brings about a decrease in HRV"

[0075] Physical Fitness: "People who have an active lifestyle and maintain a good or high level of physical fitness or above-average sporting activity can achieve an increase in their basic parasympathetic activity and thus an increase in their HRV."

[0076] Overweight: "an elevated body weight or elevated free-fat mass⁵⁷ correlates with a decrease in HRV. Both active and passive smoking lead to an increase in HRV"

[0077] Alcohol Abuse: "Regular chronic alcohol abuse above the alcohol quantity of a standard drink for women or two standard drinks for men reduces HRV, while moderate alcohol consumption up to these quantities does not change the HRV and is not associated with an increase"

[0078] Smoking: "Both active and passive smoking lead to an increase in HRV"

[0079] Sleep: Another important factor that affects your HRV score is the amount and quality of sleep.

[0080] In one embodiment, the system determines a dynamical marker of sino-atrial instability, termed heart rate fragmentation (HRF) and is used a dynamical biomarker of adverse cardiovascular events (CVEs). In healthy adults at rest and during sleep, the highest frequency at which the sino-atrial node (SAN) rate fluctuates varies between ~0.15 and 0.40 Hz. These oscillations, referred to as respiratory sinus arrhythmia, are due to vagally-mediated coupling between the SAN and breathing. However, not all fluctuations in heart rate (HR) at or above the respiratory frequency are attributable to vagal tone modulation. Under pathologic conditions, an increased density of reversals in HR acceleration sign, not consistent with short-term parasympathetic control, can be observed.

[0081] The system captures ECG data as biomarkers for cardiac diseases such as myocardial infarction, cardiomyopathy, atrioventricular bundle branch block, and rhythm disorders. The ECG data is cleaned up, and the system extracts features by taking quantiles of the distributions of measures on ECGs, while commonly used characterizing feature is the mean. The system applies commonly used measurement variables on ECGs without preselection and use dimension reduction methods to identify biomarkers, which is useful when the number of input variables is large and no prior information is available on which ones are more important. Three frequently used classifiers are used on all features and to dimension-reduced features by PCA. The three methods are from classical to modern: stepwise discriminant analysis (SDA), SVM, and LASSO logistic regression.

[0082] In one embodiment, four types of features are considered as input variables for classification: T wave type, time span measurements, amplitude measurements, and the slopes of waveforms for features such as

[0083] (1) T Wave Type. The ECGPUWAVE function labels 6 types of T waves for each beat: Normal, Inverted, Positive Monophasic, Negative Monophasic, Biphasic Negative-Positive, and Biphasic Positive-Negative based on the T wave morphology. This is the only categorical variable considered.

[0084] (2) Time Span Measurements. Six commonly used time span measurements are considered: the length of the RR interval, PR interval, QT interval, P wave, QRS wave, and T wave.

[0085] (3) Amplitude Measurements. The amplitudes of P wave, R-peak, and T wave are used as input variables. To measure the P wave amplitude, we first estimate the baseline by taking the mean of the values in the PR segment, ST segment, and TP segment (from the end of the T wave to the start of the P wave of the next heartbeat), then subtract the maximum and minimum values of the P wave by the estimated baseline, and take the one with a bigger absolute value as the amplitude of P wave. Other amplitude measurements are obtained similarly.

[0086] (4) The Slopes of Waveforms. The slopes of waveforms are also considered to measure the dynamic features of a heartbeat. Each heartbeat is split into nine segments and the slope of the waveform in each segment is estimated by simple linear regression.

[0087] The device can include EEG sensors which measure a variety of EEG responses—alpha rhythm, ASSR, SSVEP and VEP—as well as multiple mechanical signals associated with cardiac activity, speech and breathing. EEG sensors can be used where electrodes provide low contact impedance with the skin over a prolonged period of time. A low impedance stretchable fabric is used as electrodes. The system captures various EEG paradigms: ASSR, steady-state visual evoked potential (SSVEP), transient response to visual stimulus (VEP), and alpha rhythm. The EEG sensors can predict and assess the fatigue based on the neural activity in the alpha band which is usually associated with the state of wakeful relaxation and manifests itself in the EEG oscillations in the 8-12 Hz frequency range, centered around 10 Hz. The loss of alpha rhythm is also one of the key features used by clinicians to define the onset of sleep. A mechanical transducer (electret condenser microphone) within its multimodal electro-mechanical sensor, which can be used as a reference for single-channel digital denoising of physiological signals such as jaw clenching and for removing real-world motion artifacts from ear-EEG. In one embodiment, a microphone at the tip of the earpiece facing towards the eardrum can directly capture acoustic energy traveling from the vocal chords via auditory tube to the ear canal. The output of such a microphone would be expected to provide better speech quality than the sealed microphone within the multimodal sensor.

[0088] The system can detect auditory steady-state response (ASSR) as a biomarker a type of ERP which can test the integrity of auditory pathways and the capacity of these pathways to generate synchronous activity at specific frequencies. ASSRs are elicited by temporally modulated auditory stimulation, such as a train of clicks with a fixed inter-click interval, or an amplitude modulated (AM) tone.

After the onset of the stimulus, the EEG or MEG rapidly entrains to the frequency and phase of the stimulus. The ASSR is generated by activity within the auditory pathway. The ASSR for modulation frequencies up to 50 Hz is generated from the auditory cortex based on EEG. Higher frequencies of modulation (>80 Hz) are thought to originate from brainstem areas. The type of stimulus may also affect the region of activation within the auditory cortex. Amplitude modulated (AM) tones and click train stimuli are commonly used stimuli to evoke the ASSR.

[0089] The EEG sensor can be used as a brain-computer interface (BCI) and provides a direct communication pathway between the brain and the external world by translating signals from brain activities into machine codes or commands to control different types of external devices, such as a computer cursor, cellphone, home equipment or a wheelchair. SSVEP can be used in BCI due to high information transfer rate (ITR), little training and high reliability. The use of in-ear EEG acquisition makes BCI convenient, and highly efficient artifact removal techniques can be used to derive clean EEG signals.

[0090] The system can measure visually evoked potential (VEP), visually evoked response (VER) or visually evoked cortical potential (VECP). They refer to electrical potentials, initiated by brief visual stimuli, which are recorded from the scalp overlying visual cortex. VEP waveforms are extracted from the electro-encephalogram (EEG) by signal averaging. VEPs are used primarily to measure the functional integrity of the visual pathways from retina via the optic nerves to the visual cortex of the brain. VEPs better quantify functional integrity of the optic pathways than scanning techniques such as magnetic resonance imaging (MRI). Any abnormality that affects the visual pathways or visual cortex in the brain can affect the VEP. Examples are cortical blindness due to meningitis or anoxia, optic neuritis as a consequence of demyelination, optic atrophy, stroke, and compression of the optic pathways by tumors, amblyopia, and neurofibromatosis. In general, myelin plaques common in multiple sclerosis slow the speed of VEP wave peaks. Compression of the optic pathways such as from hydrocephalus or a tumor also reduces amplitude of wave peaks.

[0091] A bioimpedance (BI) sensor can be used to determine a biomarker of total body fluid content. The BIA is a noninvasive method for evaluation of body composition, easy to perform, and fast, reproducible, and economical and indicates nutritional status of patients by estimating the amount of lean body mass, fat mass, body water, and cell mass. The method also allows the assessment of patient's prognosis through the PA, which has been applied in patients with various diseases, including chronic liver disease. The phase angle varies according to the population and can be used for prognosis.

[0092] In another embodiment, the BI sensor can estimate glucose level. This is done by measuring the bioimpedance at various frequencies, where high frequency Bi is related to fluid volume of the body and low frequency BI is used to estimate the volume of extracellular fluid in the tissues.

[0093] The step of determining the amount of glucose can include comparing the measured impedance with a predetermined relationship between impedance and blood glucose level. In a particular embodiment, the step of determining the blood glucose level of a subject includes ascertaining the sum of a fraction of the magnitude of the measured impedance and a fraction of the phase of the measured impedance.

The amount of blood glucose, in one embodiment, is determined according to the equation: Predicted glucose=(0.31) Magnitude+(0.24)Phase where the impedance is measured at 20 kHz. In certain embodiments, impedance is measured at a plurality of frequencies, and the method includes determining the ratio of one or more pairs of measurements and determining the amount of glucose in the body fluid includes comparing the determined ratio(s) with corresponding predetermined ratio(s), i.e., that have been previously correlated with directly measured glucose levels. In embodiments, the process includes measuring impedance at two frequencies and determining the amount of glucose further includes determining a predetermined index, the index including a ratio of first and second numbers obtained from first and second of the impedance measurements. The first and second numbers can include a component of said first and second impedance measurements, respectively. The first number can be the real part of the complex electrical impedance at the first frequency and the second number can be the magnitude of the complex electrical impedance at the second frequency. The first number can be the imaginary part of the complex electrical impedance at the first frequency and the second number can be the magnitude of the complex electrical impedance at the second frequency. The first number can be the magnitude of the complex electrical impedance at the first frequency and the second number can be the magnitude of the complex electrical impedance at the second frequency. In another embodiment, determining the amount of glucose further includes determining a predetermined index in which the index includes a difference between first and second numbers obtained from first and second of said impedance measurements. The first number can be the phase angle of the complex electrical impedance at the first frequency and said second number can be the phase angle of the complex electrical impedance at the second frequency.

[0094] The electrodes can be in operative connection with the processor programmed to determine the amount of glucose in the body fluid based upon the measured impedance. In certain embodiments, the processor wireless communicates with an insulin pump programmed to adjust the amount of insulin flow via the pump to the subject in response to the determined amount of glucose. The BIA electrodes can be spaced between about 0.2 mm and about 2 cm from each other.

[0095] In another aspect, the BI sensor provides non-invasive monitoring of glucose in a body fluid of a subject. The apparatus includes means for measuring impedance of skin tissue in response to a voltage applied thereto and a microprocessor operatively connected to the means for measuring impedance, for determining the amount of glucose in the body fluid based upon the impedance measurement(s). The means for measuring impedance of skin tissue can include a pair of spaced apart electrodes for electrically conductive contact with a skin surface. The microprocessor can be programmed to compare the measured impedance with a predetermined correlation between impedance and blood glucose level. The apparatus can include means for measuring impedance at a plurality of frequencies of the applied voltage and the program can include means for determining the ratio of one or more pairs of the impedance measurements and means for comparing the determined ratio(s) with corresponding predetermined ratio(s) to determine the amount of glucose in the body fluid.

[0096] In a particular embodiment, the apparatus includes means for calibrating the apparatus against a directly measured glucose level of a said subject. The apparatus can thus include means for inputting the value of the directly measured glucose level in conjunction with impedance measured about the same time, for use by the program to determine the blood glucose level of that subject at a later time based solely on subsequent impedance measurements.

[0097] One embodiment measures BI at 31 different frequencies logarithmically distributed in the range of 1 kHz to 1 MHz (10 frequencies per decade). Another embodiment measures BI at two of the frequencies: 20 and 500 kHz; and in the second set of experiments, 20 kHz only. It may be found in the future that there is a more optimal frequency or frequencies. It is quite possible, in a commercially acceptable instrument that impedance will be determined at at least two frequencies, rather than only one. For practical reasons of instrumentation, the upper frequency at which impedance is measured is likely to be about 500 kHz, but higher frequencies, even as high as 5 MHz or higher are possible and are considered to be within the scope of this invention. Relationships may be established using data obtained at one, two or more frequencies.

[0098] One embodiment, specifically for determining glucose levels of a subject, includes a 2-pole BI measurement configuration that measures impedance at multiple frequencies, preferably two well spaced apart frequencies. The instrument includes a computer which also calculates the index or indices that correlate with blood glucose levels and determines the glucose levels based on the correlation(s), an artificial neural network to perform a non-linear regression.

[0099] In another embodiment, a BI sensor can estimate sugar content in human blood based on variation of dielectric permeability of a finger placed in the electrical field of transducer. The amount of sugar in human blood can also be estimated by changing the reactance of oscillating circuits included in the secondary circuits of high-frequency generator via direct action of human upon oscillating circuits elements. With this method, the amount of sugar in blood is determined based on variation of current in the secondary circuits of high-frequency generator. In another embodiment, a spectral analysis of high-frequency radiation reflected by human body or passing through the human body is conducted. The phase shift between direct and reflected (or transmitted) waves, which characterizes the reactive component of electrical impedance, represents a parameter to be measured by this method. The concentration of substances contained in the blood (in particular, glucose concentration) is determined based on measured parameters of phase spectrum. In another embodiment, glucose concentration is determined by this device based on measurement of human body region impedance at two frequencies, determining capacitive component of impedance and converting the obtained value of capacitive component into glucose concentration in patient's blood. Another embodiment measures impedance between two electrodes at a number of frequencies and deriving the value of glucose concentration on the basis of measured values. In another embodiment, the concentration of glucose in blood is determined based on mathematical model.

[0100] The microphone can also detect respiration. Breathing creates turbulence within the airways, so that the turbulent airflow can be measured using a microphone placed externally on the upper chest at the suprasternal

notch. The respiratory signals recorded inside the ear canal are weak, and are affected by motion artifacts arising from a significant movement of the earpiece inside the ear canal. A control loop involving knowledge of the degree of artifacts and total output power from the microphones can be used for denoising purposes from jaw movements. Denoising can be done for EEG, ECG, PPG waveforms.

[0101] An infrared sensor unit can detect temperature detection in conjunction with an optical identification of objects allows for more reliable identification of the objects, e.g. of the eardrum. Providing the device additionally with an infrared sensor unit, especially arranged centrically at the distal tip, allows for minimizing any risk of misdiagnosis.

[0102] In one implementation information relating to characteristics of the patient's tympanic cavity can be evaluated or processed. In this case the electronics includes a camera that detects serous or mucous fluid within the tympanic cavity can be an indicator of the eardrum itself, and can be an indicator of a pathologic condition in the middle ear. Within the ear canal, only behind the eardrum, such body fluid can be identified. Thus, evidence of any body fluid can provide evidence of the eardrum itself, as well as evidence of a pathologic condition, e.g. OME.

[0103] In a method according to the preferred embodiment, preferably, an intensity of illumination provided by the at least one light source is adjusted such that light emitted by the at least one light source is arranged for at least partially transilluminating the eardrum in such a way that it can be reflected at least partially by any object or body fluid within the subject's tympanic cavity arranged behind the eardrum. The preferred embodiment is based on the finding that translucent characteristics of the eardrum can be evaluated in order to distinguish between different objects within the ear canal, especially in order to identify the eardrum more reliably. Thereby, illumination can be adjusted such that tissue or hard bone confining the ear canal is overexposed, providing reflections (reflected radiation or light), especially reflections within a known spectrum, which can be ignored, i.e. automatically subtracted out. Such a method enables identification of the eardrum more reliably.

[0104] Transilluminating the eardrum can provide supplemental information with respect to the characteristics of the eardrum (e.g. the shape, especially a convexity of the eardrum), and/or with respect to the presence of any fluid within the tympanic cavity. Spectral patterns of reflected light which are typical for eardrum reflection and tympanic cavity reflection can be used to determine the area of interest as well as a physiologic or pathologic condition of the eardrum and the tympanic cavity, especially in conjunction with feedback controlled illumination.

[0105] The camera and process can identify pattern recognition of geometrical patterns, especially circular or ellipsoid shapes, or geometrical patterns characterizing the malleus bone, or further anatomical characteristics of the outer ear or the middle ear. Pattern recognition allows for more reliable identification of the eardrum. Pattern recognition can comprise recognition based on features and shapes such as the shape of e.g. the malleus, the malleus handle, the eardrum or specific portions of the eardrum such as the pars flaccida or the fibrocartilagenous ring. In particular, pattern recognition may comprise edge detection and/or spectral analysis, especially shape detection of a circular or ellipsoid shape with an angular interruption at the malleus bone or pars flaccida.

[0106] In a method according to the preferred embodiment, preferably, the method further comprises calibrating a spectral sensitivity of the electronic imaging unit and/or calibrating color and/or brightness of the at least one light source. Calibration allows for more reliable identification of objects. It has been found that in case the light intensity is very high allowing for passing light through a healthy eardrum, which is semitransparent, a considerable amount of light within the red spectrum can be reflected by the tympanic cavity (especially due to illumination of red mucosa confining the middle ear). Thus, calibrating brightness or the intensity of emitted light enables more accurate evaluation of the (absolute) degree of red channel reflection and its source. In other words, spectral calibration of the imaging sensor in combination with spectral calibration of the illumination means allows for the evaluation of the tissue types and conditions.

[0107] Calibration can be carried out e.g. based on feedback illumination control with respect to different objects or different kinds of tissue, once the respective object or tissue has been identified. Thereby, spectral norm curves with respect to different light intensities provide further data based on which calibration can be carried out.

[0108] FIG. 3D shows an earpiece 50 that has one or more sensors 52, a processor 54, a microphone 56, and a speaker 58. The earpiece 50 may be shaped and sized for an ear canal of a subject. The outer portion of the earpiece 50 can include the microphone as well as touch sensors capturing user input (such as to increase/decrease volume, or to indicate a verbal question to the LLM). The transducer 52 may be any of the previously discussed sensors (EEG, ECG, camera, temperature, pressure, among others). In general, the sensor 52 may be positioned within the earpiece at a position that, when the earpiece 50 is placed for use in the ear canal, corresponds to a location on a surface of the ear canal that exhibits a substantial shape change correlated to a musculoskeletal movement of the subject. The position depicted in FIG. 3B is provided by way of example only, and it will be understood that any position exhibiting substantial displacement may be used to position the sensor(s) 52 for use as contemplated herein. In one aspect, the sensor 52 may be positioned at a position that, when the earpiece is placed for use in the ear canal, corresponds to a location on a surface of the ear canal that exhibits a maximum surface displacement from a neutral position in response to the musculoskeletal movement of the subject. In another aspect, the transducer 52 may be positioned at a position that, when the earpiece is placed for use in the ear canal, corresponds to a location on a surface of the ear canal that exceeds an average surface displacement from a neutral position in response to the musculoskeletal movement of the subject. It will be understood that, while a single transducer 52 is depicted, a number of transducers may be included, which may detect different musculoskeletal movements, or may be coordinated to more accurately detect a single musculoskeletal movement.

[0109] The processor 54 may be coupled to the microphone 56, speaker 58, and sensor(s) 52, and may be configured to detect the musculoskeletal movement of the subject based upon a pressure change signal from the transducer 52, and to generate a predetermined control signal in response to the musculoskeletal movement. The predetermined control signal may, for example, be a mute signal for the earpiece, a volume change signal for the earpiece, or, where the earpiece is an earbud for an audio

player (in which case the microphone **56** may optionally be omitted), a track change signal for the audio player coupled to the earpiece.

[0110] FIG. 4A shows a deployed device **30** that is plugged into a sound cable. The device has an outer face **32** that includes a microphone or microphone array and energy scavenger on one side. The other side is an output port **34** that is snugged fitted to the ear canal using a custom ear insert such as that of FIG. 2A-2B.

[0111] FIG. 4B shows in more detail the device **30** having an outer face **32** that can include microphone or microphone array. Further, the device can harvest energy with solar cells on the face **32**. The unit can optionally connect to a remote microphone, camera, cellular transceiver, and/or battery via cable **34**. The cable or output port **34** can be connected to a behind-the-ear clip-on housing, for example. The device **30** has a custom form protrusion or extension **36** that is custom to the wearer's ear structures. The extension **36** can include hearing aid electronics and/or body sensors as detailed above.

[0112] In one particular variation for treating tinnitus, device may utilize an audio signal, such as music and in particular music having a dynamic signal with intensities varying over time with multiple peaks and troughs throughout the signal. Other audio signals such as various sounds of nature, e.g., rainfall, wind, waves, etc., or other signals such as voice or speech may alternatively be used so long as the audio signal is dynamic. This audio signal may be modified according to a masking algorithm and applied through the device **14** and to the patient to partially mask the patient's tinnitus. An example of how an audio signal may be modified is described in detail in U.S. Pat. No. 6,682,472 (Davis), which is incorporated herein by reference in its entirety and describes a tinnitus treatment which utilizes software to spectrally modify the audio signal in accordance with a predetermined masking algorithm which modifies the intensity of the audio signal at selected frequencies. The described predetermined masking algorithm provides intermittent masking of the tinnitus where the tinnitus is completely masked during peaks in the audio signal and where the perceived tinnitus is detectable to the patient during troughs in the audio signal. Such an algorithm provides for training and habituation by the patient of their tinnitus. Accordingly, the intensity of the audio signal may be modified across the spectrum of the signal and may also be modified to account for any hearing loss that the patient may have incurred. The audio signal having a dynamic spectrum with varying intensities. The audio signal may completely mask the patient's tinnitus during peaks in the signal while during troughs in the audio signal, the tinnitus may be perceived by the patient. Moreover, the masking algorithm may be modified to account for any hearing loss of the patient.

[0113] FIGS. 5A-5B show an AR version where the earpiece **30** includes a 5G transceiver in the unit **30** that is connected to antenna, mike and camera in an extension **39**, which in turn projects from the outer face **32**. The extension **39** is semi-rigid and can be adjusted by the user to aim at predetermined areas. In one embodiment, a plurality of extensions **39** extend like whiskers or hairs that provides signal reception but hard to see. In other embodiments, a single extension **39** includes antenna(s) wrapped on the outside of the extension while Images can be captured and transmitted over fiber optics to a high resolution imager.

Sound vibration can also be optically captured where incident sound waves modulate the light guided in optical fibers without the help of electricity. Intensity modulation by beam deflection at a moving membrane. For example, an optical mike from PKI can be used.

[0114] To supplement the whisker antenna, the front and edge of the earpiece **30** has 3D printed MIMO antennas for Wifi, Bluetooth, and 5G signals. The extension **39** further includes a microphone and camera at the tip to capture audio visual information to aid the user as an augmented reality system. The earpiece contains an inertial measurement unit (IMU) coupled to the intelligent earpiece. The IMU is configured to detect inertial measurement data that corresponds to a positioning, velocity, or acceleration of the intelligent earpiece. The earpiece also contains a global positioning system (GPS) unit coupled to the earpiece that is configured to detect location data corresponding to a location of the intelligent earpiece. At least one camera is coupled to the intelligent earpiece and is configured to detect image data corresponding to a surrounding environment of the intelligent guidance device.

