



- (51) International Patent Classification:
A61B 5/04 (2006.01) *G06F 19/00* (2011.01)
- (21) International Application Number:
PCT/SG2017/050508
- (22) International Filing Date:
10 October 2017 (10.10.2017)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
10201608507P 11 October 2016 (11.10.2016) SG
- (71) Applicant: NATIONAL UNIVERSITY OF SINGAPORE [SG/SG]; 21 Lower Kent Ridge Road, Singapore 119077 (SG).
- (72) Inventors: PATANAIAK, Amiya; c/o Duke-NUS Medical School, Level 6, Cognitive Neuroscience Laboratory, 8 College Road, Singapore 169857 (SG). CHEE, Wei Liang, Michael; c/o Duke-NUS Medical School, Level 6, Cognitive Neuroscience Laboratory, 8 College Road, Singapore 169857 (SG). LEE, Jian Yun; 127 Chempaka Kuning Link, Singapore 486331 (SG). ONG, Ju Lynn; c/o Duke-NUS Medical School, Level 6, Cognitive Neuroscience Laboratory, 8 College Road, Singapore 169857 (SG).

(74) Agent: SPRUSON & FERGUSON (ASIA) PTE LTD;
P.O. Box 1531, Robinson Road Post Office, Singapore 903031 (SG).

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

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

(54) Title: DETERMINING SLEEP STAGES

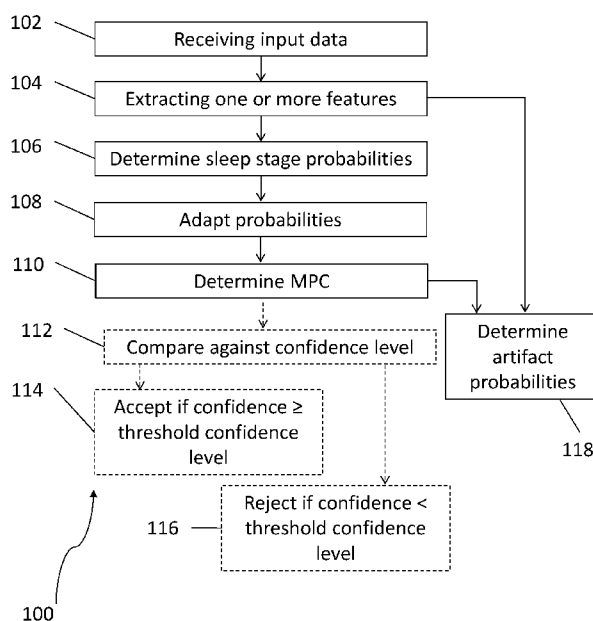


FIG. 1

(57) Abstract: Disclosed is a method and system for automatically determining sleep stages. The method comprises receiving input data representing a continuous sequence of sleep epochs measured from a subject; extracting one or more features for each of a plurality of sleep epochs and applying a plurality of weightings for emphasising at least one feature of each sleep epoch of the plurality of sleep epochs; determining from the weighted features of the respective sleep epoch a sleep stage probability for each of a plurality of sleep stages, each sleep stage probability defining a likelihood the subject was in the respective sleep stage at the time the respective sleep epoch was measured; adapting one or more of the sleep stage probabilities of each sleep epoch based on one or more sleep stage probabilities of at least one neighbouring sleep epoch in the plurality of sleep epochs; and determining the final most probable class (FMPC) of the respective sleep epoch, the FMPC being the sleep stage with the highest adapted probability.

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

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

DETERMINING SLEEP STAGES

TECHNICAL FIELD

[1] The present disclosure relates to automation of the process of sleep
5 scoring. Methods disclosed herein are further applicable to profiling of
 sleep stages in real-time.

BACKGROUND

[2] Sleep has a strong bi-directional relationship with health and well-being.
10 Sleep disorders affect as many as 30% of the population. Sleep consists of
 different stages. Sleep architecture refers to the duration and temporal
 arrangement of these stages and is disrupted in sleep disorders.

[3] To standardise sleep staging, the American Academy of Sleep Medicine
 (AASM) developed guidelines for terminology and scoring of sleep stages
15 using rules based on the R&K score, named after the key developers, Allan
 Rechtschaffen and Anthony Kales. According to the AASM, the different
 stages of a sleep cycle include:

 wake;
 stages S1, S2 and S3/S4 of the R&K rule; and
20 rapid eye movement (REM) sleep.

[4] To facilitate scoring or profiling of sleep stages according to the above
 rules, polysomnographic (PSG) equipment is used for recording, among
 others, electroencephalogram (EEG) and electrooculograms (EOG)
 measurements. With the recordings from multiple sensors a trained
25 specialist manually annotates the sleep stages according to the AASM

scoring system. This process is a highly manual, visual sleep scoring method that is difficult, time consuming and costly. Moreover, the scoring or annotation process has an element of subjectivity in that experts will differ on how to stage some epochs – there being typically only 80% to 82% congruence in expert opinions.

[5] Some attempts have been made to computerise the scoring of sleep stages.