[0115] In one embodiment, the earpiece **30** is a standalone 5G phone controlled by a machine computer interface through the EEG sensor as detailed above. In another variation, voice command can be issued to the 5G phone. In another variation, a foldable display is provided with a larger battery and 5G transceiver. When the user opens the foldable display, the data processing including deep learning operations, 5G transceiver communication, IMU and GPS operations are automatically transferred to the foldable display to conserve power.

[0116] The earpiece for providing social and environmental awareness can continuously observe the user and his surroundings as well as store preference information, such as calendars and schedules, and access remote databases. Based on this observed data, the earpiece can proactively provide feedback to the user. Proactive functions can, for example, remind a user where he should be, inform the user of the name of a person he is speaking with, warn the user when the user may be approaching a hazardous situation, etc. This is advantageous over the state of the art because the user of the earpiece can be provided information without having to request it. This can result in the user being provided feedback that he may not have known he could receive. Additionally, it allows the user to receive feedback without wasting extra time or effort. In some circumstances, this proactive feedback can prevent potential embarrassment for the user (for example, he need not ask the earpiece the name of a person he is speaking with). When left and right earpieces are deployed, the stereo microphones and cameras of the system provide depth and distance information to the device. The device can then use this information to better determine social and environmental elements around the user. The combination of the global positioning system (GPS), the inertial measurement unit (IMU) and the camera is advantageous as the combination can provide more accurate feedback to the user. In one embodiment, the earpiece relies on the GPS and IMU and 5G streams from a smart phone to minimize size and power consumption. The earpiece can use smart phone memory to store object data regarding previously determined objects. The memory also stores previously determined user data associated with the user. The earpiece also includes a processor connected to the IMU, the GPS unit and the at least one camera. The

processor is configured to recognize an object in the surrounding environment. This is done by analyzing the image data based on the stored object data and at least one of the inertial measurement data or the location data. The processor is also configured to determine a desirable event or action based on the recognized object, the previously determined user data, and a current time or day. The processor is also configured to determine a destination based on the determined desirable event or action. The processor is also configured to determine a navigation path for navigating the intelligent guidance device to the destination. This is determined based on the determined destination, the image data, and at least one of the inertial measurement data or the location data. The processor is also configured to determine output data based on the determined navigation path. A speaker is included that is configured to provide audio information to the user based on at least one of the recognized object, determined desirable event or action, or navigation path. The AR system provides continuous social and environmental awareness by the earpiece. The method includes detecting, via an inertial measurement unit (IMU), a global position system unit (GPS) or a camera, inertial measurement data corresponding to a positioning, velocity, or acceleration of the earpiece. Location data corresponding to a location of the earpiece or image data corresponding to a surrounding environment of the earpiece is also determined. The method also includes storing, in a memory, object data regarding previously determined objects and previously determined user data regarding the user. The method also includes recognizing, by a processor, an object in the surrounding environment by analyzing the image data based on the stored object data and at least one of the inertial measurement data or the location data. The method further includes determining, by the processor, a desirable event or action based on the recognized object, the previously determined user data, and a current time or day. The processor also determines a destination based on the determined desirable event or action. The processor may determine a navigation path for navigating the intelligent guidance device to the destination based on the determined destination, the image data, and at least one of the inertial measurement data or the location data. The processor may determine output data based on the determined navigation path. The method further includes providing, via a speaker or a vibration unit, audio or haptic information to the user based on at least one of the recognized object, the determined desirable event or action, or the navigation path. In one embodiment, the earpiece of FIGS. 5A-5B also includes an antenna configured to transmit the image data, the inertial measurement data, the location data and the object data to a remote processor and to receive processed data from the remote processor (such as a 5G smart phone or a remote server array, among others). The remote processor is configured to recognize an object in the surrounding environment by analyzing the image data based on the stored object data and at least one of the inertial measurement data or the location data. The remote processor is also configured to determine a desirable event or action based on the recognized object, the previously determined user data, and a current time or day. The remote processor is also configured to determine a destination based on the determined desirable event or action. The remote processor is also configured to determine a navigation path for navigating the intelligent guidance device to the destination based on the determined

destination, the image data, and at least one of the inertial measurement data or the location data. The remote processor is further configured to determine output data based on the determined navigation path. The earpiece also includes a speaker configured to provide audio information to the user based on at least one of the recognized object, determined desirable event or action, or navigation path. For example, the user may give a voice command, "Take me to building X in Y campus." The earpiece may instruct the smart phone download a relevant map if not already stored, or may navigate based on perceived images from the stereo cameras. As the user follows the navigation commands from the earpiece 30, the user may walk by a coffee shop in the morning, and the earpiece 100 would recognize the coffee shop and the time of day, along with the user's habits, and appropriately alert the user. The earpiece 30 may verbally alert the user through the speaker. The user may use an input device or a web page to adjust settings, which for example may control the types of alerts, what details to announce, and other parameters which may relate to object recognition or alert settings. The user may turn on or off certain features as needed. When navigating indoors, the GPS may not provide enough information to a blind user to navigate around obstacles and reach desired locations or features. The earpiece cameras may recognize, for instance, stairs, exits, and restrooms and appropriately match the images with the GPS mapping system to provide better guidance.

[0117] The earpiece 30 can respond to a request which may be, for example, a request to identify a person, identify a room, identify an object, identify any other place, navigate to a certain location such as an address or a particular room in a building, to remind the user of his current action, what color an object is, if an outfit matches, where another person is pointing or looking. For other example, when the detected data suggests that the user requires an opinion of another person, a communication channel may be established with a device of another person. For example, when the detected speech regarding an outfit of the user, facial recognition data regarding the user being indecisive or wondering about what to wear, and/or perceived action of a user in front of a mirror indicate that the user needs fashion advice from another person, a video teleconference between the user and a friend of the user may be established. From prior conversations/interactions, the earpiece may have previously stored a user's friend's contact information. The processor may categorize types of friends of the user and recognize that this communication needs to be with a friend that the user is comfortable with. The processor may output data to the user letting the user know that a video conference or teleconference will be established with the friend. The earpiece 30 may provide a video connection to a friend of the user or send a picture of the outfit to a friend of the user. In this example, the friend may provide a response as to whether or not the outfit matches. The friend may also assist the user in finding an alternate outfit that matches.

[0118] In order to program a hearing aid to be tailored to the user's hearing needs, the user's hearing threshold may be measured using a sound-stimulus-producing device and calibrated headphone. The measurement of the hearing threshold may take place in a sound-isolating room. For example, the measurement may occur in a room where there is very little audible noise. The sound-stimulus-producing device and the calibrated headphones may be referred to as an audiometer. As shown in FIG. 6, the audiometer may gen-

erate pure tones at various frequencies between 125 Hz and 12,000 Hz that are representative of the frequency bands in which the tones are included. In the example of FIG. 6, the system will start at 500 Hz with progressing loudness before going to the next row at 1 kHz, 2, 3, 4, 5 and 20 kHz, respectively, and then repeats the test in different ambient environments such as restaurant, office, home, theater, party, concert, among others and records the best audio responses in view of the “noise”. This is how the learning machine or neural network learns the optimum settings based on the specific setting. Moreover, as the user operates the device, the learning system improves its performance. The user can train the device to provide optimal variable processing factors for different listening criteria, such as to maximize speech intelligibility, listening comfort, or pleasantness. In one embodiment, based on the detected environment, the learning machine can adjust amplification and gain control. It can optimize the gain at each or every frequency of the amplifier in a particular acoustic environment. In one aspect, a method includes providing an in-ear device to a user anatomy; determine an audio response chart for a user based on a plurality of environments (restaurant, office, home, theater, party, concert, among others), determining a current environment, and updating the hearing aid parameters to optimize the amplifier response to the specific environment. The environment can be auto detected based on GPS position data or external data such as calendaring data or can be user selected using voice command, for example. In another embodiment, a learning machine automatically selects an optimal set of hearing aid parameters based on ambient sound and other confirmatory data.

[0119] FIG. 7A shows an exemplary learning machine such as an artificial neural network (ANN) that can be trained on optimizing hearing assistance for the user. The ANN or connectionist systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems “learn” to perform tasks by considering examples, generally without being programmed with any task-specific rules. An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it. Neural networks come in many shapes and sizes and with varying degrees of complexity. Deep neural networks are defined as having at least two “hidden” processing layers, which are not directly connected to a system’s input or output. Each hidden layer refines the results fed to it by previous layers, adding in new considerations based on prior knowledge.

[0120] In one embodiment, machine learning is used for segregating sounds. The deep learning network can also be used to identify user health. In embodiments that measure user health with heart rate, BI, ECG, EEG, temperature, or other health parameters, if an outlier situation exists, the system can flag to the user to follow up as an unusual, sustained variation from normal health parameters. While this approach may not identify exact causes of the variation, the user can seek help early. FIG. 7B shows an exemplary analysis with normal targets and outliers as warning labels. For example, a patient may be mostly healthy, but when he or she is sick, the information pops out as outliers from the

usual data. Such outliers can be used to scrutinize and predict patient health. The data can be population based, namely that if a population spatially or temporally has the same symptoms, and upon checking with the medical hospitals or doctors to confirm the prediction, public health warnings can be generated. There are two main kinds of machine learning techniques: Supervised learning: in this approach, a training data sample with known relationships between variables is submitted iteratively to the learning algorithm until quantitative evidence (“error convergence”) indicates that it was able to find a solution which minimizes classification error. Several types of artificial neural networks work according to this principle; and Unsupervised learning: in this approach, the data sample is analyzed according to some statistical technique, such as multivariate regression analysis, principal components analysis, cluster analysis, etc., and automatic classification of the data objects into subclasses might be achieved, without the need for a training data set.

[0121] Medical prognosis can be used to predict the future evolution of disease on the basis of data extracted from known cases such as the prediction of mortality of patients admitted to the Intensive Care Unit, using physiological and pathological variables collected at admission. Medical diagnosis can be done, where ML is used to learn the relationship between several input variables (such as signs, symptoms, patient history, lab tests, images, etc.) and several output variables (the diagnosis categories). An example from my research: using symptoms related by patients with psychosis, an automatic classification system was devised to propose diagnoses of a particular disease. Medical therapeutic decisions can be done where ML is used to propose different therapies or patient management strategies, drugs, etc., for a given health condition or diagnosis. Example from my research: patients with different types of brain hematomas (internal bleeding) were used to train a neural network so that a precise indication for surgery was given after having learned the relationships between several input variables and the outcome. Signal or image analysis can be done, where ML is used to learn how features extracted from physiological signals (such as an EKG) or images (such as an x-ray, tomography, etc.) are associated to some diagnoses. ML can even be used to extract features from signals or images, for example, in the so-called “signal segmentation”. Example from my research: non-supervised algorithms were used to extract different image textures from brain MRIs (magnetic resonance imaging), such as bone, meninges, white matter, gray matter, vases, ventricles, etc., and then classifying automatically unknown images, painting each identified region with a different color. In another example large data sets containing multiple variables obtained from individuals in a given population (e.g., those living in a community, or who have a given health care plan, hospital, etc.), are used to train ML algorithms, so as to discover risk associations and predictions (for instance, what patients have a higher risk of emergency risk readmissions or complications from diabetes. Public health can apply ML to predict, for instance, when and where epidemics are going to happen in the future, such as food poisoning, infectious diseases, bouts of environmental diseases, and so on.

[0122] FIG. 7C shows an exemplary system to collect lifestyle and genetic data from various populations for subsequent prediction and recommendation to similarly situated users. The system collects attributes associated with

individuals that co-occur (i.e., co-associate, co-aggregate) with attributes of interest, such as specific disorders, behaviors and traits. The system can identify combinations of attributes that predispose individuals toward having or developing specific disorders, behaviors and traits of interest, determining the level of predisposition of an individual towards such attributes, and revealing which attribute associations can be added or eliminated to effectively modify his or her lifestyle to avoid medical complications. Details captured can be used for improving individualized diagnoses, choosing the most effective therapeutic regimens, making beneficial lifestyle changes that prevent disease and promote health, and reducing associated health care expenditures. It is also desirable to determine those combinations of attributes that promote certain behaviors and traits such as success in sports, music, school, leadership, career and relationships. For example, the system captures information on epigenetic modifications that may be altered due to environmental conditions, life experiences and aging. Along with a collection of diverse nongenetic attributes including physical, behavioral, situational and historical attributes, the system can predict a predisposition of a user toward developing a specific attribute of interest. In addition to genetic and epigenetic attributes, which can be referred to collectively as pangenetic attributes, numerous other attributes likely influence the development of traits and disorders. These other attributes, which can be referred to collectively as non-pangenetic attributes, can be categorized individually as physical, behavioral, or situational attributes.

[0123] FIG. 7C displays one embodiment of the attribute categories and their interrelationships according to the one embodiment and illustrates that physical and behavioral attributes can be collectively equivalent to the broadest classical definition of phenotype, while situational attributes can be equivalent to those typically classified as environmental. In one embodiment, historical attributes can be viewed as a separate category containing a mixture of genetic, epigenetic, physical, behavioral and situational attributes that occurred in the past. Alternatively, historical attributes can be integrated within the genetic, epigenetic, physical, behavioral and situational categories provided they are made readily distinguishable from those attributes that describe the individual's current state. In one embodiment, the historical nature of an attribute is accounted for via a time stamp or other time-based marker associated with the attribute. As such, there are no explicit historical attributes, but through use of time stamping, the time associated with the attribute can be used to make a determination as to whether the attribute is occurring in what would be considered the present, or if it has occurred in the past. Traditional demographic factors are typically a small subset of attributes derived from the phenotype and environmental categories and can be therefore represented within the physical, behavioral and situational categories.

[0124] Since the system captures information from various diverse populations, the data can be mined to discover combinations of attributes regardless of number or type, in a population of any size, that cause predisposition to an attribute of interest. The ability to accurately detect predisposing attribute combinations naturally benefits from being supplied with datasets representing large numbers of individuals and having a large number and variety of attributes for each. Nevertheless, the one embodiment will function properly with a minimal number of individuals and attri-

butes. One embodiment of the one embodiment can be used to detect not only attributes that have a direct (causal) effect on an attribute of interest, but also those attributes that do not have a direct effect such as instrumental variables (i.e., correlative attributes), which are attributes that correlate with and can be used to predict predisposition for the attribute of interest but are not causal. For simplicity of terminology, both types of attributes are referred to herein as predisposing attributes, or simply attributes, that contribute toward predisposition toward the attribute of interest, regardless of whether the contribution or correlation is direct or indirect.

[0125] FIG. 7D shows a deep learning machine using deep convolutionary neural networks for detecting genetic based drug-drug interaction. One embodiment uses an AlexNet: 8-layer architecture, while another embodiment uses a VGGNet: 16-layer architecture (each pooling layer and last 2 FC layers are applied as feature vector). In one embodiment for drugs, the indications of use and other drugs used capture most of many important covariates. One embodiment access data from SIDER (a text-mined database of drug package inserts), the Offsides database that contains information complementary to that found in SIDER and improves the prediction of protein targets and drug indications, and the Twosides database of mined putative DDIs also lists predicted adverse events, all available at the <http://PharmGKB.org> Web site.

[0126] The system of FIG. 7D receives data on adverse events strongly associated with indications for which the indication and the adverse event have a known causative relationship. A drug-event association is synthetic if it has a tight reporting correlation with the indication ($p \geq 0.1$) and a high relative reporting (RR) association score ($RR \geq 2$). Drugs reported frequently with these indications were 80.0 (95% CI, 14.2 to 3132.8; $P < 0.0001$, Fisher's exact test) times as likely to have synthetic associations with indication events. Disease indications are a significant source of synthetic associations. The more disproportionately a drug is reported with an indication (x axis), the more likely that drug will be synthetically associated. For example, adverse events strongly associated with drugs are retrieved from the drug's package insert. These drug-event pairs represent a set of known strong positive associations. Adverse events related to sex and race are also analyzed. For example, for physiological reasons, certain events predominantly occur in males (for example, penile swelling and azoospermia). Drugs that are disproportionately reported as causing adverse events in males were more likely to be synthetically associated with these events. Similarly, adverse events that predominantly occur in either relatively young or relatively old patients are analyzed.

[0127] "Off-label" adverse event data is also analyzed, and off-label uses refer to any drug effect not already listed on the drug's package insert. For example, the SIDER database, extracted from drug package inserts, lists 48,577 drug-event associations for 620 drugs and 1092 adverse events that are also covered by the data mining. Offsides recovers 38.8% (18,842 drug-event associations) of SIDER associations from the adverse event reports. Thus, Offsides finds different associations from those reported during clinical trials before drug approval.

[0128] Polypharmacy side effects for pairs of drugs (Twosides) are also analyzed. These associations are limited to only those that cannot be clearly attributed to either drug

alone (that is, those associations covered in Offsides). The database contains a significant association for which the drug pair has a higher side-effect association score, determined using the proportional reporting ratio (PRR), than those of the individual drugs alone. The system determines pairwise similarity metrics between all drugs in the Offsides and SIDER databases. The system can predict shared protein targets using drug-effect similarities. The side-effect similarity score between two drugs is linearly related to the number of targets that those drugs share.

[0129] The system can determine relationships between the proportion of shared indications between a pair of drugs and the similarity of their side-effect profiles in Offsides. The system can use side-effect profiles to suggest new uses for old drugs. While the preferred system predicts existing therapeutic indications of known drugs, the system can recommend drug repurposing using drug-effect similarities in Offsides. Corroboration of class-wide interaction effects with EMRs. The system can identify DDIs shared by an entire drug class. The class-class interaction analysis generates putative drug class interactions. The system analyzes laboratory reports commonly recorded in EMRs that may be used as markers of these class-specific DDIs.

[0130] In one embodiment, the knowledge-based repository may aggregate relevant clinical and/or behavioral knowledge from one or more sources. In an embodiment, one or more clinical and/or behavioral experts may manually specify the required knowledge. In another embodiment, an ontology-based approach may be used. For example, the knowledge-based repository may leverage the semantic web using techniques, such as statistical relational learning (SRL). SRL may expand probabilistic reasoning to complex relational domains, such as the semantic web. The SRL may achieve this using a combination of representational formalisms (e.g., logic and/or frame based systems with probabilistic models). For example, the SRL may employ Bayesian logic or Markov logic. For example, if there are two objects—‘asian male’ and ‘smartness’, they may be connected using the relationship ‘Asian males are smart’. This relationship may be given a weight (e.g., 0.3). This relationship may vary from time to time (populations trend over years/decades). By leveraging the knowledge in the semantic web (e.g., all references and discussions on the web where ‘blonde’ and ‘smartness’ are used and associated) the degree of relationship may be interpreted from the sentiment of such references (e.g., positive sentiment: TRUE; negative sentiment: FALSE). Such sentiments and the volume of discussions may then be transformed into weights. Accordingly, although the system originally assigned a weight of 0.3, based on information from semantic web about Asian males and smartness, may be revised to 0.9.

[0131] In an embodiment, Markov logic may be applied to the semantic web using two objects: first-order formulae and their weights. The formulae may be acquired based on the semantics of the semantic web languages. In one embodiment, the SRL may acquire the weights based on probability values specified in ontologies. In another embodiment, where the ontologies contain individuals, the individuals can be used to learn weights by generative learning. In some embodiments, the SRL may learn the weights by matching and analyzing a predefined corpus of relevant objects and/or textual resources. These techniques may be used to not only to obtain first-order waited formulae for clinical parameters,

but also general information. This information may then be used when making inferences.

[0132] For example, if the first order logic is ‘obesity causes hypertension, there are two objects involved: obesity and hypertension. If data on patients with obesity and as to whether they were diagnosed with diabetes or not is available, then the weights for this relationship may be learnt from the data. This may be extended to non-clinical examples such as person’s mood, beliefs etc.

[0133] The pattern recognizer may use the temporal dimension of data to learn representations. The pattern recognizer may include a pattern storage system that exploits hierarchy and analytical abilities using a hierarchical network of nodes. The nodes may operate on the input patterns one at a time. For every input pattern, the node may provide one of three operations: 1. Storing patterns, 2. Learning transition probabilities, and 3. Context specific grouping.

[0134] A node may have a memory that stores patterns within the field of view. This memory may permanently store patterns and give each pattern a distinct label (e.g. a pattern number). Patterns that occur in the input field of view of the node may be compared with patterns that are already stored in the memory. If an identical pattern is not in the memory, then the input pattern may be added to the memory and given a distinct pattern number. The pattern number may be arbitrarily assigned and may not reflect any properties of the pattern. In one embodiment, the pattern number may be encoded with one or more properties of the pattern.

[0135] In one embodiment, patterns may be stored in a node as rows of a matrix. In such an embodiment, C may represent a pattern memory matrix. In the pattern memory matrix, each row of C may be a different pattern. These different patterns may be referred to as C-1, C-2, etc., depending on the row in which the pattern is stored.

[0136] The nodes may construct and maintain a Markov graph. The Markov graph may include vertices that correspond to the store patterns. Each vertex may include a label of the pattern that it represents. As new patterns are added to the memory contents, the system may add new vertices to the Markov graph. The system may also create a link between vertices to represent the number of transition events between the patterns corresponding to the vertices. For example, when an input pattern is followed by another input pattern j for the first time, a link may be introduced between the vertices i and j and the number of transition events on that link may be set to 1. System may then increment the number of transition counts on the link from i and j whenever a pattern from i to pattern j is observed. The system may normalize the Markov graph such that the links estimate the probability of a transaction. Normalization may be achieved by dividing the number of transition events on the outgoing links of each vertex by the total number of transition events from the vertex. This may be done for all vertices to obtain a normalized Markov graph. When normalization is completed, the sum of the transition probabilities for each node should add to 1. The system may update the Markov graph continuously to reflect new probability estimates.

[0137] The system may also perform context-specific grouping. To achieve this, the system may partition a set of vertices of the Markov graph into a set of temporal groups. Each temporal group may be a subset of that set of vertices

of the Markov graph. The partitioning may be performed such that the vertices of the same temporal group are highly likely to follow one another.

[0138] The node may use Hierarchical Clustering (HC) to for the temporal groups. The HC algorithm may take a set of pattern labels and their pair-wise similarity measurements as inputs to produce clusters of pattern labels. The system may cluster the pattern labels such that patterns in the same cluster are similar to each other.