Some of these attempts provide home-based monitoring systems such as wrist-worn devices. However, while wrist-worn and other devices are becoming increasingly popular, they only measure surrogates of sleep such as respiratory rate, motion and heart rate. The gold standard for sleep measurement remains a polysomnography annotated by a human expert.

[6] The manual sleep scoring process employs 30-second sequential epochs.

The frequency and amplitude of waveforms are measured and the expert applies the standardised criteria for scoring sleep stages. One such standardisation requires the expert to accept the sleep stage that defines the majority of an epoch in the event that two or more stages co-exist during a single epoch. The file sizes of the recordings are relatively large (~150 Mb).

[7] What is needed is a method for profiling or determining sleep stages that is as accurate and more reliable than expert opinion. It is also desirable that there be provided a reliable sleep scoring method that can be employed close to, or in, real-time.

SUMMARY OF THE PRESENT DISCLOSURE

- [8] The present disclosure provides a method for automatically determining sleep stages, comprising:
- receiving input data representing a continuous sequence of sleep epochs
- 5 measured from a subject;
- extracting one or more features for each of a plurality of sleep epochs and
- applying a plurality of weightings for emphasising at least one feature of
- each sleep epoch of the plurality of sleep epochs;
- determining, from the weighted features of the respective sleep epoch, a
- 10 sleep stage probability for each of a plurality of sleep stages, each sleep
- stage probability defining a likelihood the subject was in the respective
- sleep stage at the time the respective sleep epoch was measured;
- adapting the sleep stage probabilities of each sleep epoch based on one or
- more sleep stage probabilities of at least one neighbouring sleep epoch in
- 15 the plurality of sleep epochs; and
- determining the final most probable class (FMPC) of the respective sleep
- epoch, the FMPC being the sleep stage with the highest adapted
- probability.
- [9] The present disclosure further provides a system for automatically
- 20 determining sleep stages, comprising at least one processor and at least
- one memory unit communicatively coupled to each respective
- processor and comprising instructions that, when executed by the
- processor cause the system to:
- receive input data from one or more sensors, the data representing a
- 25 continuous sequence of sleep epochs measured from a subject;

extract one or more features for each of a plurality of the sleep epochs and
apply a plurality of weightings for emphasising at least one feature of each
sleep epoch of the plurality of sleep epochs;

determine, from the weighted features of the respective sleep epoch, a

5 sleep stage probability for each of a plurality of sleep stages, each sleep
stage probability defining a likelihood the subject was in the respective
sleep stage at the time the respective sleep epoch was measured;

adapt the sleep stage probabilities of each sleep epoch based on one or
more sleep stage probabilities of at least one neighbouring sleep epoch in

10 the plurality of sleep epochs;

determine the final most probable class (FMPC) of the respective sleep
epoch, the FMPC being the sleep stage with the highest adapted
probability; and

output to a display the FMCP of the respective epoch.

15 [10] Embodiments of the invention may enable the sleep stage of a
subject – which may be interchangeably referred to as a person, human,
human subject, patient or similar – to be determined in during the epoch
for which the sleep stage is being determined.

20 BRIEF DESCRIPTION OF THE DRAWINGS

[11] In the drawings:

FIG. 1 illustrates a method for automatic sleep profiling – in other
words, a method for automatically determining sleep stages;

FIG. 2 is a schematic overview of a system for performing the
25 method of FIG. 1;

FIG. 3 shows a schematic overview of an exemplary pre-processor module used in the system of FIG. 2;

FIG. 4 shows a stacked spectrogram presently comprising spectrograms of three channels;

5 FIG. 5A is a schematic overview of the classifier module;

FIG. 5B shows the stacked spectrogram of FIG. 4 being split and re-stacked for detection of EEG artefacts;

FIG. 6A is a schematic overview of a first classifier block of the module of FIG. 5A;

10 FIG. 6B is a schematic overview of an artefact detection module;

FIG. 7A shows the detailed architecture of the first classifier block of FIG 6;

FIG. 7B shows the detailed network architecture of the artefact detection module;

15 FIG. 8 shows a schematic overview of a second classifier block of the module of FIG. 5A;

FIG. 9 shows an exemplary recording comprising input signals over one epoch; assessed or determined sleep stages for epochs in the time period over which the recording has taken place, and epochs with confidence scores lower than a threshold confidence score as well as epochs which may have artefacts;

20

FIG. 10 is a model specification comprising weightings of the various features identifiable in a recorded epoch;

FIG. 11 and 12 are example of metadata accompanying the model specification of FIG. 10;

25

FIG. 13 illustrates the specification for a compressed feature set (CFS) file format;

FIG. 14 shows a schematic of a system for performing the method of FIG. 1; and

5 FIG. 15 shows an exemplary computing device suitable for executing the method of FIG. 1.

DETAILED DESCRIPTION

[12] Embodiments of the present invention will be described, by way of
10 example only, with reference to the drawings. Like reference numerals and characters in the drawings refer to like elements or equivalents.