[0139] As data is fed into the pattern recognizer, the transition probabilities for each pattern and pattern-of-patterns may be updated based on the Markov graph. This may be achieved by updating the constructed transition probability matrix. This may be done for each pattern in every category of patterns. Those with higher probabilities may be chosen and placed in a separate column in the database called a prediction list.

[0140] Logical relationships among the patterns may be manually defined based on the clinical relevance. This relationship is specified as first-order logic predicates along with probabilities. These probabilities may be called beliefs. In one embodiment, a Bayesian Belief Network (BBN) may be used to make predictions using these beliefs. The BBN may be used to obtain the probability of each occurrence. These logical relationships may also be based on predicates stored the knowledge base.

[0141] The pattern recognizer may also perform optimization for the predictions. In one embodiment, this may be accomplished by comparing the predicted probability for a relationship with its actual occurrence. Then, the difference between the two may be calculated. This may be done for p occurrences of the logic and fed into a K-means clustering algorithm to plot the Euclidean distance between the points. A centroid may be obtained by the algorithm, forming the optimal increment to the difference. This increment may then be added to the (p+1)th occurrence. Then, the process may be repeated. This may be done until the pattern recognizer predicts logical relationships up to a specified accuracy threshold. Then, the results may be considered optimal.

[0142] When a node is at the first level of the hierarchy, its input may come directly from the data source, or after some preprocessing. The input to a node at a higher-level may be the concatenation of the outputs of the nodes that are directly connected to it from a lower level. Patterns in higher-level nodes may represent particular coincidences of their groups of children. This input may be obtained as a probability distribution function (PDF). From this PDF, the probability that a particular group is active may be calculated as the probability of the pattern that has the maximum likelihood among all the patterns belonging to that group.

[0143] The system can use an expert system that can assess hypertension in according with the guidelines. In addition, the expert system can use diagnostic information and apply the following rules to assess hypertension:

[0144] Hemoglobin/hematocrit: Assesses relationship of cells to fluid volume (viscosity) and may indicate risk factors such as hypercoagulability, anemia.

[0145] Blood urea nitrogen (BUN)/creatinine: Provides information about renal perfusion/function.

[0146] Glucose: Hyperglycemia (diabetes mellitus is a precipitator of hypertension) may result from elevated catecholamine levels (increases hypertension).

[0147] Serum potassium: Hypokalemia may indicate the presence of primary aldosteronism (cause) or be a side effect of diuretic-therapy.

[0148] Serum calcium: Imbalance may contribute to hypertension.

[0149] Lipid panel (total lipids, high-density lipoprotein [HDL], low-density lipoprotein [LDL], cholesterol, triglycerides, phospholipids): Elevated level may indicate predisposition for/presence of atherosomatous plaques.

[0150] Thyroid studies: Hyperthyroidism may lead or contribute to vasoconstriction and hypertension. Serum/urine aldosterone level: May be done to assess for primary aldosteronism (cause).

[0151] Urinalysis: May show blood, protein, or white blood cells; or glucose suggests renal dysfunction and/or presence of diabetes.

[0152] Creatinine clearance: May be reduced, reflecting renal damage.

[0153] Urine vanillylmandelic acid (VMA) (catecholamine metabolite): Elevation may indicate presence of pheochromocytoma (cause); 24-hour urine VMA may be done for assessment of pheochromocytoma if hypertension is intermittent.

[0154] Uric acid: Hyperuricemia has been implicated as a risk factor for the development of hypertension.

[0155] Renin: Elevated in renovascular and malignant hypertension, salt-wasting disorders.

[0156] Urine steroids: Elevation may indicate hyperadrenalinism, pheochromocytoma, pituitary dysfunction, Cushing's syndrome.

[0157] Intravenous pyelogram (IVP): May identify cause of secondary hypertension, e.g., renal parenchymal disease, renal/ureteral-calculi.

[0158] Kidney and renography nuclear scan: Evaluates renal status (TOD).

[0159] Excretory urography: May reveal renal atrophy, indicating chronic renal disease.

[0160] Chest x-ray: May demonstrate obstructing calcification in valve areas; deposits in and/or notching of aorta; cardiac enlargement.

[0161] Computed tomography (CT) scan: Assesses for cerebral tumor, CVA, or encephalopathy or to rule out pheochromocytoma.

[0162] Electrocardiogram (ECG): May demonstrate enlarged heart, strain patterns, conduction disturbances. Note: Broad, notched P wave is one of the earliest signs of hypertensive heart disease.

[0163] The system may also be adaptive. In one embodiment, every level has a capability to obtain feedback information from higher levels. This feedback may inform about certain characteristics of information transmitted bottom-up through the network. Such a closed loop may be used to optimize each level's accuracy of inference as well as transmit more relevant information from the next instance.

[0164] The system may learn and correct its operational efficiency over time. This process is known as the maturity process of the system. The maturity process may include one or more of the following flow of steps:

[0165] a. Tracking patterns of input data and identifying predefined patterns (e.g. if the same pattern was observed several times earlier, the pattern would have already taken certain paths in the hierarchical node structure).

[0166] b. Scanning the possible data, other patterns (collectively called Input Sets (IS)) required for those paths. It also may check for any feedback that has come from higher levels of hierarchy. This feedback may be either positive or negative (e.g., the relevance of the information transmitted to the inferences at higher levels). Accordingly, the system may decide whether to send this pattern higher up the levels or not, and if so whether it should it send through a different path.

[0167] c. Checking for frequently required ISs and pick the top ‘F’ percentile of them.

[0168] d. Ensuring it keeps this data ready.

[0169] In one embodiment, information used at every node may act as agents reporting on the status of a hierarchical network. These agents are referred to as Information Entities (In En). In En may provide insight about the respective inference operation, the input, and the result which collectively is called knowledge.

[0170] This knowledge may be different from the KB. For example, the above described knowledge may include the dynamic creation of insights by the system based on its inference, whereas the KB may act as a reference for inference and/or analysis operations. The latter being an input to inference while the former is a product of inference. When this knowledge is subscribed to by a consumer (e.g. administering system or another node in a different layer) it is called “Knowledge-as-a-Service (KaaS)”

[0171] One embodiment processes behavior models are classified into four categories as follows:

[0172] a. Outcome-based;

[0173] b. Behavior-based;

[0174] c. Determinant-based; and

[0175] d. Intervention-based.

[0176] One or more of the following rules of thumb may be applied during behavioral modeling:

[0177] One or more interventions affect determinants;

[0178] One or more determinants affect behavior; and

[0179] One or more behaviors affect outcome.

[0180] A behavior is defined to be a characteristic of an individual or a group towards certain aspects of their life such as health, social interactions, etc. These characteristics are displayed as their attitude towards such aspects. In analytical terms, a behavior can be considered similar to a habit. Hence, a behavior may be observed POPTM for a given data from a user. An example of a behavior is dietary habits.

[0181] Determinants may include causal factors for behaviors. They either cause someone to exhibit the same behavior or cause behavior change. Certain determinants are quantitative but most are qualitative. Examples include one’s perception about a food, their beliefs, their confidence levels, etc.

[0182] Interventions are actions that affect determinants. Indirectly they influence behaviors and hence outcomes. System may get both primary and secondary sources of data. Primary sources may be directly reported by the end-user and AU. Secondary data may be collected from sensors such as their mobile phones, cameras, microphone, as well as those collected from general sources such as the semantic web.

[0183] These data sources may inform the system about the respective interventions. For example, to influence a determinant called forgetfulness which relates to a behavior called medication, the system sends a reminder at an appropriate time, as the intervention. Then, feedback is obtained

whether the user took the medication or not. This helps the system in confirming if the intervention was effective.

[0184] The system may track a user’s interactions and request feedback about their experience through assessments. The system may use this information as part of behavioral modeling to determine if the user interface and the content delivery mechanism have a significant effect on behavior change with the user. The system may use this information to optimize its user interface to make it more personalized over time to best suit the users, as well as to best suit the desired outcome.

[0185] The system also may accommodate data obtained directly from the end-user, such as assessments, surveys, etc. This enables users to share their views on interventions, their effectiveness, possible causes, etc. The system’s understanding of the same aspects is obtained by way of analysis and service by the pattern recognizer.

[0186] Both system-perceived and end user-perceived measures of behavioral factors may be used in a process called Perception Scoring (PS). In this process, hybrid scores may be designed to accommodate both above mentioned aspects of behavioral factors. Belief is the measure of confidence the system has, when communicating or inferring on information. Initially higher beliefs may be set for user-perceived measures.

[0187] Over time, as the system finds increasing patterns as well as obtains feedback in pattern recognizer, the system may evaluate the effectiveness of intervention(s). If the system triggers an intervention based on user-perceived measures and it doesn’t have significant effect on the behavior change, the system may then start reducing its belief for user-perceived measures and instead will increase its belief for system-perceived ones. In other words, the system starts believing less in the user and starts believing more in itself. Eventually this reaches a stage where system can understand end-users and their behavioral health better than end-users themselves. When perception scoring is done for each intervention, it may result in a score called Intervention Effectiveness Score (IES).

[0188] Perception scoring may be done for both end-users as well as AU. Such scores may be included as part of behavior models during cause-effect analysis.

[0189] Causes may be mapped with interventions, determinants, and behavior respectively in order of the relevance. Mapping causes with interventions helps in back-tracking the respective AU for that cause. In simple terms, it may help in identifying whose actions have had a pronounced effect on the end-user’s outcome, by how much and using which intervention. This is very useful in identifying AUs who are very effective with specific interventions as well as during certain event context. Accordingly, they may be provided a score called Associated User Influence Score. This encompasses information for a given end-user, considering all interventions and possible contexts relevant to the user’s case.

[0190] The system may construct one or plans including one or more interventions based on analysis performed, and may be implemented. For example, the system may analyze eligibility of an intervention for a given scenario, evaluating eligibility of two or more interventions based on combinatorial effect, prioritizing interventions to be applied, based on occurrence of patterns (from pattern recognizer), and/or submitting an intervention plan to the user or doctor in a format readily usable for execution.

[0191] This system may rely on the cause-effect analysis for its planning operations. A plan consists of interventions and a respective implementation schedule. Every plan may have several versions based on the users involved in it. For example, the system may have a separate version for the physician as compared to a patient. They will in turn do the task and report back to the system. This can be done either directly or the system may indirectly find it based on whether a desired outcome with the end user was observed or not.

[0192] The methodology may be predefined by an analyst. For every cause, which can be an intervention(s), determinant(s), behavior(s) or combinations of the same, the analyst may specify one or more remedial actions. This may be specified from the causal perspective and not the contextual perspective.

[0193] Accordingly, the system may send a variety of data and information to pattern recognizer and other services, as feedback, for these services to understand about the users. This understanding may affect their next set of plans which in turn becomes an infinite cyclic system where system affects the users while getting affected by them at the same time. Such a system is called a reflexive-feedback enabled system. The system may user both positive and negative reflexive-feedback, though the negative feedback aspect may predominantly be used for identifying gaps that the system needs to address.

[0194] The system may provide information, such as one or more newly identified patterns, to an analyst (e.g., clinical analyst or doctor). In the use case, the doctor may be presented with one or more notifications to address the relationship between carbohydrates and the medication that the patient is taking.

[0195] One embodiment of the system operation includes receiving feedback relating to the plan, and revising the plan based on the feedback; the feedback being one or more patient behaviors that occur after the plan; the revised plan including one or more additional interventions selected based on the feedback; the one or more patient behaviors that occur after the plan include a behavior transition; determining one or more persons to associate with the identified intervention; automatically revising probabilities from the collected information; storing the revised probabilities, wherein the revised probabilities are used to determine the plan; and/or automatically make one or more inferences based on machine learning using one or more of the clinical information, behavior information, or personal information.

[0196] Hypertension metrics may be one type of metrics utilized within the principles of the present disclosure. A hypertension score can be based on any type of alphanumeric or visual analog scale. Hypertension scales may or may not be clinically validated and may use any scale (e.g. 1-100, 1-10, 1-4), picture, symbol, color, character, number, sound, letter, or written description of hypertension to facilitate the communication of a patient's hypertension level. The type of hypertension scale used may be determined according to a patient's and/or healthcare provider's preferences, and may also be determined based on the needs of a patient including, for example, the patient's age and/or communication capability. In further embodiments, the selected hypertension scale(s) may be determined by a

service provider, such as, e.g., an organization implementing the principles of the present disclosure via a suitable software program or application.

[0197] Another metric may include a functionality score. A functionality score can be based on any type of alphanumeric or visual analog scale. Non-limiting examples include the American Chronic Pain Association Quality of Life (ACPA QOL) Scale, Global Assessment of Functioning (GAF) Scale, and Short Form SF-36 Health Survey. Functionality scales may or may not be clinically validated and may use any picture, symbol, color, character, number, sound, letter, written description of quality of life, or physical functioning to facilitate communication of a patient's functionality level. The functionality score may be, e.g., based on an assessment of a patient's ability to exercise as well as perform daily tasks and/or perform routine tasks such as, e.g., getting dressed, grocery shopping, cooking, cleaning, climbing stairs, etc. In some embodiments, the selected functionality scale(s) may be determined by a service provider, such as, e.g., an organization implementing the principles of the present disclosure via a suitable software program or application.

[0198] A further metric may include a patient's medication usage. Medication use encompasses pharmacologic and therapeutic agents used to treat, control, and/or alleviate hypertension, including prescription drugs as well as over-the-counter medications, therapeutic agents, and other non-prescription agents. Medication use may include different classes of pharmacologic agents. Medication use can be reported in any appropriate units, such as number of doses taken, percentage of treatment plan completed, frequency of doses, and/or dose strength; and may also specify additional information such as the type of formulation taken and the route of administration (oral, enteral, topical, transdermal, parenteral, sublingual etc.). Molecular alternatives (e.g., acid, salt, solvate, complex, and pro-drug forms, etc.) and formulations (e.g., solid, liquid, powder, gel, and suspensions, etc.) are further contemplated. Reported medication use may, for example, include the number of doses and types of medication taken since a previous reported medication use, and may also indicate the number of closes and types of medication taken within a period of time, such as within, the previous 2 hours, 4 hours, 6 hours, 12 hours, 18 hours, 24 hours, 36 hours, or 48 hours. In some embodiments, for example, medication use may be reported in terms of dosage units recommended by a manufacturer or healthcare provider for a given medication (e.g., minimum, maximum, or range of appropriate unit dosage per unit time).

[0199] Reported medication use may allow for tracking compliance with a treatment regime. For example, a record of reported medication use may assist a healthcare provider in evaluating medication efficacy, adjusting dosage, and/or adding other medications as necessary.

[0200] In some embodiments of the present disclosure, a patient or healthcare provider may create a patient profile comprising, e.g., identifying, characterizing, and/or medical information, including information about a patient's medical history, profession, and/or lifestyle. Further examples of information that may be stored in a patient profile includes diagnostic information such as family medical history, medical symptoms, duration of hypertension, localized vs. general hypertension, etc. Further contemplated as part of a patient profile are non-pharmacologic treatment(s) (e.g., chiropractic, radiation, holistic, psychological, acupuncture,

etc.), lifestyle characteristics (e.g., diet, alcohol intake, smoking habits), cognitive condition, behavioral health, and social well-being.

[0201] A patient profile may, for example, be stored in a database and accessible for analysis of the patient's reported hypertension metrics. In some embodiments, a patient profile may be created before collecting and/or transmitting a set of hypertension metrics to be received by a server and/or database in other embodiments, a patient profile may be created concurrently with, or even after transmitting/receiving one or more hypertension metrics. In some embodiments a patient profile may be used to establish one or more hypertension metric(s) and/or reference values. A patient profile may, for example, allow for setting threshold values or ranges, wherein reported hypertension metrics that fall outside of those limits trigger an alert to be sent to the patient or a healthcare provider. Threshold values, limits, or ranges may also be set without reference to a patient profile. In some embodiments, one or more target value(s) (e.g., hypertension metric value(s)) may be set to determine how the reported hypertension metrics compare with the target value(s).

[0202] The methods and systems disclosed herein may rely on one or more algorithm(s) to analyze one or more of the described metrics. The algorithm(s) may comprise analysis of data reported in real-time, and may also analyze data reported in real-time in conjunction with auxiliary data stored in a hypertension management database. Such auxiliary data may comprise, for example, historical patient data such as previously-reported hypertension metrics (e.g., hypertension scores, functionality scores, medication use), personal medical history, and/or family medical history. In some embodiments, for example, the auxiliary data includes at least one set of hypertension metrics previously reported and stored for a patient. In some embodiments, the auxiliary data includes a patient profile such as, e.g., the patient profile described above. Auxiliary data may also include statistical data, such as hypertension metrics pooled for a plurality of patients within a similar group or subgroup. Further, auxiliary data may include clinical guidelines such as guidelines relating to hypertension management, including evidence-based clinical practice guidelines on the management of acute and/or chronic hypertension or other chronic conditions.

[0203] Analysis of a set of hypertension metrics according to the present disclosure may allow for calibration of the level, degree, and/or quality of hypertension experienced by providing greater context to patient-reported data. For example, associating a hypertension score of 7 out of 10 with high functionality for a first patient, and the same score with low functionality for a second patient may indicate a relatively greater debilitating effect of hypertension on the second patient than the first patient. Further, a high hypertension score reported by a patient taking a particular medication such as opioid analgesics may indicate a need to adjust the patient's treatment plan. Further, the methods and systems disclosed herein may provide a means of assessing relative changes in a patient's distress due to hypertension over time. For example, a hypertension score of 5 out of 10 for a patient who previously reported consistently lower hypertension scores, e.g., 1 out of 10, may indicate a serious issue requiring immediate medical attention.

[0204] Any combination(s) of hypertension metrics may be used for analysis in the systems and methods disclosed.

In some embodiments, for example, the set of hypertension metrics comprises at least one hypertension score and at least one functionality score. In other embodiments, the set of hypertension metrics may comprise at least one hypertension score, at least one functionality score, and medication use. More than one set of hypertension metrics may be reported and analyzed at a given time. For example, a first set of hypertension metrics recording a patient's current status and a second set of hypertension metrics recording the patient's status at an earlier time may both be analyzed and may also be used to generate one or more recommended actions.

[0205] Each hypertension metric may be given equal weight in the analysis, or may also be given greater or less weight than other hypertension metrics included in the analysis. For example, a functionality score may be given greater or less weight with respect to a hypertension score and/or medication use. Whether and/or how to weigh a given hypertension metric may be determined according to the characteristics or needs of a particular patient. As an example, Patient A reports a hypertension score of 8 (on a scale of 1 to 10 where 10 is the most severe hypertension) and a functionality score of 9 (on a scale of 1 to 10 where 10 is highest functioning), while Patient B reports a hypertension score of 8 but a functionality score of 4. The present disclosure provides for the collection, analysis, and reporting of this information, taking into account the differential impact of one hypertension score on a patient's functionality versus that same hypertension score's impact on the functionality of a different patient.

[0206] Hypertension metrics may undergo a pre-analysis before inclusion in a set of hypertension metrics and subsequent application of one or more algorithms. For example, a raw score may be converted or scaled according to one or more algorithm(s) developed for a particular patient. In some embodiments, for example, a non-numerical raw score may be converted to a numerical score or otherwise quantified prior to the application of one or more algorithms. Patients and healthcare providers may retain access to raw data (e.g., hypertension metric data prior to any analysis)

[0207] Algorithm(s) according to the present disclosure may analyze the set of hypertension metrics according to any suitable methods known in the art. Analysis may comprise, for example, calculation of statistical averages, pattern recognition, application of mathematical models, factor analysis, correlation, and/or regression analysis. Examples of analyses that may be used herein include, but are not limited to, those disclosed in U.S. Patent Application Publication No. 2012/0246102 A1 the entirety of which is incorporated herein by reference.

[0208] The present disclosure further provides for the determination of an aggregated hypertension assessment score. In some embodiments, for example, a set of pairs metrics may be analyzed to generate a comprehensive and/or individualized assessment of hypertension by generating a composite or aggregated score. In such embodiments, the aggregated score may include a combination of at least one hypertension score, at least one functionality score, and medication use. Additional metrics may also be included in the aggregated score. Such metrics may include, but are not limited to, exercise habits, mental well-being, depression, cognitive functioning, medication side effects, etc. Any of the aforementioned types of analyses may be used in determining an aggregated score.

[0209] The algorithm(s) may include a software program that may be available for download to an input device in various versions. In some embodiments, for example, the algorithm(s) may be directly downloaded through the Internet or other suitable communications means to provide the capability to troubleshoot a health issue in real-time. The algorithm(s) may also be periodically updated, e.g., provided content changes, and may also be made available for download to an input device.

[0210] The methods presently disclosed may provide a healthcare provider with a more complete record of a patient's day-to-day status. By having access to a consistent data stream of hypertension metrics for a patient, a healthcare provider may be able to provide the patient with timely advice and real-time coaching on hypertension management options and solutions. A patient may, for example, seek and/or receive feedback on hypertension management without waiting for an upcoming appointment with a healthcare provider or scheduling a new appointment. Such real-time communication capability may be especially beneficial to provide patients with guidance and treatment options during intervals between appointments with a healthcare provider. Healthcare providers may also be able to monitor a patient's status between appointments to timely initiate, modify, or terminate a treatment plan as necessary. For example, a patient's reported medication use may convey whether the patient is taking too little or too much medication. In some embodiments, an alert may be triggered to notify the patient and/or a healthcare provider of the amount of medication taken, e.g., in comparison to a prescribed treatment plan. The healthcare provider could, for example, contact the patient to discuss the treatment plan. The methods disclosed herein may also provide a healthcare provider with a longitudinal review of how a patient responds to hypertension over time. For example, a healthcare provider may be able to determine whether a given treatment plan adequately addresses a patient's needs based on review of the patient's reported hypertension metrics and analysis thereof according to the present disclosure.

[0211] Analysis of patient data according to the methods presently disclosed may generate one or more recommended actions that may be transmitted and displayed on an output device. In some embodiments, the analysis recommends that a patient make no changes to his/her treatment plan or routine. In other embodiments, the analysis generates a recommendation that the patient seek further consultation with a healthcare provider and/or establish compliance with a prescribed treatment plan. In other embodiments, the analysis may encourage a patient to seek immediate medical attention. For example, the analysis may generate an alert to be transmitted to one or more output devices, e.g., a first output device belonging to the patient and a second output device belonging to a healthcare provider, indicating that the patient is in need of immediate medical treatment. In some embodiments, the analysis may not generate a recommended action. Other recommended actions consistent with the present disclosure may be contemplated and suitable according to the treatment plans, needs, and/or preferences for a given patient.