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

[14] Unless specifically stated otherwise, and as apparent from the following, it will be appreciated that throughout the present specification,
25 discussions utilizing terms such as “identifying”, “adapting”, “scanning”,

“calculating”, “analysing”, “determining”, “replacing”, “generating”, “initializing”, “initiating”, “receiving”, “outputting”, or the like, refer to the action and processes of a computer system, or similar electronic device, that manipulates and transforms data represented as physical quantities within the computer system into other data similarly represented as physical quantities within the computer system or other information storage, transmission or display devices.

[15] The present specification also discloses apparatus for performing the operations of the methods. Such apparatus may be specially constructed for the required purposes, or may comprise a computer or other device selectively activated or reconfigured by a computer program stored in the computer. The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various machines may be used with programs in accordance with the teachings herein. Alternatively, the construction of more specialized apparatus to perform the required method steps may be appropriate. The structure of a computer will appear from the description below.

[16] In addition, the present specification also implicitly discloses a computer program, in that it would be apparent to the person skilled in the art that the individual steps of the method described herein may be put into effect by computer code. The computer program is not intended to be limited to any particular programming language and implementation thereof. It will be appreciated that a variety of programming languages and coding thereof may be used to implement the teachings of the disclosure contained herein. Moreover, the computer

program is not intended to be limited to any particular control flow. There are many other variants of the computer program, which can use different control flows without departing from the spirit or scope of the invention.

5 [17] Furthermore, one or more of the steps of the computer program may be performed in parallel rather than sequentially. Such a computer program may be stored on any computer readable medium. The computer readable medium may include storage devices such as magnetic or optical disks, memory chips, or other storage devices suitable for interfacing with
10 a computer. The computer readable medium may also include a hard-wired medium such as exemplified in the Internet system, or wireless medium such as exemplified in the GSM mobile telephone system. The computer program when loaded and on a general-purpose computer effectively results in an apparatus that implements the steps of the
15 preferred method.

[18] FIG. 1 illustrates a method 100 for automatically determining sleep stages, otherwise known as sleep staging. The method 100 broadly comprises:

Step 102: receiving input data;
20 Step 104: extracting one or more features for each sleep epoch;
Step 106: determining sleep stage probabilities one of which is an intermediate most probable class (IMPC);
Step 108: adapting sleep stage probabilities; and
Step 110: determining a most probable class, or final most
25 probably class (FMPC), of the respective sleep epoch.

[19] Receiving input data (102) comprises receiving one or both of electroencephalogram (EEG) measurements and electrooculography (EOG) measurements. Other forms of measurement may be taken, for example, using an electrocardiogram (ECG).

5 [20] The input data comprises a continuous stream of EEG data measured from a person. That data may come in the form of European Data Format (EDF), or another format as desired. This data is partitioned into sequence of sleep epochs, with each epoch comprising 30 seconds of continuous data. The sequence is continuous in that it comprises a
10 plurality of sleep epochs recorded end to end (i.e. when one epoch finished the next epoch commences), so that each instant in time is present in an epoch and, in some embodiments, is present in only one epoch. This enables the information derived or determined about one epoch (e.g. the sleep stage represented by that epoch) to be used to infer
15 or refine information about neighbouring epochs.

[21] As used herein, “neighbouring epochs” are epochs that appear adjacent in measured input data. For example, where the epochs comprise 30 second time intervals, an epoch measured from time $t = 15:32.30$ will be a neighbour to epochs measured from time $t = 15:32.00$
20 to $t = 15:33.00$. Similarly, a plurality of neighbouring epochs comprise epochs that appear sequentially in time – e.g. epochs recorded at time $t = 15:32.30$ (E_1), $15:33.00$ (E_2), $15:33.30$ (E_3), $15:34.00$ (E_4), $15:34.30$ (E_5) and so on. For epochs to be neighbours, they may collectively define a continuous period of time. In addition, E_1 is an earlier neighbouring epoch

of epochs E_2 , E_3 and so on. Similarly, E_n is a later neighbouring epoch of epochs E_{n-1} , E_{n-2} and so on.

[22] The sequence of epochs may comprise the entirety of the input data. The sequence may alternatively comprise a section of the input data represented by a continuous series of epochs. In either case, the data must be at least 30 seconds long so as to constitute one epoch. The present methods intend to reduce the reliance on human determination of sleep stages, by identifying the sleep stage of an epoch with reference to features of the epoch and those of surrounding epochs.

[23] Step 104 involves extracting one or more features for each sleep epoch and applying a weighting model to the feature of a plurality of sleep epochs to form hierarchies of concepts, by building up multiple layers of abstractions. Step 104 may involve pre-processing the sleep epochs to identify features in the sleep epochs as discussed with reference to FIG. 3.

Step 104 may further comprise conversion to compressed feature set (CFS) format. The pre-processing and conversion steps may be applied to the entire sequence, the entire input data set or even the entire recording of the period over which the subject is being monitored. The CFS format extracts a spectrogram comprising of compact set of time X frequency

domain features from each epoch of data. The CFS format packs the frequency domain features into a compressed binary file. This can reduce processing and data transport overhead in the subsequent steps. This can further enable real-time automated determination of sleep stages for each epoch.

[24] Where a CFS file has been used to transmit the feature set to a terminal for processing, step 104 involves unpacking the CFS file (i.e. at the terminal) to extract a sequence of features corresponding to each epoch. In any event, the features are forward passed through a weighting model comprising a plurality of weightings that are sequentially applied to the features for each epoch – see also FIG. 6A.

[25] The weightings are selected to emphasise some features and deemphasise others. Applying the weightings ensures that features, such as frequencies and amplitudes, that are known to be associated with particular sleep stages, are emphasised over those that are not (e.g. noise frequencies). In effect, the weightings can be selected to ensure at least one feature of each epoch is emphasised relative to at least one other feature of the respective epoch. Where a weighting is selected to emphasise a feature that is not present (e.g. is null), the weighting can nevertheless emphasise the zero feature relative to other features by deemphasising the other features.

[26] In some embodiments, the steps of passing forward the sequence of features into a weighting model and sequentially applying weightings comprises the following: the CFS file is unpacked to extract a sequence of features corresponding to each epoch and passing it through a trained convolutional neural network (CNN). CNNs are biologically inspired powerful machine learning models, especially suited to find visual patterns (see, for example, “Gradient-based Learning Applied to Document Recognition”, LeCun et al., Proceedings of the IEEE, 86(11):2278-2324, November 1998).

[27] The CNN model comprises multiple kernels stacked in layers as discussed with reference to FIG. 6A. Each kernel comprises many weightings, including a bias term, that are sensitive to specific spatiotemporal patterns. At the lowest level of the CNN (L1, 620), each
5 kernel is convolved with the spectrogram corresponding to each epoch. This extracts low-level features such as sharp changes in frequency and presence of certain frequencies across time, amongst others. The output of this layer is then fed forward to an activation layer (L2, 622), which rectifies the input by only allowing positive values to pass through. The
10 process of convolving and rectifying is repeated in subsequent layers. Higher-level layers build higher-level abstract patterns using lower-level features. Eventually, the highest-level kernels (L15, 616) may only be sensitive to specific sleep stages. The output of the final convolutional layer is then fed into a softmax activation layer (L16, 618), which
15 generates probabilities that a given epoch is associated with each sleep stage and identifies the sleep stage with the highest probability of being present in that epoch– the intermediate most probable class (IMPC) (step 106). In other words, the final convolutional layer decides, based on the output of the penultimate layer in the convolutional neural network, the
20 sleep stage the subject is most likely to have been in at the time of measurement of the epoch.

[28] The inventors recognise that the probability that a particular epoch represents a particular sleep stage is influenced by the neighbouring epochs (e.g. those immediately preceding or succeeding, in
25 the time domain, the epoch in question). For example, in a continuous

sequence of three epochs, if the last two epochs are representative of wake, then it is more likely that the third epoch either continues to be wake or transitions to stage 1 sleep than transitioning to REM sleep.

[29] Accordingly, step 108 involves adapting the sleep stage

5 probabilities of each sleep epoch based on one or more sleep stage probabilities of a neighbouring sleep epoch, or multiple neighbouring sleep epochs. This adaptation step ensures that a determination of the sleep stage of a particular epoch can be affected by the results of a similar determination made in respect of its neighbours.

10 [30] In some cases, the sleep stage probability of a single neighbouring epoch may be used to adapt the probabilities of the epoch in question. In other cases, the sleep stage probability of the epoch may be adapted using one or more sleep stage probabilities of at least one earlier neighbouring epoch and at least one later neighbouring epoch. In other words, both

15 earlier and later epochs can be indicative of the sleep stage of the epoch in question. In some cases, for epoch E_n , the five immediately preceding or earlier neighbours E_{n-5} , E_{n-4} , E_{n-3} , E_{n-2} and E_{n-1} and the five immediately succeeding or later neighbours E_{n+1} , E_{n+2} , E_{n+3} , E_{n+4} and E_{n+5} may be used. Alternatively, the number of earlier and later neighbours may depend on

20 the rate at which the subject, or subject's with similar characteristics (e.g. age or gender), moves between sleep stages. For example, where a subject transitions from sleep stage 1 to sleep stage 2 progressively over a 120 second time period, the four immediately earlier neighbours and four immediately later neighbours – each of which represent a 120 second

time interval – may be used to clarify the classification of the sleep stage of the epoch in question.

[31] Those sleep stages that are more immediate neighbours than others (e.g. the sleep stage immediately preceding the sleep stage in question is a more immediate neighbour than any earlier sleep stage) may be more important in determining sleep stage of the epoch in question. The adapting step therefore involves applying weights to the neighbouring epochs to affect the degree to which the sleep stages of those epochs influence a determination made of the sleep stage of the epoch in question.