[0212] The present disclosure further provides a means for monitoring a patient's medication use to determine when his/her prescription will run out and require a refill. For example, a patient profile may be created that indicates a prescribed dosage and frequency of administration, as well

as total number of dosages provided in a single prescription. As the patient reports medication use, those hypertension metrics may be transmitted to a server and stored in a database in connection with the patient profile. The patient profile stored on the database may thus continually update with each added metric and generate a notification to indicate when the prescription will run out based on the reported medication use. The notification may be transmitted and displayed on one or more output devices, e.g., to a patient and/or one or more healthcare providers. In some embodiments, the one or more healthcare providers may include a pharmacist. For example, a pharmacist may receive notification of the anticipated date a prescription will run out in order to ensure that the prescription may be timely refilled.

[0213] Patient data can be input for analysis according to the systems disclosed herein through any data-enabled device including, but not limited to, portable/mobile and stationary communication devices, and portable/mobile and stationary computing devices. Non-limiting examples of input devices suitable for the systems disclosed herein include smart phones, cell phones, laptop computers, netbooks, personal computers (PCs), tablet PCs, fax machines, personal digital assistants, and/or personal medical devices. The user interface of the input device may be web-based, such as a web page, or may also be a stand-alone application. Input devices may provide access to software applications via mobile and wireless platforms, and may also include web-based applications.

[0214] The input device may receive data by having a user, including, but not limited to, a patient, family member, friend, guardian, representative, healthcare provider, and/or caregiver, enter particular information via a user interface, such as by typing and/or speaking. In some embodiments, a server may send a request for particular information to be entered by the user via an input device. For example, an input device may prompt a user to enter sequentially a set of hypertension metrics, e.g., a hypertension score, a functionality score, and information regarding use of one or more medications (e.g., type of medication, dosage taken, time of day, route of administration, etc.). In other embodiments, the user may enter data into the input device without first receiving a prompt. For example, the user may initiate an application or web-based software program and select an option to enter one or more hypertension metrics. In some embodiments, one or more hypertension scales and/or functionality scales may be preselected by the application or software program. For example, a user may have the option of selecting the type of hypertension scale and/or functionality scale for reporting hypertension metrics within the application or software program. In other embodiments, an application or software program may not include preselected hypertension scales or functionality scales such that a user can employ any hypertension scale and/or functionality scale of choice.

[0215] The user interface of an input device may allow a user to associate hypertension metrics with a particular date and/or time of day. For example, a user may report one or more hypertension metrics to reflect a patient's present status. A user may also report one or more hypertension metrics to reflect a patient's status at an earlier time.

[0216] Patient data may be electronically transmitted from an input device over a wired or wireless medium to a server, e.g., a remote server. The server may provide access to a database for performing an analysis of the data transmitted,

e.g., set of hypertension metrics. The database may comprise auxiliary data for use in the analysis as described above. In some embodiments, the analysis may be automated, and may also be capable of providing real-time feedback to patients and/or healthcare providers. The analysis may generate one or more recommended actions, and may transmit the recommended action(s) over at wired or wireless medium for display on at least one output device. The at least one output device may include, e.g., portable/mobile and stationary communication devices, and portable/mobile and stationary computing devices. Non-limiting examples of output devices suitable for the systems disclosed herein include smart phones, cell phones, laptop computers, netbooks, personal computers (PCs), tablet PCs, fax machines, personal digital assistants, and/or personal medical devices. In some embodiments, the input device is the at least one output device. In other embodiments, the input device is one of multiple output devices. In some embodiments of the present disclosure, the one or more recommended actions are transmitted and displayed on each of two output devices. In such an example, one output device may belong to a patient and the other device may belong to a healthcare provider. Yet other intervention can include music, image, or video. The music can be synchronized with respect to a blood pulse rate in one embodiment, and in other embodiments to biorhythmic signal—either to match the biorhythmic signal, or, if the signal is too fast or too slow, to go slightly slower or faster than the signal, respectively. In order to entrain the user's breathing, a basic melody is preferably played which can be easily identified by almost all users as corresponding to a particular phase of respiration. On top of the basic melody, additional layers are typically added to make the music more interesting, to the extent required by the current breathing rate, as described hereinabove. Typically, the basic melody corresponding to this breathing includes musical cords, played continuously by the appropriate instrument during each phase. For some applications, it is desirable to elongate slightly the length of one of the respiratory phases, typically, the expiration phase. For example, to achieve respiration which is 70% expiration and 30% inspiration, a musical composition written for an E:I ratio of 2:1 may be played, but the expiration phase is extended by a substantially-unnoticed 16%, so as to produce the desired respiration timing. The expiration phase is typically extended either by slowing down the tempo of the notes therein, or by extending the durations of some or all of the notes.

[0217] Although music for entraining breathing is described hereinabove as including two phases, it will be appreciated by persons skilled in the art that the music may similarly include other numbers of phases, as appropriate. For example, user may be guided towards breathing according to a 1:2:1:3 pattern, corresponding to inspiration, breath holding (widely used in Yoga), expiration, and post-expiratory pause (rest state). In one embodiment, the volume of one or more of the layers is modulated responsive to a respiration characteristic (e.g., inhalation depth, or force), so as to direct the user to change the characteristic, or simply to enhance the user's connection to the music by reflecting therein the respiration characteristic. Alternatively, or additionally, parameters of the sound by each of the musical instruments may be varied to increase the user's enjoyment. For example, during slow breathing, people tend to prefer to hear sound patterns that have smoother structures than

during fast breathing and/or aerobic exercise. Further alternatively or additionally, random musical patterns and/or digitized natural sounds (e.g., sounds of the ocean, rain, or wind) are added as a decoration layer, especially for applications which direct the user into very slow breathing patterns. The inventor has found that during very slow breathing, it is desirable to remove the user's focus from temporal structures, particularly during expiration. Still further alternatively or additionally, the server maintains a musical library, to enable the user to download appropriate music and/or music-generating patterns from the Internet into device. Often, as a user's health improves, the music protocols which were initially stored in the device are no longer optimal, so the user downloads the new protocols, by means of which music is generated that is more suitable for his new breathing training. The following can be done:

- [0218] obtaining clinical data from one or more laboratory test equipment and checking the data on a blockchain;
- [0219] obtaining genetic clinical data from one or more genomic equipment and storing genetic markers in the EMR/HER including germ line data and somatic data over time;
- [0220] obtaining clinical data from a primary care or a specialist physician database;
- [0221] obtaining clinical data from an in-patient care database or from an emergency room database;
- [0222] saving the clinical data into a clinical data repository;
- [0223] obtaining health data from fitness devices or from mobile phones;
- [0224] obtaining behavioral data from social network communications and mobile device usage patterns;
- [0225] saving the health data and behavioral data into a health data repository separate from the clinical data repository; and
- [0226] providing a decision support system (DSS) to apply genetic clinical data to the subject, and in case of an adverse event for a drug or treatment, generating a drug safety signal to alert a doctor or a manufacturer, wherein the DSS includes rule-based alerts on pharmacogenetics, oncology drug regimens, wherein the DSS performs ongoing monitoring of actionable genetic variants.

[0227] FIG. 7E illustrates one embodiment of a system for collaboratively treating a patient with a disease such as cancer. In this embodiment, a treating physician/doctor logs into a consultation system 1 and initiates the process by clicking on “Create New Case” (500). Next, the system presents the doctor with a “New Case Wizard” which provides a simple, guided set of steps to allow the doctor to fill out an “Initial Assessment” form (501). The doctor may enter Patient or Subject Information (502), enter Initial Assessment of patient/case (504), upload Test Results, Subject Photographs and X-Rays (506), accept Payment and Service Terms and Conditions (508), review Summary of Case (510), or submit Forms to a AI machine based “consultant” such as a Hearing Service AI Provider (512). Other clinical information for the cancer subject includes the imaging or medical procedure directed towards the specific disease that one of ordinary skill in the art can readily identify. The list of appropriate sources of clinical information for cancer includes but it is not limited to: CT scan, MRI scan, ultrasound scan, bone scan, PET Scan, bone marrow

test, barium X-ray, endoscopy, lymphangiogram, IVU (Intravenous urogram) or IVP (IV pyelogram), lumbar puncture, cystoscopy, immunological tests (anti-malignant antibody screen), and cancer marker tests.

[0228] After the case has been submitted, the AI Machine Consultant can log into the system 1 and consult/process the case (520). Using the Treating Doctors Initial Assessment and Photos/X-Rays, the Consultant will click on “Case Consultation” to initiate the “Case Consultation Wizard” (522). The consultant can fill out the “Consultant Record Analysis” form (524). The consultant can also complete the “Prescription Form” (526) and submit completed forms to the original Treating Doctor (528). Once the case forms have been completed by the Consulting Doctor, the Treating Doctor can access the completed forms using the system. The Treating Doctor can either accept the consultation results (i.e. a pre-filled Prescription form) or use an integrated messaging system to communicate with the Consultant (530). The Treating Doctor can log into the system (532), click on Patient Name to review (534), review the Consultation Results (Summary Letter and pre-filled Prescription Form) (536). If satisfied, the Treating Doctor can click “Approve Treatment” (538), and this will mark the case as having being approved (540). The Treating Doctor will be able to print a copy of the Prescription Form and the Summary Letter for submission to hearing aid manufacturer or provider (542). Alternatively, if not satisfied, the Treating Doctor can initiate a computer dialog with the Consultant by clicking “Send a Message” (544). The Treating Doctor will be presented with the “Send a Message” screen where a message about the case under consultation can be written (546). After writing a message, the Treating Doctor would click “Submit” to send the message to the appropriate Consultant (548). The Consultant will then be able to reply to the Treating Doctor’s Message and send a message/reply back to the Treating Doctor (550).

Blockchain Authentication

[0229] In some embodiments, the described technology provides a peer-to-peer cryptographic currency trading method for initiating a market exchange of one or more Blockchain tokens in a virtual wallet for purchasing an asset (e.g., a security) at a purchase price. The system can determine, via a two-phase commit, whether the virtual wallet has a sufficient quantity of Blockchain tokens to purchase virtual assets (such as electricity only from renewable solar/wind/ . . . sources, weather data or location data) and physical asset (such as gasoline for automated vehicles) at the purchase price. In various embodiments, in response to verifying via the two-phase commit that the virtual wallet has a sufficient quantity of Blockchain tokens, the IoT machine purchases (or initiates a process in furtherance of purchasing) the asset with at least one of the Blockchain tokens. In one or more embodiments, if the described technology determines that the virtual wallet has insufficient Blockchain tokens for purchasing the asset, the purchase is terminated without exchanging Blockchain tokens.

Cloud Storage Security

[0230] In another aspect, a distributed file storage system includes nodes are incentivized to store as much of the entire network’s data as they can. Blockchain currency is awarded for storing files, and is transferred in Bitcoin or Ether

transactions, as in. Files are added to the network by spending currency. To enable an IoT device such as a car or a robot to access cloud data securely, and to grant access right to agents of the IoT device such as media players in the car, for example, the following methods can be used for accessing data, content, or application stored in a cloud storage, comprising: authorizing a first client device; receiving an authorization request from the first client device; generating an authorization key for accessing the cloud server and storing the key in a blockchain; providing the authorization key to the first client device; receiving the authorization key from an IoT device as a second client device working as an agent of the first client device; granting access to the second client device based on the authorization key; receiving a map of storage locations of cloud objects associated with an application or content, each storage location identified in a blockchain; and reassembling the application or content from the storage locations.

[0231] In implementation, the blockchain is decentralized and does not require a central authority for creation, processing or verification and comprises a public digital ledger of all transactions that have ever been executed on the blockchain and wherein new blocks are added to the blockchain in a linear, chronological order. The public digital ledger of the blockchain comprises transactions and blocks. Blocks in the blockchain record and confirm when and in what sequence transactions are entered and logged into the blockchain. The transactions comprise desired electronic content stored in the blockchain. The desired electronic content includes a financial transaction. The financial transaction includes a cryptocurrency transaction, wherein the cryptocurrency transaction includes a BITCOIN or an ETHEREUM transaction. An identifier for the received one or more blocks in the blockchain includes a private encryption key.

Medical History

[0232] The above permissioned blockchain can be used to share sensitive medical data with different authorized institutions. The institutions are trusted parties and vouched for by the trusted party. A Patient-Provider Relationship (PPR) Smart Contract is issued when one node from a trusted institution stores and manages medical records for the patient. The PPR defines an assortment of data pointers and associated access permissions that identify the records held by the care provider. Each pointer consists of a query string that, when executed on the provider’s database, returns a subset of patient data. The query string is affixed with the hash of this data subset, to guarantee that data have not been altered at the source. Additional information indicates where the provider’s database can be accessed in the network, i.e. hostname and port in a standard network topology. The data queries and their associated information are crafted by the care provider and modified when new records are added. To enable patients to share records with others, a dictionary implementation (hash table) maps viewers’ addresses to a list of additional query strings. Each string can specify a portion of the patient’s data to which the third party viewer is allowed access. For SQL data queries, a provider references the patient’s data with a SELECT query on the patient’s address. For patients uses an interface that allows them to check off fields they wish to share through a graphical interface. The system formulates the appropriate SQL queries and uploads them to the PPR on the blockchain.

[0233] In one embodiment, the transaction 303 includes the recipient's address 324 (e.g., a hash value based on the receiver's public key), the Blockchain token 309 (i.e., a patient ID 328 and personally identifiable information such as Social Security 326), past medical institution relationship information 331 (if any), and optional other information 310. The transaction 323 is digitally signed by the patient who is the sender's private key to create a digital signature 332 for verifying the sender's identity to the network nodes. The network nodes decrypt the digital signature 332, via the sender's previously exchanged public key, and compare the unencrypted information to the transaction 323. If they match, the sender's authenticity is verified and, after a proper chain of ownership is verified via the ledgers (as explained above), the receiver is recorded in the ledgers as the new Blockchain token 329 authorized owner of the medical information. Off-chain storage warehouses can be used for containing the patient's medical history so that the current owner (or all prior owners) can access the patient medical information for treatment. Further, the information can be segmented according to need. This way, if a medication such as cannabis that requires the patient to be an adult, the system can be queried only to the information needed (such as is this patient an adult) and the system can respond only as to the query and there is no need to send other question (in the adult age example, the system replies only adult or not and does not send the birthday to the inquiring system).

[0234] In another embodiment, the system includes two look up tables, a global registration look up table (GRLT) where all participants (medical institutions and patients) are recorded with name or identity string, blockchain address for the smart contract, and Patient-Provider lookup table (PPLT). This is maintained by a trusted host authority such as a government health authority or a government payor authority. One embodiment maps participant identification strings to their blockchain address or Ethereum address identity (equivalent to a public key). Terms in the smart contract can regulate registering new identities or changing the mapping of existing ones. Identity registration can thus be restricted only to certified institutions. The PPLT maps identity strings to an address on the blockchain.

[0235] Patients can poll their PPLT and be notified whenever a new relationship is suggested or an update is available. Patients can accept, reject or delete relationships, deciding which records in their history they acknowledge. The accepting or rejecting relationships is done only by the patients. To avoid notification spamming from malicious participants, only trusted providers can update the status variable. Other contract terms or rules can specify additional verifications to confirm proper actor behavior.

[0236] When Provider 1 adds a record for a new patient, using the GRLT on the blockchain, the patient's identifying information is first resolved to their matching Ethereum address and the corresponding PPLT is located. Provider 1 uses a cached GRLT table to look up any existing records of the patient in the PPLT. For all matching PPLTs, Provider 1 broadcasts a smart contract requesting patient information to all matching PPLT entries. If the cache did not produce a result for the patient identity string or blockchain address, Provider 1 can send a broadcast requesting institutions who handles the patient identity string or the blockchain address to all providers. Eventually, Provider 2 responds with its addresses. Provider 2 may insert an entry for Provider 1 into

its address resolution table for future use. Provider 1 caches the response information in its table and can now pull information from Provider 2 and/or supplement the information known to Provider 2 with hashed addresses to storage areas controlled by Provider 1.

[0237] Next, the provider uploads a new PPR to the blockchain, indicating their stewardship of the data owned by the patient's Ethereum address. The provider node then crafts a query to reference this data and updates the PPR accordingly. Finally, the node sends a transaction which links the new PPR to the patient's PPLT, allowing the patient node to later locate it on the blockchain.

[0238] A Database Gatekeeper provides an off-chain, access interface to the trusted provider node's local database, governed by permissions stored on the blockchain. The Gatekeeper runs a server listening to query requests from clients on the network. A request contains a query string, as well as a reference to the blockchain PPR that warrants permissions to run it. The request is cryptographically signed by the issuer, allowing the gatekeeper to confirm identities. Once the issuer's signature is certified, the gatekeeper checks the blockchain contracts to verify if the address issuing the request is allowed access to the query. If the address checks out, it runs the query on the node's local database and returns the result over to the client.

[0239] A patient selects data to share and updates the corresponding PPR with the third-party address and query string. If necessary, the patient's node can resolve the third party address using the GRLT on the blockchain. Then, the patient node links their existing PPR with the care provider to the third-party's Summary Contract. The third party is automatically notified of new permissions, and can follow the link to discover all information needed for retrieval. The provider's Database Gatekeeper will permit access to such a request, corroborating that it was issued by the patient on the PPR they share.

[0240] In one embodiment that handles persons without previous blockchain history, admitting procedures are performed where the person's personal data is recorded and entered into the blockchain system. This data may include: name, address, home and work telephone number, date of birth, place of employment, occupation, emergency contact information, insurance coverage, reason for hospitalization, allergies to medications or foods, and religious preference, including whether or not one wishes a clergy member to visit, among others. Additional information may include past hospitalizations and surgeries, advance directives such as a living will and a durable power to attorney. During the time spent in admitting, a plastic bracelet will be placed on the person's wrist with their name, age, date of birth, room number, and blockchain medical record reference on it.

[0241] The above system can be used to connect the blockchain with different EHR systems at each point of care setting. Any time a patient is registered into a point of care setting, the EHR system sends a message to the GRLT to identify the patient if possible. In our example, Patient A is in registration at a particular hospital. The PPLT is used to identify Patient A as belonging to a particular plan. The smart contracts in the blockchain automatically updates Patient A's care plan. The blockchain adds a recommendation to put Patient A by looking at the complete history of treatments by all providers and optimizes treat. For example, the system can recommend the patient be enrolled in a weight loss program after noticing that the patient was

treated for sedentary lifestyle, had history of hypertension, and the family history indicates a potential heart problem. The blockchain data can be used for predictive analytics, allowing patients to learn from their family histories, past care and conditions to better prepare for healthcare needs in the future. Machine learning and data analysis layers can be added to repositories of healthcare data to enable a true “learning health system” can support an additional analytics layer for disease surveillance and epidemiological monitoring, physician alerts if patients repeatedly fill and abuse prescription access.

[0242] In one embodiment, an IoT medical device captures patient data in the hospital and automatically communicates data to a hospital database that can be shared with other institutions or doctors. First, the patient ID and blockchain address is retrieved from the patient's wallet and the medical device attaches the blockchain address in a field, along with other fields receiving patient data. Patient data is then stored in a hospital database marked with the blockchain address and annotated by a medical professional with interpretative notes. The notes are affiliated with the medical professional's blockchain address and the PPR block chain address. A professional can also set up the contract terms defining a workflow. For example, if the device is a blood pressure device, the smart contract can have terms that specify dietary restrictions if the patient is diabetic and the blood pressure is borderline and food dispensing machines only show items with low salt and low calorie, for example.

[0243] The transaction data may consist of a Colored Coin implementation (described in more detail at https://en.bitcoin.it/wiki/Colored_Coins which is incorporated herein by reference), based on Open Assets (described in more detail at <https://github.com/OpenAssets/open-assets-protocol/blob/master/specification.mediawiki> which is incorporated herein by reference), using on the OP_RETURN operator. Metadata is linked from the Blockchain and stored on the web, dereferenced by resource identifiers and distributed on public torrent files. The colored coin specification provides a method for decentralized management of digital assets and smart contracts (described in more detail at <https://github.com/ethereum/wiki/wiki/White-Paper> which is incorporated herein by reference.) For our purposes the smart contract is defined as an event-driven computer program, with state, that runs on a blockchain and can manipulate assets on the blockchain. So a smart contract is implemented in the blockchain scripting language in order to enforce (validate inputs) the terms (script code) of the contract.

Patient Behavior and Risk Pool Rated Health Plans

[0244] With the advent of personal health trackers, new health plans are rewarding consumers for taking an active part in their wellness. The system facilitates open distribution of the consumers wellness data and protect it as PHR must be, and therefore prevent lock-in of consumers, providers and payers to a particular device technology or health plan. In particular, since PHR data is managed on the blockchain a consumer and/or company can grant access to a payer to this data such that the payer can perform group analysis of an individual or an entire company's employee base including individual wellness data and generate a risk score of the individual and/or organization. Having this information, payers can then bid on insurance plans tailored for the specific organization. Enrollment then, also being managed on the blockchain, can become a real-time arbitrage process. The pseudo code for the smart contract to implement a patient behavior based health plan is as follows.

- [0245] store mobile fitness data
- [0246] store consumer data in keys with phr_info, claim_info, enrollment_info
- [0247] for each consumer:
- [0248] add up all calculated risk for the consumer
- [0249] determine risk score based on mobile fitness data
- [0250] update health plan cost based on patient behavior
- [0251] Patient and Provider Data Sharing
- [0252] A patient's Health BlockChain wallet stores all assets, which in turn store reference ids to the actual data, whether clinical documents in HL7 or FHIR format, wellness metrics of activity and sleep patterns, or claims and enrollment information. These assets and control of grants of access to them is afforded to the patient alone. A participating provider can be given full or partial access to the data instantaneously and automatically via enforceable restrictions on smart contracts.
- [0253] Utilizing the Health BlockChain the access to a patient's PHR can be granted as part of scheduling an appointment, during a referral transaction or upon arrival for the visit. And, access can just as easily be removed, all under control of the patient.
- [0254] Upon arrival at the doctor's office, an application automatically logs into a trusted provider's wireless network. The app is configured to automatically notify the provider's office of arrival and grant access to the patient's PHR. At this point the attending physician will have access to the patient's entire health history. The pseudo code for the smart contract to implement a patient and provider data sharing is as follows.
- [0255] Patient download apps and provide login credentials and logs into the provider wireless network
- [0256] Patient verifies that the provider wireless network belongs to a patient trusted provider list
- [0257] Upon entering provider premise, system automatically logs in and grants access to provider
- [0258] Patient check in data is automatically communicated with provider system to provide PHR
- [0259] Provider system synchronizes files and obtain new updates to the patient PHR and flags changes to provider.

Patient Data Sharing

[0260] Patient's PHR data is valuable information for their personal health profile in order to provide Providers (Physicians) the necessary information for optimal health care delivery. In addition this clinical data is also valuable in an aggregate scenario of clinical studies where this information is analyzed for diagnosis, treatment and outcome. Currently this information is difficult to obtain due to the siloed storage of the information and the difficulty on obtaining patient permissions.

[0261] Given a patient Health BlockChain wallet that stores all assets as reference ids to the actual data. These assets can be included in an automated smart contract for clinical study participation or any other data sharing agreement allowed by the patient. The assets can be shared as an instance share by adding to the document a randomized identifier or nonce, similar to a one-time use watermark or serial number, a unique asset (derived from the original source) is then generated for a particular access request and included in a smart contract as an input for a particular

request for the patient's health record information. A patient can specify their acceptable terms to the smart contract regarding payment for access to PHR, timeframes for acceptable access, type of PHR data to share, length of history willing to be shared, de-identification thresholds or preferences, specific attributes of the consumer of the data regarding trusted attributes such as reputation, affiliation, purpose, or any other constraints required by the patient. Attributes of the patient's data are also advertised and summarized as properties of the smart contract regarding the type of diagnosis and treatments available. Once the patient has advertised their willingness to share data under certain conditions specified by the smart contract it can automatically be satisfied by any consumer satisfying the terms of the patient and their relevance to the type of PHR needed resulting in automated, efficient and distributed means for clinical studies to consume relevant PHR for analysis. This process provides an automated execution over the Health BlockChain for any desired time period that will terminate at an acceptable statistical outcome of the required attained significance level or financial limit. The pseudo code for the smart contract to implement automated patient data sharing is as follows.

[0262] Patient download apps and provide login credential and logs into the clinical trial provider wireless network

[0263] Patient verifies that the provider wireless network belongs to a patient trusted provider list

[0264] Upon entering provider premise, system automatically logs in and grants access to provider

[0265] Patient check in data is automatically communicated with provider system to provide clinical trial data

[0266] In one embodiment, a blockchain entry is added for each touchpoint of the medication as it goes through the supply chain from manufacturing where the prescription package serialized numerical identification (SNI) is sent to wholesalers who scan and record the SNI and location and then to distributors, repackagers, and pharmacies, where the SNI/location data is recorded at each touchpoint and put on the blockchain. The medication can be scanned individually, or alternatively can be scanned in bulk. Further, for bulk shipments with temperature and shock sensors for the bulk package, temperature/shock data is captured with the shipment or storage of the medication.

[0267] A smart contract assesses against product supply chain rule and can cause automated acceptance or rejection as the medication goes through each supply chain touchpoint. The process includes identifying a prescription drugs by query of a database system authorized to track and trace prescription drugs or similar means for the purpose of monitoring the movements and sale of pharmaceutical products through a supply chain; a.k.a. e-pedigree trail; serialized numerical identification (SNI), stock keeping units (SKU), point of sale system (POS), systems etc. in order to compare the information; e.g. drug name, manufacturer, etc. to the drug identified by the track and trace system and to ensure that it is the same drug and manufacturer of origin. The process can verify authenticity and check pedigree which can be conducted at any point along the prescription drug supply chain; e.g. wholesaler, distributor, doctor's office, pharmacy. The most optimal point for execution of this process would be where regulatory authorities view the greatest vulnerability to the supply chain's integrity. For

example, this examination process could occur in pharmacy operations prior to containerization and distribution to the pharmacy for dispensing to patients.

[0268] An authenticated prescription drug with verified drug pedigree trail can be used to render an informational object, which for the purpose of illustration will be represented but not be limited to a unique mark; e.g. QR Code, Barcode, Watermark, Stealth Dots, Seal or 2 Dimensional graphical symbol, hereinafter called a certificate, seal, or mark. An exemplary embodiment for use of said certificate, mark, or seal can be used by authorized entities as a warrant of the prescription drug's authenticity and pedigree. For example, when this seal is appended to a prescription vial presented to a patient by a licensed pharmacy, it would represent the prescription drug has gone through an authentication and logistics validation process authorized by a regulatory agency (s); e.g. HHS, FDA, NABP, VIPP, etc. An exemplary embodiment for use of said certificate, mark or seal would be analogous to that of the functioning features, marks, seals, and distinguishing characteristics that currently authenticate paper money and further make it difficult to counterfeit. Furthermore, authorized agents utilizing the certificate process would be analogous to banks participating in the FDIC program.

[0269] A user; e.g. patient equipped with the appropriate application on a portable or handheld device can scan the certificate, mark or seal and receive an audible and visible confirmation of the prescription drug's name, and manufacturer. This will constitute a confirmation of the authenticity of the dispensed prescription drug. Extensible use of the certificate, mark, or seal will include but not be limited to; gaining access to website (s) where additional information or interactive functions can be performed; e.g. audible narration of the drug's characteristics and physical property descriptions, dosing, information, and publications, etc. A user; e.g. patient equipped with the appropriate application on a portable or handheld device can scan the certificate, mark, or seal and be provided with notifications regarding; e.g. immediate recall of the medication, adverse events, new formulations, critical warnings of an immediate and emergency nature made by prescription drug regulatory authorities and, or their agents. A user; e.g. patient equipped with a portable or handheld device with the appropriate application software can use the portable and, or handheld device to store prescription drug information in a secure, non-editable format on their device for personal use; e.g. MD's Office Visits, Records Management, Future Authentications, Emergency use by first responders etc. A user; e.g. patient equipped with the appropriate application on a portable or handheld device can scan the drug via an optical scan, picture capture, spectroscopy or other means of identifying its physical properties and characteristics; e.g. spectral signature, size, shape, color, texture, opacity, etc and use this data to identify the prescription drug's name, and manufacturer. A user; e.g. patient equipped with the appropriate application on a portable or handheld device and having the certification system can receive updated information (as a subscriber in a client/server relationship) on a continuing or as needed ad hoc basis (as permitted) about notifications made by prescription drug regulatory authorities regarding; e.g. immediate recall of medications, adverse events, new formulations and critical warnings of an immediate and emergency nature. A user; e.g. patient, subscriber to the certificate system equipped with the appropriate application

on a portable or handheld device will be notified by audible and visible warnings of potential adverse affects between drug combinations stored in their device's memory of previously "Certified Drugs." A user; e.g. patient subscriber to the certification system equipped with the appropriate application on a portable or handheld device will receive notification of potential adverse affects from drug combinations, as reported and published by medical professionals in documents and databases reported to; e.g. Drug Enforcement Administration (DEA), Health and Human Services, (HHS) Food and Drug Administration, (FDA) National Library of Medicines, (NLM) and their agents; e.g., Daily Med, Pillbox, RX Scan, PDR, etc.

[0270] 1. A method for prescription drug authentication by receiving a certificate representing manufacturing origin and distribution touchpoints of a prescription drug on a blockchain.

[0271] 2. A method of claim 1, comprising retrieving active pharmaceutical ingredients (API) and inactive pharmaceutical ingredients (IPI) from the blockchain.

[0272] 3. A method of claim 2, comprising authenticating the drug after comparing the API and IPI with data from Drug Enforcement Administration (DEA) Health and Human Services, (HHS) Food and Drug Administration, (FDA) National Library of Medicines, (NLM) etc. for the purpose of identifying the prescription drug(s) and manufacture name indicated by those ingredients.

[0273] 4. A method of claim 1, comprising tracing the drug through a supply chain from manufacturer to retailer, dispenser with Pedigree Trail, Serialized Numerical Identification (SNI), Stock Keeping Units (SKU), Point of Sale System (POS) E-Pedigree Systems.

[0274] 5. A method of claim 1, comprising generating a certificate, seal, mark and computer scannable symbol such as 2 or 3 dimensional symbol; e.g. QR Code, Bar Code, Watermark, Stealth Dots, etc. RECOGNITION OF EXERCISE PATTERN AND TRACKING OF CALORIE CONSUMPTION

[0275] The learning system can be used to detect and monitor user activities as detected by the ITE sensors. FIG. 8A illustrates the positions of a ski 126' and skier 128' during a lofting maneuver on the slope 132'. The ski 126' and skier 128' speed down the slope 132' and launch into the air 136 at position "a," and later land at position "b" in accord with the well-known Newtonian laws of physics. With an airtime sensor, described above, the unit 10 calculates and stores the total airtime that the ski 126' (and hence the skier 128') experiences between the positions "a" and "b" so that the skier 128' can access and assess the "air" time information. Airtime sensors such as the sensor 14 may be constructed with known components. Preferably, the sensor 14 incorporates either an accelerometer or a microphone. Alternatively, the sensor 14 may be constructed as a mechanical switch that detects the presence and absence of weight onto the switch. Other airtime sensors 14 will become apparent in the description which follows. The accelerometer senses vibration-particularly the vibration of a vehicle such as a ski or mountain bike-moving along a surface, e.g., a ski slope or mountain bike trail. This voltage output provides an acceleration spectrum over time; and information about airtime can be ascertained by performing calculations on that spectrum. Based on the information, the system can reconstruct

the movement path, the height, the speed, among others and such movement data is used to identify the exercise pattern. For example, the skier may be interested in practicing mogul runs, and the system can identify foot movement and speed and height information and present the information post exercises as feedback. Alternatively, the system can make live recommendations to improve performance to the athlete.

[0276] FIG. 8B illustrates a sensing unit 10" mounted onto a mountain bike 138. FIG. 8B also shows the mountain bike 138 in various positions during movement along a mountain bike race course 140 (for illustrative purposes, the bike 138 is shown without a rider). At one location "c" on the race course 140, the bike 138 hits a dirt mound 142 and catapults into the air 144. The bike 138 thereafter lands at location "d". As above, with speed and airtime sensors, the unit 10 provides information to a rider of the bike 138 about the speed attained during the ride around the race course 140; as well as information about the airtime between location "c" and "d". In this case, the system can recommend a cadence to be reached by the rider, strengthen of abdominals, back and arms, for example.

[0277] For golf exercise, It is beneficial to require the golfer to swing the golf club a plurality of times at each swing position to account for variations in each swing. The swing position at which the golf club is swung can be determined by analysis of the measured acceleration provided by the accelerometer, e.g., the time at which the acceleration changes. Data obtained during the training stage may be entered into a virtual table of swing positions and estimated carrying distances for a plurality of different swing positions and a plurality of different swings. A sample format for such a table is as follows, and includes the averaged carrying distance for each of four different swing positions. The swing analyzer provides a golfer with an excellent estimation of the carrying distance of a golf ball for a golf club swing at a specific swing position because it has been trained on actual swings by the golfer of the same club and conversion of information about these swings into estimated carrying distances. The golfer can improve their golf game since they can better select a club to use to hit a golf club for different situations during a round of golf. Also, the swing pattern is used to identify each club path responsible for the curve of any shot and this information is used to improve the golfer. The direction of the club path relative to the target, out-to-in (fade pattern) or in-to-out (draw pattern), is what I refer to as a players swing pattern. Players that swing from in-to-out will tend to hit draws and players that swing from out-to-in will tend to hit fades. Where the ball is struck on the face of the driver (strike point) can drastically alter the effect of a players swing pattern on ball flight. Thus, the camera detects where the ball is struck, and a computer physics model of ball behavior is presented to the golfer to improve the score. Shots struck off the heel will tend to fade more or draw less and shots struck off the toe will tend to draw more or fade less. Thus, camera images of the shots struck of heel or toe can also be used to provide pattern recognition/prediction and for training purposes.

[0278] For tennis, examples of motions determined for improvement are detailed next. The system can detect if the continental grip is achieved. Throwing Action pattern is also detected, as the tennis serve is an upwards throwing action that would deliver the ball into the air if it were a baseball pitch. Ball Toss improvements can be determined when the

player lines the straight arm up with the net post and release the ball when your hand reaches eye level. The system checks the forward direction so the player can drive weight (and built up momentum) forward into the ball and into the direction of the serve.

[0279] The sensors can work with a soccer training module with kinematics of ball control, dribbling, passing, crossing, shooting, heading, volleying, taking throw-ins, penalties, corner kicks and free kicks, tackling, marking, juggling, receiving, shielding, clearing, and goalkeeping. The sensors can work with a basketball training module with kinematics of crossover dribble, behind back, pull back dribble, low dribble, basic dribble, between legs dribble, Overhead Pass, Chest Pass, Push Pass, Baseball Pass, Off-the-Dribble Pass, Bounce Pass, Jump Shot, Dunk, Free throw, Layup, Three-Point Shot, Hook Shot.

[0280] The sensors can work with a baseball training module with kinematics of Hitting, Bunting, Base Running and Stealing, Sliding, Throwing, Fielding Ground Balls, Fielding Fly Balls, Double Plays and Relays, Pitching and Catching, Changing Speeds, Holding Runners, Pitching and Pitcher Fielding Plays, Catching and Catcher Fielding Plays.

[0281] For weight training, the sensor can be in gloves as detailed above, or can be embedded inside the weight itself, or can be in a smart watch, for example. The user would enter an app indicating that the user is doing weight exercises and the weight is identified as a dumbbell, a curl bar, and a bar bell. Based on the arm or leg motion, the system automatically detects the type of weight exercise being done. In one embodiment, with motion patterns captured by glove and sock sensors, the system can automatically detect specific exercises.

[0282] In one implementation, an HMM is used to track weightlifting motor skills or sport enthusiast movement patterns. Human movement involves a periodic motion of the legs. Regular walking involves the coordination of motion at the hip, knee and ankle, which consist of complex joints. The muscular groups attached at various locations along the skeletal structure often have multiple functions. The majority of energy expended during walking is for vertical motion of the body. When a body is in contact with the ground, the downward force due to gravity is reflected back to the body as a reaction to the force. When a person stands still, this ground reaction force is equal to the person's weight multiplied by gravitational acceleration. Forces can act in other directions. For example, when we walk, we also produce friction forces on the ground. When the foot hits the ground at a heel strike, the friction between the heel and the ground causes a friction force in the horizontal plane to act backwards against the foot. This force therefore causes a breaking action on the body and slows it down. Not only do people accelerate and brake while walking, they also climb and dive. Since reaction force is mass times acceleration, any such acceleration of the body will be reflected in a reaction when at least one foot is on the ground. An upwards acceleration will be reflected in an increase in the vertical load recorded, while a downwards acceleration reduce the effective body weight. Sensors can be placed on the four branches of the links connect to the root node (torso) with the connected joint, left shoulder (LS), right shoulder (RS), left hip (LH), and right hip (RH). Furthermore, the left elbow (LE), right elbow (RE), left knee (LK), and right knee (RK) connect the upper and the lower extremities. The wireless monitoring devices can also be

placed on upper back body near the neck, mid back near the waist, and at the front of the right leg near the ankle, among others.

[0283] The sequence of human motions can be classified into several groups of similar postures and represented by mathematical models called model-states. A model-state contains the extracted features of body signatures and other associated characteristics of body signatures. Moreover, a posture graph is used to depict the inter-relationships among all the model-states, defined as PG(ND,LK), where ND is a finite set of nodes and LK is a set of directional connections between every two nodes. The directional connection links are called posture links. Each node represents one model-state, and each link indicates a transition between two model-states. In the posture graph, each node may have posture links pointing to itself or the other nodes.

[0284] In the pre-processing phase, the system obtains the human body profile and the body signatures to produce feature vectors. In the model construction phase, the system generate a posture graph, examine features from body signatures to construct the model parameters of HMM, and analyze human body contours to generate the model parameters of ASMs. In the motion analysis phase, the system uses features extracted from the body signature sequence and then applies the pre-trained HMM to find the posture transition path, which can be used to recognize the motion type. Then, a motion characteristic curve generation procedure computes the motion parameters and produces the motion characteristic curves. These motion parameters and curves are stored over time, and if differences for the motion parameters and curves over time is detected, the system then runs the sport enthusiast through additional tests to confirm the detected motion.

[0285] In one exemplary process for determining exercise in the left or right half of the body, the process compares historical left shoulder (LS) strength against current LS strength (3200). The process also compares historical right shoulder (RS) strength against current RS strength (3202). The process can compare historical left hip (LH) strength against current LH strength (3204). The process can also compare historical right hip (RH) strength against current RH strength (3206). If the variance between historical and current strength exceeds threshold, the process generates warnings (3208). Furthermore, similar comparisons can be made for sensors attached to the left elbow (LE), right elbow (RE), left knee (LK), and right knee (RK) connect the upper and the lower extremities, among others.

[0286] In one embodiment, the accelerometers distinguish between lying down and each upright position of sitting and standing based on the continuous output of the 3D accelerometer. The system can detect (a) extended time in a single position; (b) extended time sitting in a slouching posture (kyphosis) as opposed to sitting in an erect posture (lordosis); and (c) repetitive stressful movements, such as may be found on some manufacturing lines, while typing for an extended period of time without proper wrist support, or while working all day at a weight lifting exercise, among others. In one alternative embodiment, angular position sensors, one on each side of the hip joint, can be used to distinguish lying down, sitting, and standing positions. In another embodiment, the system repeatedly records position and/or posture data over time. In one embodiment, magnetometers can be attached to a thigh and the torso to provide absolute rotational position about an axis coincident with

Earth's gravity vector (compass heading, or yaw). In another embodiment, the rotational position can be determined through the in-door positioning system as discussed above.

[0287] The system can identify illnesses and prevent over-exertion leading to illnesses such as a stroke. Depending on the severity of the stroke, sport enthusiasts can experience a loss of consciousness, cognitive deficits, speech dysfunction, limb weakness, hemiplegia, vertigo, diplopia, lower cranial nerve dysfunction, gaze deviation, ataxia, hemianopia, and aphasia, among others. Four classic syndromes that are characteristically caused by lacunar-type stroke are: pure motor hemiparesis, pure sensory syndrome, atactic hemiparesis syndrome, and clumsy-hand dysarthria syndrome. Sport enthusiasts with pure motor hemiparesis present with face, arm, and leg weakness. This condition usually affects the extremities equally, but in some cases it affects one extremity more than the other. The most common stroke location in affected sport enthusiasts is the posterior limb of the internal capsule, which carries the descending corticospinal and corticobulbar fibers. Other stroke locations include the pons, midbrain, and medulla. Pure sensory syndrome is characterized by hemibody sensory symptoms that involve the face, arm, leg, and trunk. It is usually the result of an infarct in the thalamus. Atactic hemiparesis syndrome features a combination of cerebellar and motor symptoms on the same side of the body. The leg is typically more affected than the arm. This syndrome can occur as a result of a stroke in the pons, the internal capsule, or the midbrain, or in the anterior cerebral artery distribution. Sport enthusiasts with clumsy-hand dysarthria syndrome experience unilateral hand weakness and dysarthria. The dysarthria is often severe, whereas the hand involvement is more subtle, and sport enthusiasts may describe their hand movements as "awkward." This syndrome is usually caused by an infarct in the pons. Different patterns of signs can provide clues as to both the location and the mechanism of a particular stroke. The system can detect symptoms suggestive of a brainstem stroke include vertigo, diplopia, bilateral abnormalities, lower cranial nerve dysfunction, gaze deviation (toward the side of weakness), and ataxia. Indications of higher cortical dysfunction-such as neglect, hemianopsia, aphasia, and gaze preference (opposite the side of weakness)-suggest hemispheric dysfunction with involvement of a superficial territory from an atherothrombotic or embolic occlusion of a mainstem vessel or peripheral branch.

[0288] In one embodiment, the ear-worn (earable) device operates stand alone with a touch sensitive input that the user can tap or slide finger across a surface to select an action. The touch sensitive input can coexist with a microphone array that captures sound for processing. The microphone array in one ear can collaborate with the other unit to provide bi-aural/stereo signal enhancement, and AI is used to perform noise removal and signal clean up.

[0289] In another embodiment that collaborates with other units such as eye-glasses or wearable devices with cameras, the ear-worn device can perform distributed visual and sound analysis with a camera on a compact, lightweight wearable device that can be attached to the eye, clothing or objects, or simply can be on a mobile phone. The camera on the wearable device can record moments and transactions with minimal interaction.

[0290] The captured data is processed in the cloud, allowing for editing and formatting into different presentation formats. The ear-worn device and/or the wearable device

can include touch gestures for control, automatic power management, and various sensors for environmental and biometric data collection. Additionally, the wearable device can have a light beam or projector for replaying moments, and can be controlled through gestures and voice commands. The device simplifies capturing life events and financial transactions, enhancing user experiences through contextual awareness and multimedia capabilities.

[0291] The camera can have a wide field of view and optical image stabilizer. It enables users to capture multimedia data like video, audio, and depth data. This data is uploaded to a cloud platform where it can be processed, edited, and formatted into various presentation formats based on user preferences or machine learning algorithms. The cloud platform is designed as a scalable distributed computing environment with real-time streaming capabilities.

[0292] The wearable multimedia device includes a depth sensor, such as LiDAR or Time of Flight (TOF), that generates point cloud data to provide 3D surface mapping of objects. This 3D data can be processed using augmented reality (AR) and virtual reality (VR) applications in the device's application ecosystem.

[0293] The wearable device can also include various environmental sensors, such as ambient light, magnetometer, pressure, and voice activity detectors. This sensor data is included in the context data to provide additional information that can enrich the content presentation.

[0294] Furthermore, the ear-worn device or wearable device can incorporate biometric sensors like heart rate and fingerprint scanners. This biometric data can be used to document transactions or indicate the user's emotional state during a captured moment, such as elevated heart rate signaling excitement or fear.

[0295] Users can initiate data capture sessions on the device through touch gestures or voice commands. The device incorporates sensors to detect when it is not being worn and automatically powers down to conserve energy. It also features technologies like photovoltaic surfaces for extended battery life and inductive charging options. Data captured by the device is encrypted, compressed, and stored securely in an online database associated with user accounts.