[32] Adapting the sleep stage probabilities of one epoch, using the sleep stage probabilities of neighbouring epochs, includes determining, from the weighting model or CNN outputs of one or more neighbouring epochs, an intermediate most probable class (IMPC), the IMPC being the sleep stage that has the highest probability of being the true stage of a particular epoch. The IMPCs of the neighbouring epochs along with the sleep stage probabilities of the current epoch constitute a feature vector, which is fed into classification module 2 (700, Figure 7) comprising a trained multi-layer perceptron (MLP). The MLP constructs a non-linear mapping between the feature vector and adapted sleep probabilities. For example, passing a weighted input (i.e. the input after the abovementioned weighting has been applied) through a tansig function results in a non-linear mapping between the feature vector and the adapted sleep probabilities. The sleep stage probabilities of the sleep epoch are thus adapted to take into account the sleep stage probabilities

and determined sleep stage of the neighbouring epoch(s). This produces a set of adapted probabilities (i.e. adapted sleep stage probabilities). The epoch in question is then determined to represent the sleep stage with the highest adapted probability. In other words, the sleep stage with the highest adapted probability is the final most probable class (FMPC) of the respective sleep epoch.

[33] In some cases the probability that a particular epoch is representative of a particular sleep stage may be low even after steps 102, 104, 106, 108 and 110. In addition, two sleep stages may have similar probabilities or likelihoods of being represented by the epoch.

[34] The method 100 may therefore include an assessment of the confidence that a particular determination of sleep stage is accurate. To that end, the method 100 comprises the further steps of computing a confidence score based on sleep stage probabilities as obtained by the CNN and/ or MLP (step 112). If a minimum confidence level is specified, then all epochs with confidence lower than this threshold are marked for review (step 116). Otherwise, an epoch with confidence equal to or higher than the threshold is accepted as being correctly assessed (step 114).

[35] Optionally, the method 100 may include detection of EEG artefacts. Sources of EEG artefacts include electrode displacement, motion of a subject, EMG and ocular activity as well as electromagnetic noise from nearby power lines and electronic devices. These artefacts may complicate power analysis that may be performed on the EEG data post sleep staging. Artefacts can also affect sleep staging by obscuring physiological signals. Step 118 involves determining artefact probabilities

based on: (i) the extracted feature(s) for each sleep epoch (obtained from step 104); and (ii) the most probable class (MPC), or final most probably class (FMPC), of the sleep epoch (obtained from step 110).

[36] FIG. 2 is a schematic overview of a system 200 for performing the method of FIG. 1. The system 200 may be a distributed system comprising multiple devices, which may be co-located or may be located remotely from one another. The system 200 may also be embodied in a single device.

[37] To perform the method of FIG. 1 on a sleep recording, the system 200 accepts input from one or more sensors. In the embodiment shown in FIG. 2, the system 200 accepts input from:

one or both electroencephalogram (EEG) channels: C4-A1 and C3-A2; and

two electrooculography (EOG) channels: E2-A1 and E1-A2.

[38] These inputs (i.e. recording electrodes) are attached to a human subject 202 and generate digitally sampled signals for each channel. The data is usually sampled at 256 Hz or more. The data may then be stored in EDF format in a storage device. This data is then partitioned into 30 second epochs of sleep and classified into one of the following five stages: Wake, Stage 1, Stage 2, Stage 3/4 and REM. The inputs define a continuous period of recording as discussed in relation to step 102.

[39] The system 200 may alternatively, or in addition, accept data from memory 204 and use that data to either analyse the subject's data offline, or to train the weighting model used in the method of FIG. 1. The data

stored in memory 204 may be in the form of European Data Format (EDF) files or another file format.

[40] The data received from all sensors, or from memory 204, is normalised. For example, where the expected data should be in μV , measurements that are taken in a unit of measurement other than μV are converted to μV .

[41] Both the electrodes attached to the human subject 202 and the memory 204 form part of the client side 206 of system 200 – in other words, the part of the system 200 that provides inputs to the server side 208 of the system and/or makes requests for processing or information from the server side 208.

[42] After any necessary data normalisation, the data is sent to a pre-processing module 210. The pre-processing module 210 may process the input data to reduce the amount of that data. By strategically reducing the amount of data, the time taken to process that data can be significantly reduced, as can the storage and transmission overheads required to store and transmit that data. The pre-processor 210 achieves this by combining certain channels together and transforming the time domain signal to frequency X time features. The pre-processor 210, then packs these features into a binary file as per the compressed feature set (CFS) file specifications.

[43] After pre-processing and conversion by the pre-processor 210, the data is sent to the server 208. The data is sent over a data transport layer 212. The pre-processor 210 and server 208 may comprise parts of the

same device or system, or the server 208 may be located remotely from the pre-processor 210.

[44] The server side 208 performs automatic sleep scoring (i.e. determining sleep stages for each epoch) and sends the sleep scores along with corresponding confidence values back to the client side 206. The client side 206 then displays the results to the end user on a display.

[45] FIG. 3 shows the architecture of the pre-processor module 300 corresponding to pre-processor 210 of FIG. 2. The pre-processor module 300 receives data from one or more input channels. In the present instance, the input channels comprise:

- 2 electroencephalogram (EEG) channels (C4-A1 and C3-A2) 302; and
- two electrooculogram (EOG) channels (E2-A1 and E1-A2) 304

[46] Within the pre-processor module, the EEG data from channels C4-A1 and C3-A2 is averaged (at 306) to construct one single EEG channel. Additional EEG or measurements channels may also be used and the overall average taken. Alternatively, a single measurement channel may be used in which case the averaging process is not required. In some cases, where a measurement channel (e.g. an EEG sensor) has malfunctioned or is otherwise not delivering useable data, the data from the other channel may similarly be sent for filtering without first averaging the data.