[0296] The wearable and/or earable device integrates advanced technologies such as point cloud data for 3D mapping, GNSS receivers for location tracking, inertial sensors for orientation detection, environmental sensors for ambient data collection, and biometric sensors for health monitoring. It also includes communication technologies like Bluetooth and WiFi for connectivity with other devices. The device can be controlled through touch gestures, voice commands, or head gestures.

[0297] In addition to personal events, the system streamlines the documentation of financial transactions by generating detailed data including images, audio recordings, location information, and transaction details. This data is sent to the cloud platform for storage or processing by financial applications. The platform also supports third-party applications for various purposes like live broadcasting, senior monitoring, memory recall, and personal guidance.

[0298] One embodiment involves determining user requests in the form of speech within the context data, converting speech into text, and identifying real-world objects using this text. This functionality could be applied in scenarios where users verbally request information about

objects around them, enabling the device to recognize and provide details about those objects based on the converted speech.

[0299] AI in the form of a large language model (LLM) can assist the user. The earable device can use the LLM's natural language understanding capabilities to provide the user with subtle, voice-based navigation instructions without drawing attention. For example, the LLM could detect the user's location and provide directions like "Turn left at the next intersection" or "Your destination is 50 meters ahead on the right" directly through the earbuds. This would allow the user to navigate without constantly looking at a screen or map, keeping their focus on their surroundings. The earbuds could also leverage the LLM's awareness of the user's context to provide relevant information, like pointing out landmarks, warning about obstacles, or suggesting nearby points of interest.

[0300] An LLM integrated with an earable device can assist the user in having more natural and personalized small talk conversations. By accessing the user's calendar, contacts, and other personal data, the LLM could provide subtle prompts or suggestions to help the user warm up a conversation. For example, it could remind the user of a shared interest or experience with the other person, or offer a personalized icebreaker based on information it has about the conversation partner. The LLM could also analyze the conversation in real-time and offer the user discreet feedback or prompts to keep the discussion flowing smoothly, without the other person noticing. This kind of personalized, context-aware interaction through an earable device would allow the user to have more meaningful and effortless conversations, without the LLM's presence being obvious.

[0301] One embodiment includes processing audio data to identify user requests, converting speech to text, sending the text to a transportation service processor, receiving transport status and vehicle descriptions, and relaying this information back to the wearable multimedia device or another device. This feature could be utilized for tasks like requesting transportation services verbally through the device, receiving real-time updates on transport availability, and getting details about vehicles for hire.

[0302] In one scenario, the method involves utilizing VR or AR applications to generate content using digital images or depth data and sending this content to the wearable multimedia device or other devices. Practical applications could include immersive experiences where users can view augmented reality content related to captured moments or locations.

[0303] One embodiment employs artificial intelligence applications to determine user preferences from past requests and process context data accordingly. This functionality could enhance user experiences by customizing how data is processed and presented based on individual preferences learned over time.

[0304] One embodiment uses localization applications to determine the location of the wearable multimedia device based on digital images or depth data and relay this information back to the device. It could be beneficial for location-based services where users need accurate positioning information related to captured moments or activities.

[0305] One scenario involves financial applications creating financial records based on transaction data, biometric data, and location information within the context data. This feature could streamline financial record-keeping by auto-

matically generating detailed records of transactions with associated biometric and location data for user reference.

[0306] One embodiment employs environmental applications to generate content related to the operating environment of the device based on environmental sensor data and send this content back to the device. This functionality could provide users with contextual information about their surroundings captured through environmental sensors.

[0307] In another embodiment, a video editing application edits video and audio based on user requests or preferences for specific film styles and sends the edited content back to the wearable multimedia device or other devices. Users could customize video and audio content captured by the device according to their preferences for different film styles before viewing or sharing it.

[0308] A method for operating an earable device, comprising:

[0309] Providing sensor data to a Large Language Model (LLM).

[0310] Monitoring user interactions and conversations using audio sensors in the earable device.

[0311] Analyzing the user's speech, tone, and conversational patterns using the LLM.

[0312] Accessing the user's personal data, including calendar, contacts, and preferences, to understand the context of the conversation.

[0313] The method of claim 1, further comprising:

[0314] Providing subtle prompts or suggestions to the user to help initiate or guide the conversation based on the analyzed context.

[0315] The method of claim 1, wherein the LLM is configured to:

[0316] Detect changes in the user's emotional state or engagement level during the conversation.

[0317] Offer discreet feedback or prompts to the user to maintain a smooth and natural flow of the conversation.

[0318] The method of claim 1, wherein the earable device is designed to integrate the LLM's presence seamlessly, ensuring the user's conversation partner is unaware of the device's involvement.

[0319] The method of claim 1, further comprising:

[0320] Continuously learning and adapting the LLM's conversational strategies based on the user's feedback and preferences over time.

[0321] The method of claim 1, wherein the earable device is capable of:

[0322] Processing audio data to identify user requests, such as for transportation services.

[0323] Converting speech to text and relaying the information to relevant service providers.

[0324] Receiving and conveying real-time updates on the status of the requested services.

[0325] The method of claim 1, further comprising:

[0326] Utilizing VR or AR applications on the earable device to generate immersive digital content based on captured images or depth data.

[0327] The method of claim 1, wherein the earable device employs:

[0328] Artificial intelligence applications to customize data processing and presentation based on learned user preferences.

- [0329] The method of claim 1, further comprising:
- [0330] Leveraging localization applications to determine the user's location based on sensor data and provide location-based services.
- [0331] The method of claim 1, wherein the earable device integrates:
- [0332] Financial applications to automatically generate detailed records of transactions, including associated biometric and location data.
- [0333] The method of claim 1, further comprising:
- [0334] Utilizing environmental applications to provide contextual information about the user's surroundings based on sensor data.
- [0335] An earable device, comprising
- [0336] Sensors to detect biometrics.
- [0337] A Large Language Model (LLM) for facilitating natural and personalized small talk conversations, wherein the LLM can perform:
- [0338] Access to the user's calendar, contacts, and personal data.
- [0339] Providing subtle prompts or suggestions to assist the user in initiating conversations.
- [0340] Analyzing real-time conversations and offering discreet feedback or prompts to enhance conversational flow.
- [0341] The earable device of claim 1, wherein the LLM utilizes information from the user's calendar and contacts to suggest relevant topics or icebreakers during conversations.
- [0342] The earable device of claim 1, further comprising a context-aware interaction system that tailors prompts and feedback based on the conversation partner's responses and cues.
- [0343] The earable device of claim 1, wherein the LLM operates in a manner that seamlessly integrates into conversations, enhancing user interactions without overtly revealing its presence.
- [0344] The earable device of claim 1, enabling users to engage in more meaningful and effortless conversations through personalized prompts and feedback provided by the integrated LLM.
- [0345] In another aspect, a method for processing audio data on an earable device, comprising:
- [0346] Identifying user requests from audio inputs.
- [0347] Converting speech to text.
- [0348] Sending text to a transportation service processor.
- [0349] Receiving transport status and vehicle descriptions.
- [0350] Relaying transportation information back to the wearable multimedia device or another connected device.
- [0351] An embodiment utilizing VR or AR applications on the earable device to generate digital content from images or depth data, providing immersive experiences for users.
- [0352] A functionality employing artificial intelligence applications on the earable device to customize data processing and presentation based on user preferences learned over time.
- [0353] An embodiment using localization applications on the earable device to determine accurate positioning information based on digital images or depth data, enhancing location-based services for users.
- [0354] A scenario involving financial applications on the earable device to automatically generate detailed financial records from transaction data, biometric data, and location information within context data for user reference.
- [0355] An embodiment utilizing environmental applications on the earable device to provide contextual information about the operating environment captured through environmental sensor data, enhancing user awareness of their surroundings.
- [0356] The LLM could incorporate user biometrics to customize its response in scenarios like detecting fear or threats, falls, or medical emergencies:
- Biometric Monitoring Device with Heart Rate Measurement:
- [0357] The biometric monitoring device allows for heart rate measurement activated by a single user-gesture, providing convenient and user-friendly heart rate monitoring.
- [0358] This device collects various physiological data beyond heart rate, such as body fat, caloric intake, sleep quality, and more, showcasing the potential for comprehensive biometric tracking.
- [0359] Personal LLM Agents can assist users with personal data and services. These agents can be deeply integrated with personal devices and services to reduce repetitive tasks and focus on important matters.
- [0360] While current commercial products integrate LLMs with personal assistants for tasks like drafting documents and enhancing search experiences, deeper functional integration is still in early exploration stages.
- [0361] EEG-Based Biometrics can provide valuable insights into brain activity. Combining EEG data with other biometrics like heart rate can offer a more comprehensive view of the user's physiological state. Incorporating these insights, an LLM integrated with an earable device could utilize real-time biometric data such as heart rate, EEG signals, and potentially other metrics like blood pressure or acceleration to detect anomalies indicative of fear or threats, falls, or medical emergencies. For instance:
- [0362] Fear or Threat Detection: An LLM could analyze changes in heart rate and EEG patterns to detect heightened stress levels or fear responses. If significant anomalies are detected, the LLM could automatically dial for assistance.
- [0363] Fall Detection: Acceleration data combined with sudden changes in heart rate could indicate a fall. Upon detecting such patterns, the LLM could summon help or alert emergency services.
- [0364] Medical Emergency Detection: Abnormalities in biometric data such as a sudden drop in heart rate or irregular EEG patterns could signal a medical emergency. In such cases, the LLM could initiate a call to 911 or notify designated contacts for assistance. By leveraging a combination of biometric sensors and advanced algorithms within the earable device, the LLM can provide personalized responses tailored to the user's physiological state in critical situations.
- Biometric Monitoring Device with Heart Rate Measurement:
- [0365] A biometric monitoring device, comprising:
- [0366] An optical heart rate sensor for measuring heart rate;
- [0367] One or more additional sensors for measuring other physiological data, such as body fat, caloric intake, and sleep quality; and
- [0368] An LLM evaluating the sensor output and recommending user actions.

[0369] The biometric monitoring device of claim 1, wherein the heart rate measurement is activated by a single user-gesture, providing convenient and user-friendly heart rate monitoring.

[0370] The biometric monitoring device of claim 1, further comprising a communication interface for transmitting the collected physiological data to other devices or services.

[0371] A system comprising:

[0372] The biometric monitoring device of claim 1;

[0373] A personal LLM (Large Language Model) agent integrated with the biometric monitoring device;

[0374] Wherein the LLM agent assists the user with personal data and services, and is deeply integrated with the device to reduce repetitive tasks and focus on important matters.

[0375] The system of claim 4, wherein the LLM agent is capable of deeper functional integration beyond current commercial products that integrate LLMs with personal assistants for tasks like drafting documents and enhancing search experiences.

[0376] The system of claim 4, further comprising:

[0377] EEG (Electroencephalography) sensors integrated with the biometric monitoring device;

[0378] Wherein the combination of heart rate, EEG, and other biometric data provides a comprehensive view of the user's physiological state.

[0379] The system of claim 6, wherein the LLM agent is configured to:

[0380] Analyze the real-time biometric data, including heart rate and EEG signals;

[0381] Detect anomalies indicative of fear, threats, falls, or medical emergencies;

[0382] Respond accordingly, such as automatically dialing for assistance or summoning help.

[0383] The system of claim 7, wherein the LLM agent leverages a combination of biometric sensors and advanced algorithms within the earable device to provide personalized responses tailored to the user's physiological state in critical situations.

[0384] The biometric monitoring device of claim 1, wherein the device is configured to track the user's physiological data, including heart rate, heart rate variability, and activity levels, over an extended period of time.

[0385] The biometric monitoring device of claim 1, further comprising a behavior change therapy module that provides personalized guidance and feedback to the user based on the tracked physiological data.

[0386] The biometric monitoring device of claim 2, wherein the behavior change therapy module is configured

to analyze the user's physiological trends over time and adjust the personalized guidance and feedback accordingly.

[0387] The biometric monitoring device of claim 3, wherein the behavior change therapy module is configured to recommend lifestyle modifications, such as adjustments to sleep, exercise, or stress management, based on the user's physiological data and behavior patterns.

[0388] The biometric monitoring device of claim 4, wherein the behavior change therapy module is further configured to monitor the user's progress and adherence to the recommended lifestyle modifications, and provide additional support or adjustments to the therapy plan as needed.

[0389] The biometric monitoring device of claim 5, wherein the behavior change therapy module is integrated with a cloud-based platform or mobile application, allowing for remote monitoring, data synchronization, and collaboration with healthcare professionals.

Example: SuperApp

[0390] LLMs can be integrated with mobile devices to enable seamless interaction with various phone features and apps. This allows the LLM to become a centralized interface for the user, similar to how ChatGPT has become a powerful alternative to traditional web search.

[0391] Some key capabilities of an LLM-powered “super app” include:

[0392] App Control: The LLM can understand natural language commands and translate them into actions to control various apps on the user's phone. This includes making calls, scheduling calendar events, performing web searches, and more.

[0393] Personalization: By learning the user's preferences, habits, and context over time, the LLM can provide highly personalized recommendations and assistance, tailoring the experience to the individual.

[0394] Multimodal Interaction: The LLM can process and generate content in different formats, such as text, images, and even code, allowing the user to interact with the “super app” in a variety of ways. 4

[0395] Cross-App Integration: The LLM can seamlessly integrate and coordinate actions across multiple apps and services, providing a unified experience for the user without the need to switch between different applications.

[0396] Offline Capabilities: By leveraging techniques like model quantization, the LLM can be deployed directly on the user's device, enabling offline functionality and reducing reliance on network connectivity.

Pseudocode

Here's some pseudocode that demonstrates how an LLM-powered “super app” could work:

```
# Initialize LLM and connect to phone's apps/services
llm = initialize_llm()
connect_to_phone_apps()

# Main interaction loop
while True:
    user_input = get_user_input()
    intent = llm.analyze_intent(user_input)
    if intent == "make_call":
        contact_name = llm.extract_contact_name(user_input)
        make_phone_call(contact_name)
    elif intent == "schedule_event":
        event_details = llm.extract_event_details(user_input)
        add_calendar_event(event_details)
    elif intent == "web_search":
```

-continued

```

search_query = llm.extract_search_query(user_input)
perform_web_search(search_query)
elif intent == "app_control":
    app_name, action = llm.extract_app_and_action(user_input)
    control_app(app_name, action)
elif intent == "multimodal_task":
    task_details = llm.extract_multimodal_task(user_input)
    perform_multimodal_task(task_details)
else:
    llm.provide_response(user_input)
time.sleep(1) # Wait for next user input

```

[0397] In this pseudocode, the LLM is initialized and connected to the user's phone apps and services. The main interaction loop continuously listens for user input, analyzes the intent, and then performs the corresponding action, such as making a call, scheduling an event, performing a web search, or controlling an app.

[0398] The LLM's ability to understand natural language and extract relevant information allows it to serve as a centralized interface for the user, providing a seamless and personalized experience across multiple phone features and applications. This "super app" approach empowers the user

to accomplish a wide range of tasks without the need to switch between different apps, making the LLM a powerful and versatile tool for mobile interaction.

LLM Based APP Agent

[0399] A large language app model (LLAM) agent runs in the background to identify user desires and takes actions. Here is exemplary pseudocode outline for the Large Action Model (LAM) designed to make AI systems see and act on apps in a human-like manner:

```

# Initialize Large Language App Model (LLAM)
LLAM.initialize( )
# Define functions for LLAM operation
def observe_human_interface( ):
    # LLAM observes a human using an interface to learn the process
    human_interaction = observe_human( )
    return human_interaction
def replicate_process(human_interaction):
    # LLAM replicates the observed human interaction reliably
    replicated_process = LLAM.replicate_interaction(human_interaction)
    return replicated_process
def understand_human_intentions( ):
    # LLAM analyzes complex human intentions to determine objectives
    human_intentions = LLAM.analyze_intentions( )
    return human_intentions
def interact_with_apps(objectives):
    # LLAM spins up on human-oriented interfaces and interacts with apps to achieve objectives
    for objective in objectives:
        app_interface = LLAM.get_app_interface(objective)
        result = LLAM.interact_with_app(app_interface)
        if result == "success":
            continue
        else:
            handle_failure(result)
    def handle_failure(result):
        # Handle any failures or errors during app interaction
        if result == "error":
            log_error( )
            retry_interaction( )
    def log_error( ):
        # Log errors or issues encountered during app interaction
        log.error("Error encountered during app interaction")
    def retry_interaction( ):
        # Retry the interaction with the app after an error
        retry_result = LLAM.retry_interaction( )
        if retry_result == "success":
            continue
        else:
            escalate_issue( )
    def escalate_issue( ):
        # Escalate the issue if retries are unsuccessful
        escalate_result = LLAM.escalate_issue( )
        if escalate_result == "resolved":
            continue
        else:
            notify_user( )
    def notify_user( ):

```

-continued

```
# Notify the user if the issue cannot be resolved
user_notification = LLAM.notify_user("Unable to complete the task. Please check back later.")

# Main process loop for LLAM operation
while True:
    human_interaction = observe_human_interface()
    replicated_process = replicate_process(human_interaction)
    objectives = understand_human_intentions()
    interact_with_apps(objectives)
```

[0400] In this pseudocode, the LLAM is initialized and designed to observe, understand, and replicate human interactions with interfaces to achieve specific objectives within apps. The process involves observing a human using an interface, replicating the process reliably, understanding complex human intentions, spinning up on human-oriented interfaces, and interacting with apps to achieve objectives without requiring complex integrations like APIs.

[0401] The pseudocode outlines how LLAM can learn by demonstration, adapt to changing interfaces, and proactively carry out tasks across various mobile and desktop environments. It focuses on replicating observed interactions, understanding user intentions, interacting with apps to achieve objectives, handling errors or failures during interactions, and escalating issues when necessary. This approach fundamentally aims to streamline user experiences by automating tasks across multiple apps without users needing to download or use individual applications directly.

[0402] A method for creating a “super app” using a Learning and Optimization of Intelligent Agents or Mobile Apps (LLAM) system, the method comprising:

[0403] Identifying a plurality of mobile apps installed on a user’s device.

[0404] Analyzing the features, capabilities, and command sets of the identified mobile apps.

[0405] Extracting and storing the relevant information about the mobile apps in a knowledge base.

[0406] Based on detected user needs, sending instructions to a selected mobile app to perform a task for the user.

[0407] The method of claim 1, further comprising:

[0408] Monitoring the user’s interactions with the mobile apps over time.

[0409] Analyzing the user’s preferences, habits, and common tasks performed using the apps.

[0410] Updating the knowledge base with the user’s behavioral patterns and preferences.

[0411] The method of claim 2, wherein the LLAM system is configured to:

[0412] Understand natural language commands from the user.

[0413] Translate the user’s commands into actions to control and interact with the mobile apps.

[0414] The method of claim 3, further comprising:

[0415] Identifying the user’s current context, including location, time, and ongoing activities.

[0416] Selecting the most relevant mobile apps and features based on the user’s context and preferences.

[0417] The method of claim 4, wherein the LLAM system is capable of:

[0418] Seamlessly integrating and coordinating actions across multiple mobile apps to provide a unified user experience.

[0419] Reducing the need for the user to switch between different applications.

[0420] The method of claim 5, further comprising leveraging machine learning algorithms to personalize the “super app” interface and functionality based on the user’s evolving needs and preferences.

[0421] The method of claim 6, wherein the LLAM system is designed to:

[0422] Process and generate content in various formats, such as text, images, and even code.

[0423] Enable multimodal interaction, allowing the user to interact with the “super app” using different input and output methods.

[0424] The method of claim 7, further comprising deploying the LLAM system on the user’s device, enabling offline functionality and reducing reliance on network connectivity.

[0425] The method of claim 8, wherein the LLAM system is capable of:

[0426] Handling errors or failures during interactions with mobile apps.

[0427] Escalating issues to the user or seeking external assistance when necessary.

[0428] The method of claim 9, wherein the “super app” created by the LLAM system fundamentally aims to:

[0429] Streamline the user’s experience by automating tasks across multiple mobile apps.

[0430] Eliminate the need for the user to download and use individual applications directly.

[0431] The method creates a “super app” using an LLAM system that learns the features and commands of multiple mobile apps, examines user preferences, and deploys the apps in accordance with identified user needs and tasks, while considering user preferences and providing a seamless, personalized experience.

[0432] In another aspect, a method for an LLAM, comprising:

[0433] Observing human interactions with interfaces within apps;

[0434] Understanding complex human intentions;

[0435] Using voice and human-oriented interfaces; and

[0436] Translating the human intentions and Interacting with mobile apps and computer software to achieve specific objectives and acting as a SuperApp interface to the user and considering the user’s biometric state, location, appointments, observed user preferences, and other factors.

[0437] The method of claim 1, further comprising adapting to changing interfaces across mobile and desktop environments.

[0438] The method of claim 1, wherein the LLAM learns by demonstration.

[0439] The method of claim 1, wherein the LLAM proactively carries out tasks across various mobile and desktop environments.

[0440] The method of claim 1, involving replicating observed interactions accurately to achieve user-defined objectives.

[0441] The method of claim 1, wherein the LLAM handles errors or failures during interactions with interfaces.

[0442] The method of claim 1, wherein the LLAM escalates issues to human operators when necessary.

[0443] The method of claim 1, focusing on automating tasks across multiple apps to streamline user experiences.

[0444] The method of claim 1, enabling users to achieve objectives without the need to download or use individual applications directly.

[0445] The method of claim 1, enhancing user experiences by automating tasks seamlessly across various apps and interfaces.

Example: Hands-Free Assistance

[0446] Example pseudocode on how the natural language understanding and task automation capabilities of the earable device could provide users with seamless, hands-free assistance:

```
# Initialize earable device and connect to user's personal data
earable_device.initialize()
earable_device.connect_to_user_data()
# Continuously monitor user's context and biometrics
while True:
    user_location = earable_device.get_location()
    user_biometrics = earable_device.get_biometrics()
    user_calendar = earable_device.get_calendar()
    user_contacts = earable_device.get_contacts()
    # Provide navigation assistance
    if user_location_changed:
        navigation_instructions = llm.generate_navigation_instructions(user_location)
        earable_device.provide_voice_instructions(navigation_instructions)
    # Detect emergency situations
    if llm.detect_emergency(user_biometrics):
        earable_device.call_emergency_services()
    # Assist with social interactions
    if user_in_conversation:
        conversation_context = llm.analyze_conversation(user_biometrics, user_calendar,
user_contacts)
        earable_device.provide_discreet_suggestions(conversation_context)
    # Automate common tasks
    user_voice_command = earable_device.get_voice_command()
    task_response = llm.process_voice_command(user_voice_command)
    earable_device.execute_task(task_response)
    time.sleep(1) # Wait for next iteration
```

[0447] In this pseudocode, the earable device would continuously monitor the user's location, biometrics, calendar, and contacts to provide seamless, context-aware assistance. The device would leverage the natural language understanding (NLU) capabilities of the LLM for:

[0448] Navigation Assistance: Detect changes in the user's location and provide discreet, voice-based navigation instructions to guide the user through their surroundings.

[0449] Emergency Detection: Analyze the user's biometric data in real-time to detect potential emergency situations, such as a medical emergency or a fall, and automatically initiate the appropriate response, like calling emergency services.

[0450] Social Interaction Support: Assist the user during conversations by analyzing the context (user's calendar, contacts, and biometrics) to provide discreet suggestions for icebreakers, relevant information, or prompts to keep the discussion flowing smoothly.

[0451] Task Automation: Process the user's voice commands to automate common tasks, such as setting reminders, sending messages, or placing orders, without the user having to interact with a screen or manual controls.

[0452] The earable device would continuously monitor the user's context and respond accordingly, providing a seamless, hands-free experience powered by the LLM's natural language understanding and task automation capabilities.

Example Travel Assistance

[0453] Exemplary pseudocode for how the LLM could detect the user's intent and control an app, such as Lyft/Uber, to help the user get home after a meeting can include:

```
# Initialize earable device and connect to user's apps/services
earable_device.initialize()
earable_device.connect_to_user_apps()
earable_device.connect_to_user_calendar()
earable_device.connect_to_uber_app()
# Define intent detection and app control functions
def detect_meeting_end(calendar_data):
    # Analyze user's calendar to detect when the current meeting ends
    # and if there are no other meetings scheduled immediately after
```

-continued

```

current_meeting = calendar_data.get_current_meeting( )
next_meeting = calendar_data.get_next_meeting( )
if current_meeting.end_time < next_meeting.start_time:
    return True
else:
    return False
def confirm_uber_request(user_response):
    # Confirm user's response to the Uber request
    if user_response == "yes" or user_response == "confirm":
        return True
    else:
        return False
def request_uber(travel_mode, travel_time):
    # Use Uber app integration to request a ride
    if travel_mode == "fastest":
        uber_app.request_fastest_ride( )
    elif travel_mode == "cheapest":
        uber_app.request_cheapest_ride( )
    else:
        uber_app.request_ride( )
    # Return the estimated travel time
    return uber_app.get_estimated_travel_time( )
# Main control loop
while True:
    calendar_data = earable_device.get_calendar_data( )
    if detect_meeting_end(calendar_data):
        # Confirm with user if they want to request an Uber ride
        earable_device.provide_voice_prompt("Your meeting has ended. Would you like me to
request an Uber ride home?")
        user_response = earable_device.get_user_response( )
        if confirm_uber_request(user_response):
            # Request the Uber ride based on user's preference
            earable_device.provide_voice_prompt("Would you like the fastest or cheapest ride?")
            travel_mode = earable_device.get_user_response( )
            estimated_travel_time = request_uber(travel_mode, calendar_data.get_user_location( ))
            # Confirm the Uber request with the user
            earable_device.provide_voice_prompt("Okay, I've requested an {travel_mode} Uber
ride. The estimated travel time is {estimated_travel_time} minutes. Please confirm on your phone or tap
the earable surface to proceed.")
            user_confirmation = earable_device.get_user_confirmation( )
            if user_confirmation:
                earable_device.provide_voice_prompt("Ride requested. Enjoy your trip!")
            else:
                earable_device.provide_voice_prompt("Okay, I've canceled the Uber request.")
        else:
            earable_device.provide_voice_prompt("No problem, I won't request an Uber ride.")
    time.sleep(60) # Check for meeting end every minute

```

[0454] In this pseudocode, the earable device is connected to the user's calendar and the Uber app. The LLM is responsible for:

[0455] Detecting Meeting End: The LLM analyzes the user's calendar data to detect when the current meeting has ended and there are no other meetings scheduled immediately after.

[0456] Confirming Uber Request: The LLM prompts the user to confirm if they would like to request an Uber ride home. The user can respond verbally or by tapping the earable surface.

[0457] Requesting Uber Ride: If the user confirms, the LLM asks them to choose between the fastest or cheapest Uber ride option. It then uses the Uber app integration to request the appropriate ride and provides the estimated travel time.

[0458] Confirming Uber Request: The LLM prompts the user to confirm the Uber request on their phone or by tapping the earable surface. If the user confirms, the ride is booked, and the LLM provides a confirmation message. If the user declines, the LLM cancels the Uber request.

[0459] By integrating the LLM with the user's calendar and Uber app, the earable device can provide a seamless, hands-free experience in helping the user get home after a meeting, leveraging the LLM's intent detection and app control capabilities.

Example: Calendaring Meeting

[0460] To schedule an appointment based on available time and user preference learned from history, the LLM can follow a process that leverages historical data and user preferences to optimize the scheduling process. Here's an exemplary pseudocode outline for this scenario:

```

# Initialize LLM and connect to calendar/appointment app
LLM.initialize( )
LLM.connect_to_calendar_app( )

# Define functions for scheduling based on historical data and user preferences
def consolidate_calls_into_time_block( ):
    # Analyze historical call data to identify time blocks where calls are consolidated
    # based on predetermined criteria
    return consolidated_time_block

def infer_criteria_for_time_block_selection(calendar_request, attendee):
    # Use historical data and attendee information to infer criteria for selecting the time block
    inferred_criteria = analyze_data(calendar_request, attendee)
    return inferred_criteria

def schedule_appointment(time_block, criteria):
    # Utilize the calendar/appointment app (e.g., Calendly) to schedule the appointment
    if criteria == "poll_attendees":
        Calendly.poll_attendees_for_ideal_time(time_block)
    else:
        Calendly.schedule_appointment(time_block)

# Main scheduling loop
while True:
    calendar_request, attendee = LLM.get_calendar_request( )
    if all_calls_consolidated():
        time_block = consolidate_calls_into_time_block( )
        criteria = infer_criteria_for_time_block_selection(calendar_request, attendee)
        schedule_appointment(time_block, criteria)
    time.sleep(60) # Check for new calendar requests every minute

```

[0461] In this pseudocode, the LLM first initializes and connects to the calendar/appointment app. It then defines functions to analyze historical call data, infer criteria for selecting time blocks, and schedule appointments based on these insights. The main loop continuously checks for new calendar requests, consolidates calls into time blocks if needed, infers criteria for selecting time blocks, and schedules appointments accordingly using features like polling attendees for ideal meeting times in apps like Calendly.

Example: Connecting and Discovering App Features

[0462] The earable can discover available app and receive app messages. Exemplary pseudocode on how an LLM could control apps connected to the earable or mobile phone supporting the earable device can be:

```

# Initialize earable device and connect to user's apps
earable_device.initialize( )
earable_device.connect_to_user_apps( )

# Define LLM-powered app control functions
def launch_app(app_name):
    earable_device.launch_app(app_name)
def switch_app(app_name):
    earable_device.switch_to_app(app_name)
def close_app(app_name):
    earable_device.close_app(app_name)
def get_app_status( ):
    return earable_device.get_active_app()
def perform_app_action(app_name, action):
    earable_device.perform_app_action(app_name, action)

# Main control loop
while True:
    user_voice_command = earable_device.get_voice_command( )
    intent = llm.analyze_voice_command(user_voice_command)
    if intent == "launch_app":
        app_name = llm.extract_app_name(user_voice_command)
        launch_app(app_name)
    elif intent == "switch_app":
        app_name = llm.extract_app_name(user_voice_command)
        switch_app(app_name)
    elif intent == "close_app":
        app_name = llm.extract_app_name(user_voice_command)
        close_app(app_name)
    elif intent == "get_app_status":

```

-continued

```

active_app = get_app_status( )
earable_device.provide_voice_response(f"The active app is {active_app}")

elif intent == "perform_app_action":
    app_name = llm.extract_app_name(user_voice_command)
    action = llm.extract_app_action(user_voice_command)
    perform_app_action(app_name, action)

else:
    earable_device.provide_voice_response("I'm sorry, I didn't understand that command.")

time.sleep(1) # Wait for next iteration

```

[0463] In this pseudocode, the earable device would be initialized and connected to the user's various apps and services. The LLM would then be responsible for analyzing the user's voice commands and translating them into specific actions to control the connected apps.

[0464] The key functions include:

[0465] Launch App: The LLM can analyze the user's voice command to determine which app they want to launch, and then instruct the earable device to launch that app.

[0466] Switch App: Similarly, the LLM can detect when the user wants to switch to a different app and send the appropriate command to the earable device.

[0467] Close App: The LLM can also recognize when the user wants to close a specific app and issue the necessary command.

[0468] Get App Status: The LLM can provide the user with information about the currently active app on the earable device.

[0469] Perform App Action: The LLM can interpret more complex voice commands that involve performing specific actions within an app, such as playing music, sending a message, or setting a reminder.

[0470] By continuously monitoring the user's voice commands and translating them into the appropriate app control actions, the LLM can enable seamless, hands-free interaction with the user's connected apps and services through the earable device.

Autonomous Agents

[0471] a pseudocode outline for a temporal agent that can be sent by the Large Language Model (LLM) residing on the cloud to act as a virtual assistant and apply the knowledge warehouse concept discussed in the U.S. Pat. No. 8,131,718B2:

```

# Initialize the LLM on the cloud
class CloudLLM(LLM):
    def __init__(self):
        self.cloud_environment = "Cloud"
    def generate(self, prompt):
        # Generate response using the LLM on the cloud
        response = self.llm.generate(prompt)
        return response

# Define the Temporal Agent
class TemporalAgent:
    def __init__(self, name, cloud_llm):
        self.name = name
        self.cloud_llm = cloud_llm
        self.knowledge_warehouse = KnowledgeWarehouse()
    def act_as_virtual_assistant(self, user_input):
        # Analyze user input and determine appropriate actions
        intent = self.cloud_llm.generate_intent(user_input)
        if intent == "calendar_management":
            self.manage_calendar(user_input)
        elif intent == "information_retrieval":
            self.retrieve_information(user_input)
        else:
            # Handle other user requests
            pass
    def manage_calendar(self, user_input):
        # Leverage the knowledge warehouse to manage the user's calendar
        calendar_data = self.knowledge_warehouse.get_calendar_data()
        response = self.cloud_llm.generate_calendar_response(user_input, calendar_data)
        return response
    def retrieve_information(self, user_input):
        # Leverage the knowledge warehouse to retrieve relevant information
        information = self.knowledge_warehouse.search_information(user_input)
        response = self.cloud_llm.generate_information_response(user_input, information)
        return response

# Define the Knowledge Warehouse
class KnowledgeWarehouse:
    def __init__(self):
        (self): self.data_sources = []
    def add_data_source(self, data_source):
        self.data_sources.append(data_source)

```

-continued

```

def get_calendar_data(self):
    # Retrieve calendar data from the data sources
    calendar_data = [ ]
    for data_source in self.data_sources:
        calendar_data.extend(data_source.get_calendar_data( ))
    return calendar_data

def search_information(self, query):
    # Search for relevant information in the data sources
    information = [ ]
    for data_source in self.data_sources:
        information.extend(data_source.search_information(query))
    return information

# Main process
if __name__ == "__main__":
    # Initialize the cloud LLM
    cloud_llm = CloudLLM( )
    # Create the Temporal Agent
    temporal_agent = TemporalAgent("VirtualAssistant", cloud_llm)
    # Add data sources to the Knowledge Warehouse
    temporal_agent.knowledge_warehouse.add_data_source(CalendarDataSource( ))
    temporal_agent.knowledge_warehouse.add_data_source(DocumentDataSource( ))
    # Interact with the Temporal Agent
    user_input = "What's on my calendar today?"
    response = temporal_agent.act_as_virtual_assistant(user_input)
    print(response)

```

[0472] In this pseudocode, the TemporalAgent is responsible for acting as a virtual assistant, leveraging the capabilities of the CloudLLM and the KnowledgeWarehouse. The KnowledgeWarehouse represents the concept of the “knowledge warehouse” discussed in the U.S. Pat. No. 8,131,718B2 patent, which serves as a centralized repository for the user’s data, such as calendar information and other relevant documents. The TemporalAgent can handle various user requests, such as calendar management and information retrieval, by interacting with the KnowledgeWarehouse and generating appropriate responses using the CloudLLM. The KnowledgeWarehouse is designed to be extensible, allowing for the addition of various data sources that can be used to fulfill the user’s requests.

[0473] The pseudocode demonstrates how the Temporal Agent can be sent by the LLM residing on the cloud to act as a virtual assistant, while also incorporating the knowledge warehouse detailed in U.S. Pat. No. 8,131,718B2 patent invented by Bao Tran in 2005, the contents of which are incorporated by reference, to provide a more comprehensive and personalized user experience. In an embodiment called CloudHub, the warehouse provides a platform where users can grant access to their existing applications to enable various features and functionalities on their devices. Users can delegate tasks with their permission without storing their identity information or passwords preemptively. This approach eliminates the need for proxy accounts or additional subscriptions for existing services, enhancing safety, security, and efficiency. By leveraging Connected Apps, users can seamlessly integrate external applications without sharing sensitive credentials. This process allows users to authorize apps to access data within Anypoint Platform using OAuth 2.0 and OpenID Connect authentication protocols. Connected Apps can act on behalf of a user or on their own behalf (client credentials), providing flexibility in managing access rights and permissions. The benefits of using Connected Apps include auditable usage tracking, the ability to revoke granted access without requiring password changes, and seamless integration with external applications while maintaining security and privacy. This approach

ensures that users can delegate API access and log in to third-party applications using their Anypoint Platform credentials, enhancing the overall user experience without compromising security or requiring additional subscriptions. In summary, CloudHub’s Connected Apps feature enables users to delegate access to their existing applications securely and efficiently, ensuring that tasks are performed with user permission, maintaining high levels of safety and security without the need for proxy accounts or additional subscriptions.

[0474] The personal assistants assist the user in handling a fixed set of events, conditions, and actions taken when the designated event occurs under the stated conditions. In general, the intelligent agent possesses the following characteristics:

[0475] 1. Delegation: The user entrusts the agent to tackle some or all of an activity.

[0476] 2. Personalization: The user determines how the agent interacts. In many cases, the agent learns about the user and adapts its actions accordingly (along the lines of a personal assistant).

[0477] 3. Sociability: The agent is able to interact with other agents in ways similar to interpersonal communications: This includes some degrees of give and take, flexibility, and goal-oriented behavior.

[0478] 4. Predictability: The user has a reasonable expectation of the results.

[0479] 5. Mobility: The ability to go out—usually onto the network—to accomplish the delegated task.

[0480] 6. Cost effective: The benefits gained by the user, including time, information, filtering, among others, should be of greater value than the cost (monetary, time, re-work, etc.).

[0481] 7. Skills: The agent has its own expertise. A simple agent may be capable of only executing a simple command containing no ambiguity or the ability to effectively deal with the ambiguity in the command.

[0482] 8. Living within constraints: This can be as simple as, “find me the suit, but do not purchase it,” or become as complex as, “go only to the most likely

information sources, since there is a fee to just access an information source.” Some information services, for example, allow the user to set the maximum amount of money to be spent on any search.

[0483] Intelligent agents are typically built using rule-based expert systems as well as pattern recognizers such as neural networks. As discussed in U.S. Pat. No. 5,537,590, expert systems (also called knowledge systems) are defined to be intelligent computer programs that use knowledge and inference procedures to solve problems that are hard enough as to require in their solution, significant expertise. The rule based system then interacts with the LLMs to augment the expert system decisions.

[0484] To augment expert system decisions with general knowledge, a rule-based system or expert system interacts with Large Language Models (LLMs) to enhance decision-making processes. These systems leverage prescribed knowledge-based rules or expert models to solve problems by incorporating general knowledge from LLMs. The aim is to combine the expertise of human experts with the vast knowledge encapsulated in LLMs to improve decision accuracy and broaden the scope of problem-solving capabilities.

[0485] The interaction between rule-based systems or expert systems and LLMs involves integrating the structured rules or expert knowledge with the unstructured data and language understanding capabilities of LLMs. By doing so, these systems can benefit from the comprehensive information stored in LLMs to make more informed decisions, especially in complex scenarios where traditional rule-based systems may fall short due to limited predefined rules.

[0486] This approach allows for a more dynamic and adaptable decision-making process, where the expertise of human experts is complemented by the extensive knowledge base of LLMs. By combining these two sources of information, the system can make decisions that are not only based on predefined rules but also enriched by the broader context and insights provided by LLMs. This integration enables a more robust and versatile decision-making framework that can handle a wider range of scenarios and adapt to new information effectively. Exemplary Pseudocode for Software Combining Expert System and Large Language Model (LLM) is as follows:

-
1. Initialize Expert System Rules and LLM Model
 2. Define Function to Interact with Expert System and LLM:
 - Input: Query or Problem Statement
 - Output: Decision or Recommendation
 3. Main Program Loop:
 - Prompt User for Input
 - Call Function to Process Input using Expert System Rules and LLM
 - Display Decision or Recommendation
 4. Function to Process Input using Expert System and LLM:
 - Apply Expert System Rules to Analyze Input
 - Extract Key Information and Context
 - Pass Information to LLM for Language Understanding and General Knowledge
 - Combine Expert System Analysis with LLM Insights
 - Generate Decision or Recommendation based on Combined Analysis
-

[0487] The pseudocode outlines a software system that combines an expert system with an LLM to make decisions or provide recommendations based on user input. The software initializes the expert system rules and the LLM model at the start. It defines a function to interact with both systems, taking a query or problem statement as input and returning a decision or recommendation. The main program

loop prompts the user for input, processes it using the function that combines the expert system and LLM, and displays the resulting decision or recommendation. The processing function applies the expert system rules to analyze the input, extracts key information, sends this information to the LLM for language understanding and general knowledge, combines the results from both systems, and generates a decision or recommendation based on this combined analysis. The program ends after providing the decision or recommendation to the user. This pseudocode illustrates a basic structure for integrating an expert system with an LLM in software development to leverage both structured expert knowledge and unstructured language understanding capabilities for enhanced decision-making processes.

[0488] The assistant has an “enactor” that processes data from sensors and can change the environment through actuators. The enactor receives instructions from a “predictor/goal generator” which is connected to a “general knowledge warehouse” and specialized knowledge modules containing specialized neural networks and specialized LLMs. For example, one LLM may be dedicated to travel recommendations, while another LLM may be dedicated to daily consumer items and electronics, and yet another LLM can specialize in a technical domain such as medical domain. The knowledge warehouse and predictor/goal generator are connected to various expert modules like a scheduler, information locator, communicator, form filler, trainer, legal expert, medical expert, etc.

[0489] The assistant provides an interface that frees the user from learning complex search languages and allows some functions to be automatically performed. The assistant uses machine learning to learn the user's styles, techniques, preferences and interests. The information locator generates queries based on user characteristics to retrieve data of interest, submits the queries, communicates the results to the user, and updates the knowledge warehouse. The assistant supports refining queries, managing search/LLM costs, and automatically incorporating changes to search interfaces and information sources.

[0490] The assistant schedules and executes multiple information retrieval tasks based on user priorities, dead-

lines and preferences using the scheduler. When operating with a handheld device, the assistant splits into two personalities—one on the handheld for user interaction and one on a cloud computer for executing background searches. The assistant intelligently interacts with the user, filters information, and provides timely access to relevant data to enable accurate situational analysis by the user.

[0491] In addition to LLMs, the system can use expert (knowledge) systems with two basic elements: inference engine and knowledge base. The knowledge base holds all information related to the tasks at hand: the rules and the data on which they will be applied. The inference engine is a mechanism that can operate the information contained in the knowledge base. In a rule-based system, the knowledge base is divided into a set of rules and working memory (or database). Expert systems typically consist of an interpretive language where the user may write his or her program statements and the conditions associated with those statements, an inference engine, which provides the mechanism through which the expert rules are interpreted and fired, and an executive front-end or expert shell, that helps users write application programs using the language, and that helps them run the expert applications developed, and that also helps them develop and query reports or the generated diagnostics. In the expert system, just like an IF-THEN sentence, each rule has two parts: a premise and a conclusion. A rule is said to be fired when the inference engine finds the premise is stored as TRUE in working memory (the knowledge base) and it incorporates the conclusion of the rule to the working memory (knowledge base) too. Working memory is the database contained in the knowledge base. This holds all facts that describe the current situation. Generally, the expert system will start with very few facts. These will expand as the system learns more about the situation at hand, and as far as some rules are executed. The inference engine or rule interpreter has two tasks. First, it examines facts in working memory and rules in the rule base, and adds new facts to the database (memory) when possible. That is, it fires rules. Second, it determines in what order rules are scanned and fired. The inference engine can determine the order in which rules should be fired by different methods such as forward chaining, backward chaining, breadth- or depth-wise scan techniques, etc. Applications that use forward chaining, such as process control, are called data-driven. Applications that use backward chaining are called goal-driven. Forward chaining systems are typically used where relevant facts are contained in small sets and where many facts lead to few conclusions. A forward chaining system must have all its data at the start, rather than asking the user for information as it goes. Backward chaining should be used for applications having a large set of facts, where one fact can lead to many conclusions. A backward-chaining system will ask for more information if needed to establish a goal.

[0492] The LLAM can deploy a plurality of intelligent agent(s), each can autonomously execute tasks or subtasks, running on the cloud and cooperating with other agents as a swarm to achieve a goal:

Initialization and Setup:

[0493] The intelligent agent is initialized and connected to the cloud environment.

[0494] The agent is designed to cooperate with other agents as a swarm to achieve a common goal.

[0495] The agent has access to a knowledge warehouse and various specialized modules, such as an information locator, scheduler, communicator, and others.

Task Execution:

[0496] The user or a coordinating agent assigns a task or subtask to the intelligent agent.

[0497] The agent analyzes the task requirements and breaks it down into smaller, executable steps.

[0498] The agent leverages the specialized modules and the knowledge warehouse to gather the necessary information and resources to complete the task.

Information Locator Operation:

[0499] The information locator module within the agent requests a search budget from the user, either in monetary terms or time spent.