[47] The averaged data, along with the EOG data, is filtered by filter 308, to remove noise.

- [48] Filter 308 is a window-based filter. The window-based filter may use a finite impulse response (FIR) or an infinite impulse response (IIR) band-pass filter, for example an order 50 FIR band-pass filter or an order 2 IIR filter.
- 5 [49] Computation of filter weights is done using a Hamming window or other windowing function such as a Hanning or Blackman window. The pass-band frequency of the window may be 0.1 to 60 Hz for EEG channels, but is more preferably 0.3 to 45 Hz. Similarly, the pass-band frequency of the window may be 0.1 to 20 Hz for EOG channels, but is more preferably
- 10 0.3 to 12 Hz.
- [50] After filtering, the filtered data is then resampled at resampler 310. The resampling process generates a data set of the required resolution to produce spectrograms each of which represents an epoch in the period over which the subject is monitored.
- 15 [51] The resampling occurs at 100 Hz and employs a combination of interpolation and decimation (for details refer to Rabiner et al., "Interpolation and decimation of digital signals—A tutorial review." Proceedings of the IEEE 69.3 (1981): 300-331).
- [52] The resampling process employs windowed FIR or IIR filters to
- 20 avoid aliasing. The order of the FIR filters can be selected to suit the particular application. Higher-order filters will result in better filter performance but increase the processing time. For real-time sleep stage determination, the processing time is critical. Conversely, the lower the order of the filter the poorer the data filtration. The present FIR filters

may be of order 10-50, or alternatively 20 to 40, but are most preferably of order 30 to 35.

[53] If the original data is sampled at 100Hz, then the resampling step is skipped.

5 [54] After the resampling process the data is divided into epochs. The epochs may be 30 second each. Thus each epoch, if sampled at 100 Hz, will comprise exactly 3,000 samples per input channel.

[55] A spectrogram is then generated for every epoch of data (at 312). Each spectrogram comprises a time (X-dimension) frequency (Y-
10 dimension) decomposition of the original data for a particular channel. The spectrogram is obtained using a Fourier transform, preferably a short-time Fourier transform. For example, a Hamming window of length 128 with overlap of 29.69% may be used. A Fourier transform can then be applied using the Fast Fourier Transform (FFT) algorithm.

15 [56] As a result of the FFT, spectrograms are produced each having a first number (A) of time points and a second number (B) of frequency points. Using the Hamming window set out above, there will be 32 time points (resolution of 938 ms) and 65 frequency points (resolution of 0.7692 Hz) for each spectrogram.

20 [57] To produce square spectrograms, the initial "a" points of the (B) frequency points, or frequency bins, are considered. This corresponds to frequencies of 0 to 24.6154 Hz. The result is a square spectrogram. Presently, since A = 32, the resulting spectrogram is 32X32.

[58] The spectrogram for each channel is then stacked at 314. The stack
25 314 forms a tensor of size equal to the spectrogram size and depth equal

to the number of channels received at the filter 308 (e.g. three channels comprising the averaged EEG data and the two EOG channels). The stacked spectrogram is therefore 32 X 32 X 3.

[59] The stacked spectrograms are then converted into CFS file format, and are then sent to the server side for scoring by classification modules 1 and/or 2.

[60] The pre-processor module 300 also receives scored data from the server-side – in other words, sleep scores and their associated confidence values. The scored data is then presented to the end user.

[61] FIG. 4 shows such a stacked spectrogram (feature set) 400, comprising the 32 X 32 spectrograms 402, 404, 406 from each channel. Thus one stacked spectrogram is produced for each epoch.

[62] With further reference to FIG. 3, data from all available epochs are compressed (e.g. converted to CFS format) at compressor converter 316 before being sent to the server-side, either locally or over a data transfer layer. For data sent over the data transfer layer to a remote server, the data transfer may use TCP/IP. Notably, by providing sleep staging on a remote server, and by sending a reduced or compressed data file (e.g. the CFS file) instead of the original data to the server for classification, the EEG measurements can be further processed using a less computationally powerful device. The bandwidth requirement for the transfer of data to the server can also be significantly reduced. This facilitates use of sleep staging equipment in-house or in smaller clinics.

[63] FIG. 5A shows an exemplary server module 500 to which the encoded, stacked spectrograms in the compressed form (e.g. of a CFS file)

is sent from the pre-processor module of FIG. 3. The server module 500 consists of a decoder block 502 for decoding the stacked spectrograms 504. The server module 500 further comprises two classification stages represented by classification modules 506, 508.

5 [64] The first classification module 506 determines a probability, for each of several sleep stages, that a particular epoch is representative of the respective sleep stage. The first classification module 506 may determine a probability for each of the five sleep stages (i.e. five probabilities, including a probability the subject is in the “wake” stage).