[0500] The agent identifies the search domain based on the user's prior history and preferences, allowing the user to approve the suggested domain.

[0501] The agent then prioritizes the search request relative to any outstanding requests, again using the user's prior history and preferences.

[0502] The agent sets the deadline and search intervals for the task.

[0503] The information locator generates a search query based on the user's general discussion of the search topic.

[0504] The agent refines the search query by expanding it using a thesaurus and searching the user's local disk space for relevant terms and concepts.

[0505] The agent tailors the query to maximize the search results based on its knowledge of the user's styles, techniques, preferences, and interests.

[0506] The agent adds the query to a search launchpad database and broadcasts it to one or more information sources.

[0507] Upon receiving the search results, the agent communicates them to the user and updates the knowledge warehouse with the user's responses.

Swarm Coordination:

[0508] If the task requires cooperation with other agents, the agent coordinates with the swarm to divide the work and share information.

[0509] The agents communicate with each other, share resources, and synchronize their actions to achieve the common goal.

[0510] The agents continuously monitor the progress and adjust their plans based on feedback from the other agents and the user.

Adaptation and Learning:

[0511] The agent tracks the user's behavior and preferences regarding the search results, storing rejected documents and refining the search boundaries.

[0512] The agent uses the user's requests to refine its profile and update the profiles database, improving its ability to provide relevant information in the future.

[0513] The agent's knowledge and capabilities are continuously updated through interactions with the user and the swarm of agents, enabling it to adapt to changing requirements and environments.

[0514] By operating in this manner, the intelligent agent can autonomously execute tasks and subtasks, leveraging the information locator and swarm coordination to efficiently gather and process information, adapt to user preferences, and work collaboratively with other agents to achieve the desired goals.

[0515] The pseudocode for LLAM (Learning and Optimization of Intelligent Agents or Mobile Apps) that discovers

the features/capabilities of each intelligent agent or mobile app and learns how to optimize their deployment can be outlined as follows:

```

# Initialize LLAM for learning and optimization
class LLAM:
    def __init__(self):
        self.intelligent_agents = []
        self.mobile_apps = []
    def add_intelligent_agent(self, agent):
        self.intelligent_agents.append(agent)
    def add_mobile_app(self, app):
        self.mobile_apps.append(app)
    def discover_features(self):
        for agent in self.intelligent_agents:
            agent.extract_features()
        for app in self.mobile_apps:
            app.extract_features()
    def learn_optimization(self):
        # Implement learning algorithms to optimize the deployment of agents and apps
        pass
# Define the intelligent agent class
class IntelligentAgent:
    def __init__(self, name, features):
        self.name = name
        self.features = features
    def extract_features(self):
        # Extract features/capabilities of the intelligent agent
        pass
# Define the mobile app class
class MobileApp:
    def __init__(self, name, features):
        self.name = name
        self.features = features
    def extract_features(self):
        # Extract features/capabilities of the mobile app
        pass
# Instantiate LLAM and add intelligent agents and mobile apps
llam = LLAM()
agent1 = IntelligentAgent("Agent1", ["Feature1", "Feature2"])
app1 = MobileApp("App1", ["FeatureA", "FeatureB"])
llam.add_intelligent_agent(agent1)
llam.add_mobile_app(app1)
# Discover features of agents and apps
llam.discover_features()
# Learn how to optimize deployment
llam.learn_optimization()

```

[0516] The above pseudocode outlines a basic structure for LLAM to discover the features/capabilities of each intelligent agent or mobile app and then learn how to optimize their deployment based on these features.

Deploying Swarms of Intelligent Agents

[0517] A swarm of intelligent agents can be sent to cooperatively and collectively collect intelligence on behalf of a goal. Similar to a virus, the agents self replicate and distribute themselves until they reach a termination point or a termination signal is sent. A high-level pseudocode for the individual agents in a swarm that operate together to achieve a goal:

```

# Pseudocode for each agent in the swarm
# Initialize the agent
Initialize the agent with its specific capabilities,
lifetime, and cost structure
Set the agent's initial position and state
# Main loop
While agent is active:
    # Gather information

```

-continued

```

Observe the local environment and collect relevant information
# Communicate with other agents
Share the collected information with other agents in the swarm
Receive information from other agents
# Analyze information
Combine the information from various sources
Evaluate the current state and determine the best course of action
# Take action
Execute the determined action to contribute to the overall goal
# Update state
Update the agent's internal state based on the action taken
# Check termination conditions
If the agent's lifetime has expired or the goal has been achieved:
    Terminate the agent

```

[0518] Each agent is initialized with its specific capabilities, lifetime, and cost structure, which define its role and constraints within the swarm. The agents operate in a loop, continuously gathering information from the local environment, communicating with other agents, analyzing the combined information, and taking actions to contribute to the overall goal. The agents share information with each other to leverage the collective knowledge and capabilities of the

swarm. The agents update their internal state based on the actions taken and monitor for termination conditions, such as the expiration of their lifetime or the achievement of the overall goal. This high-level pseudocode demonstrates the key aspects of the individual agents within a swarm, where they work together in a decentralized and autonomous manner to achieve a common objective. The specific implementation details, such as the communication protocols, decision-making algorithms, and coordination mechanisms, would depend on the particular application and the requirements of the swarm intelligence system.

[0519] A method for deploying swarms of intelligent agents on a user device, the method comprising:

- [0520]** Defining a set of agent types, each agent type having a fixed lifetime and/or cost structure, wherein the agent types are specific to skills, expertise, languages, or locations; Receiving a user request for information or a decision on a particular topic;
- [0521]** Selecting a subset of agent types based on the user request, the selected agent types being relevant to the requested information or decision;
- [0522]** Instantiating a swarm of intelligent agents of the selected agent types on the user device;
- [0523]** Configuring the swarm of agents to operate autonomously, each agent collecting relevant information locally and reporting its findings to the user device or to an information warehouse;
- [0524]** Aggregating the information collected by the swarm of agents and generating a recommendation or decision based on the aggregated data;
- [0525]** Presenting the recommendation or decision to the user.
- [0526]** The method of claim 1, wherein the agent types are defined based on a predefined taxonomy or ontology, allowing for the selection of relevant agent types based on the user request.

[0527] The method of claim 1, wherein the fixed lifetime of the agents is determined based on the expected duration of the user request or the available computational resources on the user device.

[0528] The method of claim 1, wherein the cost structure of the agents is determined based on the complexity of the information being collected or the computational resources required to process the collected data.

[0529] The method of claim 1, wherein the agents operate autonomously by using local search algorithms, distributed decision-making, or other decentralized coordination mechanisms to efficiently gather and aggregate information.

[0530] The method of claim 1, wherein the recommendation or decision generated by the user device is further refined or validated by consulting external knowledge sources or expert systems.

[0531] The method of claim 1, wherein the user device is a mobile device, a desktop computer, or a smart home device, and the swarm of agents is deployed and managed within the constraints of the user device's hardware and software capabilities.

[0532] The method of claim 1, wherein the user device maintains a history of previous agent swarm deployments and uses this information to optimize the selection and configuration of agent types for future requests.

[0533] The method of claim 1, wherein the user device provides a user interface for monitoring the progress of the agent swarm and adjusting the parameters of the deployment as needed.

[0534] The method of claim 1, wherein the agent swarm deployment is integrated with other services or applications on the user device, such as personal assistants, task management tools, or decision support systems.

Complex Example: Vacation Trip Planning

[0535] For example, below is a pseudocode example of how a swarm of intelligent agents could plan a month-long vacation in a new location, with each agent responsible for a sub-trip task or a particular city, and a main coordinating module that balances all the constraints and recommends a trip plan to the user:

```

# Initialize the main vacation planning module
class VacationPlanningModule:
    def __init__(self):
        self.agents = []
        self.trip_plan = {}
        self.constraints = {}
    def add_agent(self, agent):
        self.agents.append(agent)
    def coordinate_agents(self):
        for agent in self.agents:
            agent.plan_sub_trip()
        self.trip_plan.update(agent.get_sub_trip_plan())
        self.constraints.update(agent.get_constraints())
        self.balance_constraints()
        self.recommend_trip_plan()
    def balance_constraints(self):
        # Analyze the constraints from all agents and balance to create a cohesive and feasible trip plan
    def recommend_trip_plan(self):
        # Present the balanced trip plan to the user
        print("Recommended Vacation Plan:", self.trip_plan)
    # Define the agent classes for different sub-tasks or cities
class CityAgent:
    def __init__(self, city_name):
        self.city_name = city_name
        self.sub_trip_plan = {}
        self.constraints = {}
    def plan_sub_trip(self):
        # Plan the sub-trip for the assigned city
        self.sub_trip_plan = {

```

-continued

```

    "Accommodation": "Hotel XYZ",
    "Activities": ["Museum A", "Park B", "Restaurant C"],
    "Transportation": "Public Transit"
}
self.constraints = {
    "Budget": 2000,
    "Duration": 5,
    "Preferred Dates": ["2023-06-15", "2023-06-16", "2023-06-17"]
}
def get_sub_trip_plan(self):
    return {self.city_name: self.sub_trip_plan}
def get_constraints(self):
    return {self.city_name: self.constraints}
class ActivityAgent:
    def __init__(self, activity_type):
        self.activity_type = activity_type
        self.sub_trip_plan = { }
        self.constraints = { }
    def plan_sub_trip(self):
        # Plan the sub-trip for the assigned activity type
        self.sub_trip_plan = {
            "Activity 1": "Museum Tour",
            "Activity 2": "Hiking Trail",
            "Activity 3": "Cooking Class"
        }
        self.constraints = {
            "Budget": 500,
            "Duration": 2,
            "Preferred Dates": ["2023-06-20", "2023-06-21"]
        }
    def get_sub_trip_plan(self):
        return {self.activity_type: self.sub_trip_plan}
    def get_constraints(self):
        return {self.activity_type: self.constraints}
# Main process
if __name__ == "__main__":
    vacation_planning_module = VacationPlanningModule( )
    city_agent_1 = CityAgent("Paris")
    city_agent_2 = CityAgent("Rome")
    activity_agent_1 = ActivityAgent("Sightseeing")
    activity_agent_2 = ActivityAgent("Culinary")
    vacation_planning_module.add_agent(city_agent_1)
    vacation_planning_module.add_agent(city_agent_2)
    vacation_planning_module.add_agent(activity_agent_1)
    vacation_planning_module.add_agent(activity_agent_2)
    vacation_planning_module.coordinate_agents()

```

[0536] In this pseudocode, the VacationPlanningModule is the main coordinating module that manages the swarm of intelligent agents responsible for planning different aspects of the vacation.

[0537] The agents include:

[0538] CityAgent: Responsible for planning the sub-trip for a specific city, including accommodation, activities, and transportation.

[0539] ActivityAgent: Responsible for planning the sub-trip for a specific activity type, such as sightseeing or culinary experiences.

[0540] Each agent plans its sub-trip and provides the VacationPlanningModule with the sub-trip plan and the associated constraints (e.g., budget, duration, preferred dates). The VacationPlanningModule then coordinates the agents, collects the sub-trip plans and constraints, balances the constraints to create a cohesive and feasible trip plan, and finally recommends the overall vacation plan to the user. The balance_constraints() method in the VacationPlanningModule is responsible for analyzing the constraints from all agents and finding an optimal solution that satisfies the user's preferences and requirements. This pseudocode demonstrates how a swarm of intelligent agents can work

together to plan a complex, multi-faceted vacation, with each agent responsible for a specific sub-task or city, and the main coordinating module ensuring that the overall trip plan is balanced and recommended to the user. The LLAM can check its recommendations by looking at user reviews and comments on the Internet, as well as checking the reputation of the sources it relied on: Analyzing User Reviews and Comments. The LLAM would search the internet for user reviews and comments related to the recommended items or services. It would analyze the sentiment, tone, and content of these reviews to gauge the overall user perception and satisfaction. The LLAM would look for patterns in the reviews, such as common praise or complaints, to better understand the strengths and weaknesses of the recommendations. It would also pay attention to the credibility and trustworthiness of the review sources, as discussed in the search results. Evaluating Source Reputation: The LLAM would assess the reputation and credibility of the sources it relied on to make the recommendations. It would look at factors such as the authority, objectivity, and relevance of the sources, as outlined in the search results. The LLAM would prioritize recommendations from reputable, authoritative, and unbiased sources over those from less credible or

potentially biased sources. It would also consider the overall trustworthiness of the websites or platforms hosting the information, as discussed in the search results. Iterative Refinement: Based on the analysis of user reviews and source credibility, the LLAM would refine its recommendation algorithms and models. It would learn from the user feedback and adjust its future recommendations to better align with user preferences and needs. Transparency and Accountability: The LLAM would strive to be transparent about its recommendation process, including the sources it relied on and the factors it considered. It would provide

porating the analysis of user reviews, source reputation, and iterative refinement, the LLAM can enhance the credibility and reliability of its recommendations, aligning them more closely with user preferences and needs, as outlined in the search results.

Complex Example: Startup Launch

[0542] In another example, a pseudocode for the LLAM (Large Language Model) assistance with launching a new startup:

```

python
# Initialize the LLAM and create intelligent agents
llam = LLAM( )
market_research_agent = IntelligentAgent("Market Research")
customer_validation_agent = IntelligentAgent("Customer Validation")
fundraising_agent = IntelligentAgent("Fundraising")
software_implementation_agent = IntelligentAgent("Software Implementation")
gtm_planning_agent = IntelligentAgent("GTM Planning")
# Define the startup launch process
def launch_startup( ):
    # Market Research
    market_research_agent.set_budget(10000)
    market_research_agent.set_lifetime(30)
    market_research_agent.perform_market_research( )
    market_research_results = market_research_agent.get_results( )
    # Customer Validation
    customer_validation_agent.set_budget(5000)
    customer_validation_agent.set_lifetime(45)
    customer_validation_agent.validate_customers(market_research_results)
    customer_validation_results = customer_validation_agent.get_results( )
    # Fundraising
    fundraising_agent.set_budget(50000)
    fundraising_agent.set_lifetime(90)
    fundraising_agent.raise_funds(customer_validation_results)
    fundraising_results = fundraising_agent.get_results( )
    # Software Implementation
    software_implementation_agent.set_budget(30000)
    software_implementation_agent.set_lifetime(120)
    software_implementation_agent.implement_software(fundraising_results)
    software_implementation_results = software_implementation_agent.get_results( )
    # GTM Planning
    gtm_planning_agent.set_budget(20000)
    gtm_planning_agent.set_lifetime(60)
    gtm_planning_agent.plan_go_to_market(software_implementation_results)
    gtm_planning_results = gtm_planning_agent.get_results( )
    # Provide the overall startup launch results to the user
    return {
        "Market Research": market_research_results,
        "Customer Validation": customer_validation_results,
        "Fundraising": fundraising_results,
        "Software Implementation": software_implementation_results,
        "GTM Planning": gtm_planning_results
    }
# Main process
if __name__ == "__main__":
    startup_launch_results = launch_startup( )
    llam.present_results_to_user(startup_launch_results)

```

users with clear explanations for its recommendations, allowing them to better understand the reasoning behind the suggestions. This transparency would help build user trust and accountability, as discussed in the search results.

[0541] The LLAM would also update its knowledge of the various intelligent agents or mobile apps, their features, and how to optimize their deployment, as described in the earlier pseudocode. LLMs can perform sentiment analysis due to their deep understanding of language nuances and context, enabling accurate determination of sentiment behind texts such as social media posts and customer reviews. By incor-

[0543] In this pseudocode, the LLAM initializes several intelligent agents, each responsible for a specific subtask in launching a new startup. The `launch_startup()` function coordinates the activities of these agents, providing them with a budget and a finite lifetime and follows the steps: Market Research—The `market_research_agent` performs market research and provides the results; Customer Validation—The `customer_validation_agent` validates the customers based on the market research results; Fundraising—The `fundraising_agent` raises funds using the customer validation results; Software Implementation—The `software_implementation_agent` implements the software using the fund-

raising results. GTM Planning—The gtm_planning_agent plans the go-to-market strategy using the software implementation results. The LLAM then presents the overall startup launch results to the user, integrating the information from the various intelligent agents. The agents use the information_locator to search for relevant information, refine the search queries, and communicate the results to the user, updating the knowledge warehouse as necessary. The foregoing demonstrates how the LLAM can coordinate the activities of multiple intelligent agents, each with a finite lifetime and budget, to assist the user in performing a complex operation such as launching a new startup.

[0544] FIG. 9A is a flowchart demonstrating how the trainer operates. Starting at step 290, the process moves to step 291 to determine if the user wants to manually train the intelligent assistant by providing explicit sequences, akin to a macro training operation. If manual training is desired, the routine advances to step 292 to request a specific task for training. Subsequently, at step 293, user strokes and associated task activities are captured. Alternatively, if the assistant is to learn through inference, the process transitions from step 291 to step 294 where user strokes and application activities are captured over a set period. Following this, in step 295, patterns in the strokes and activities are analyzed. Upon establishing a pattern in step 296, the routine proceeds to step 297 where data is grouped by clustering user strokes and application patterns. Moving on from step 297 to step 298, clusters are partitioned into new categories. Finally, at step 299, new categories are merged with existing ones. Either from step 293 or step 299, the process in FIG. 9A proceeds to train the assistant at step 300. Post-training, the newly trained module is integrated into the intelligent assistant's functionality at step 301. The routine then exits from either step 301 or step 296. FIG. 9B shows exemplary operation process of the information locator. Beginning at step 310, a search budget is requested from the user, which can be monetary or time-based. Subsequently, at step 311, the user identifies the search domain with suggested options based on historical preferences for approval. Step 312 involves prioritizing search requests based on past interactions and preferences. Step 313 identifies deadlines and search intervals. From step 313, the electronic assistant generates a search query in step 314 based on a general discussion of the topic by the user. Steps 315-316 refine this query by expanding it using a thesaurus and searching local disk space for relevant terms based on personal work space content. The information locator tailors queries according to user styles and preferences for an optimized search outcome. Subsequently, at step 317, queries are added to a database tracking all search requests before broadcasting them to information sources for results retrieval. The information locator presents a list of keywords from searches that identify potential documents for user action selection. Users can specify item quantities and preferred activation times for searches based on current profile preferences. The assistant tracks user behavior regarding retrieved documents in surfing and query modes, refining searches by comparing accepted and rejected documents to adjust search boundaries and enhance result relevance. While the electronic assistant may reside on a cloud computer, the information locator can be situated on either a host or handheld device. In environments with both devices working together, such as a handheld computer interacting with a host computer, two personalities of the assistant emerge: one on each device for

user interaction and background searches respectively. The host computer prioritizes retrieved documents and optimizes data transmission between devices for efficient wireless communication during docking sessions to minimize costs while ensuring consistent knowledge synchronization between devices upon synchronization.

[0545] While the invention has been disclosed in connection with the preferred embodiments shown and described in detail, various modifications and improvements thereon will become readily apparent to those skilled in the art. Accordingly, the spirit and scope of the preferred embodiment is not to be limited by the foregoing examples, but is to be understood in the broadest sense allowable by law.

What is claimed is:

1. A method for assisting a user, comprising:
receiving a verbal request from the user with a device coupled to an ear, wherein the device includes biometric sensors coupled to the ear;
selecting a learning machine language model or an intelligent agent based on the user request, the selected model or agent applying user environmental information including positioning information or nearby images in responding to the requested information or decision; and
providing a response to the verbal request.
2. The method of claim 1, comprising applying a learning machine to identify an aural environment and adjust the amplifiers for optimum hearing or to adjust amplifier parameters for a particular environment with a predetermined noise pattern.
3. The method of claim 1, comprising capturing vital signs with the in-ear device.
4. The method of claim 1, comprising capturing a user environment with a camera facing away from the ear, and providing the user environmental information to the learning machine language model or intelligent agent for use with the response.
5. The method of claim 1, comprising deploying one or more intelligent agents to provide the requested information, wherein each agent has a fixed lifetime or fixed cost structure, wherein the agent is specific to skills, expertise, languages, or locations.
6. The method of claim 1, wherein agent types are defined based on a predefined taxonomy or ontology, allowing for the selection of relevant agent types based on the user request.
7. The method of claim 1, wherein the fixed lifetime of the agent is determined based on an expected duration of the user request or available computational resources.
8. The method of claim 1, wherein the cost limit for the agent is determined based on the complexity of the information being collected or the computational resources required to process the collected data.
9. The method of claim 1, wherein the one or more intelligent agents operate autonomously by using local search algorithms, distributed decision-making, or other decentralized coordination mechanisms to efficiently gather and aggregate information.
10. The method of claim 1, wherein the recommendation or decision is further refined or validated by consulting external knowledge sources or expert systems.

11. The method of claim **1**, comprising maintaining a history of previous agent swarm deployments to optimize the selection and configuration of agent types for future requests.

12. The method of claim **1**, wherein the user device provides a user interface for monitoring the progress of the agent swarm and adjusting the parameters of the deployment as needed.

13. The method of claim **1**, wherein the agent swarm deployment is integrated with other services or applications on the user device, such as personal assistants, task management tools, or decision support systems.

14. The method of claim **1**, comprising providing super app functionality for the user with a large language model (LLM) and a plurality of software or apps.

15. The method of claim **14**, further comprising:
identifying a plurality of apps installed on a user's device;
determining features or capabilities of the identified apps
and one or more commands to access the features or
capabilities;
extracting and storing the features or capabilities of the
apps in a knowledge base;
applying the LLM to identify a user need;
identifying one or more selected apps that address the user
need; and

sending one or more commands to the selected app for the
user.

16. The method of claim **15**, further comprising monitoring user interactions with the apps over time; analyzing user preferences, habits, or common tasks performed using the apps; and

updating a database with the user's behavioral patterns
and preferences.

17. The method of claim **1**, further comprising identifying the user's current context, including location, time, and ongoing activities and selecting the one or more mobile apps based on the user's context and preferences.

18. The method of claim **1**, comprising a large language model (LLM) configured to understand requests from the user and to translate the request into actions to control and interact with hardware or with software apps.

19. The method of claim **18**, wherein the LLM generates content as text, images, and code for software apps and wherein the LLM enables multimodal interaction, allowing the user to interact with a "super app" using different input and output methods.

20. The method of claim **7**, further comprising deploying the LLM system on the user's device, enabling offline functionality and reducing reliance on network connectivity.

* * * * *