10 [65] After determining the probabilities, the first classification module 506 then identifies the intermediate most probable class (IMPC), being the sleep stage with the highest probability.

15 [66] The second classifier module 508 receives the probabilities, and IMPC where provided, from the first classifier module 506. The second classifier module then adapts the sleep stage probabilities of each sleep epoch based on one or more sleep stage probabilities of a neighbouring epoch or the sleep stage probabilities of multiple neighbouring epochs. This enables neighbouring epochs to affect the classification of the sleep stage represented by an epoch in question. The effect is that classification
20 of ambiguous epochs – for example, epochs where the IMPC is only marginally more probable than that for the sleep stage with the second highest probability, or epochs representing a transition between sleep stages – can be clarified using the classifications of neighbouring epochs. Where the neighbouring epochs are of a common sleep stage, but the
25 epoch in question is ambiguous when considered in isolation of its

neighbours, then it can be said with reasonable confidence the subject is more likely to be in the sleep stage of the neighbouring epochs than another sleep stage. Similarly, if a transition between sleep stages occurs during an epoch, two sleep stages may be similarly likely.

5 [67] As well as classification module 1, the second classifier module also determines the final most probable class (FMPC) of a particular sleep epoch based, on the adapted probabilities.

[68] The module 500 may further include an artefact detection module 512 that is configured to identify a presence of EEG artefacts. Artefact
10 detection is applied to the EEG data and on a smaller time-scale of 5 second epochs (in contrast to 30 second epochs for sleep staging). Accordingly, with reference to FIG. 5B, the stacked spectrogram 400 described above in relation to FIG. 4, comprising the 32 X 32 X 3 spectrograms from the EEG channel 402, is collected and each
15 spectrogram is temporally split into six subsections of 5 seconds each. For illustration, spectrogram 402' shows original spectrogram 402 being temporally split into six subsections of 5 seconds each. Subsequently, these six subsections are stacked together by re-stack module 510. For illustration, stacked spectrogram 402'' shows the six subsections of split
20 spectrogram 402' being stacked together. These stacks are then weighed along with their corresponding predicted sleep stage using another deep-convolutional neural network of the artefact detection module 512. Since staging is preferably performed using epochs of 30 seconds, all six subsections corresponding to a particular epoch are assumed to have the
25 same sleep stage. The artefact detection module 512 determines a

probability of a sub-section having an artefact. An artefact probability threshold may be set, e.g. at 0.5, such that sub-sections with artefact probability ≥ 0.5 are marked as “artefact” while sub-sections with artefact probability < 0.5 are marked as “not-artefact”.

5 [69] FIG. 6A illustrates an embodiment 600 of the first classifier block (classification module 1) 506 of FIG. 5A. The first classifier module 600 comprises a neural network, for example a multi-layer deep convolutional neural network (CNN). In the neural network the input is progressively convolved by each node 602, 604, 606, 608, 610, 612, 614, 616 in the
10 network. The final node 618 generates probabilities for different sleep stages. Module 624 then computes the class or sleep stage associated with the sleep stage with maximum probability (IMPC). This class or sleep stage is the IMPC, of the sleep epoch to which the CNN was applied.

[70] The neural network of the classification module 1, 600 comprises
15 9 nodes. Seven of the nodes (602, 604, 606, 608, 610, 612 and 614) comprise two layers and two of the nodes (616, 618) comprise single layers, making a 16-layer deep neural network. Each two-layer node comprises a convolution layer and an activation layer. As required of feed forward networks, the convolution operation is applied on the data from
20 a previous layer. The output of the convolution layer is fed into the activation function layer. For example, for node 602, the convolution layer is marked 620 and the activation layer is marked 622.

[71] Each convolution layer may comprise multiple convolution filters. Each convolution filter treats an input tensor – being the input
25 spectrographs for node 602 and, for the other nodes, the output data from

the previous layer – using a kernel tensor designed to emphasise a particular pattern in the input tensor.

[72] The multiple convolutional filters may be followed by an activation function layer. The span of each filter is known as its local receptive field.

5 Each filter responds to specific local patterns within its receptive field. Filters in the higher layers then combine these low-level patterns to construct more abstract patterns.

[73] The local receptive field for each filter is moved across the input in steps of δ , also known as stride length. If the input data I is of size $l \times l$ with n channels and each filter $\mathcal{F}$ is of size $d \times d \times n$, then convolution with a stride length of δ will result in an output image O of size $k \times k$, where $k = \frac{l-d}{\delta} + 1$. The stride length δ may be less than filter dimension d , thus resulting in overlapping filter spans.

10

[74] The value of output at location i, j is thus given as:

$$O(i, j) = \sum_{\gamma=1}^n \sum_{\beta=1}^d \sum_{\alpha=1}^d I(i\delta + \alpha - 1, j\delta + \beta - 1, \gamma) \mathcal{F}(\alpha, \beta, \gamma) + b; \quad i, j = 1, \dots, k$$

15 [75] In the above equation (Equation (1)) b is a bias term.

[76] The output of the filter is then sent to an activation function layer (e.g. layer 622 of node 602). If the activation function is θ , then the output from the activation layer will be

$$O'(i, j) = \theta(O(i, j))$$

[77] Each filter has the same weights and bias as it moves across the data. Therefore, each convolution layer with ω convolution filters of size $d \times d \times n$ will have $(d^2n + 1) * \omega$ weights – thus, the filter will have a

20

weight for each value in its local receptive field, or span, plus an additional bias term value b .

[78] Since the filter applies the same weights and bias to each local receptive field, the convolution computations can be massively parallelized. This is because the output of one convolution filter is determined on the same input data but is otherwise independent of the output of other convolution filters. Accordingly, the parallelised application of convolution filters can be performed on general-purpose computing on graphics processing units (GPU).

[79] The activation layer of all nodes, except the last node, comprises a rectified linear unit (RELU). For the last convolution layer, a softmax activation function is used. The RELU function is defined as:

$$f(x_i) = \max(0, x_i)$$

and the softmax function is defined as:

$$f(x)_i = \frac{e^{x_i}}{\sum_{n=1}^N e^{x_n}}; n = 1, \dots, N,$$

where, x_i is the i^{th} output from previous layer with total of N outputs.

[80] The softmax function takes an input matrix or vector of potentially arbitrary values and normalises them so that the output becomes a matrix or vector of the same dimensionality as the input matrix or vector, comprising real values that add up to a desired number. As the softmax function is intended to identify the probability that an epoch is representative of a particular sleep stage, and all probabilities should sum to 1, the values generated by the softmax function add up to 1.

[81] The architecture of the network is shown in FIG. 6A, with the final layer or node 618 of the network 600 generating class probabilities, or sleep stage probabilities, for each sleep stage.

[82] The probabilities are fed into a most probable class (MPC) unit
5 624, which computes the class with highest probability (IMPC):

$$c = \operatorname{argmax}(p^i); i = 1, \dots, 5$$

[83] Where the five sleep stages are being used for assessment of sleep, the probabilities for the five possible sleep stages along with the IMPC constitute the output of classification module 1. Using the filter resolutions (i.e. spans or local receptive fields) specified in table 700,
10 along with the relevant number of filters and stride length, the CNN, 700 has a total of 177,669 weights. The weights of the CNN are obtained by training the network on a large amount of annotated historical data from previous sleep recordings. To start off the training process, weights for each kernel are initialized randomly using a method attributed to Glorot
15 & Bengio (Glorot, Xavier, and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." In International conference on artificial intelligence and statistics, pp. 249-256. 2010). Batches of data, comprising 200 to 400 epochs from a randomly shuffled collection of historical data, are fed forward through the CNN. The
20 discrepancy between the predicted and true sleep stages is obtained using a categorical cross-entropy loss function. Using a method known as stochastic gradient descent with momentum attributed to Rumelhart et al. (Rumelhart, David E.; Hinton, Geoffrey E.; Williams, Ronald J. "Learning representations by back-propagating errors". Nature 1986, 323 (6088):

533–536) the weights are adjusted so as to reduce the discrepancy between predicted and true classification, as measured by the loss function. Training is carried out until the performance of the CNN saturates or starts deteriorating for an independent unseen dataset.

5 [84] When the data is being processed offline (i.e. no real-time results are needed), the block 1 classifier 600 of FIG. 6A can be run on all available input data (i.e. the entire period over which the subject has been monitored) before the processed data is sent to the second block classifier 800 shown in FIG. 8.

10 [85] FIG. 6B is a schematic overview of the artefact detection module 512, according to an example embodiment. The artefact detection module comprises a deep convolutional neural network (CNN). In the neural network, the input is progressively convolved by each node in the network.

15 [86] FIG. 7B shows an embodiment of the detailed network architecture of the artefact detection module 512.

[87] FIG. 9 shows a sample offline data scored by the system described herein. The raw signals 902 from a particular epoch (the particular epoch being identified by 904) along with the sleep classifications 910 are visualized. The stage classifications 910 are shown on the top with epochs having low relative confidence marked (e.g. 908). Artefacts are indicated by marking 909 on the hypnogram. The current embodiment generates sleep stages in portable formats such as comma separated value (CSV), javascript object notation (JSON), extensible markup language (XML) or
25 EDF format and is independent of any particular visualization software.

[88] For online processing, such as in applications requiring real-time determination of sleep stages, the first block classifier 600 may classify a particular epoch and one or more neighbouring epochs before the second block classifier 800 can commence processing the particular epoch using the outputs from the first stage classifier 600 for both the particular epoch and the one or more neighbouring epochs.

[89] In some embodiments, at least 5 neighbouring epochs of data are first processed by block 1 classifier before the second classifier 800 commences processing a particular epoch. For example, for any particular epoch, the output of block 1 classifier 600 may be fed in to block 2 classifier 800, along with the block 1 classifier IMPC outputs for five preceding and five succeeding epochs. In the online mode, when succeeding IMPC outputs are not available, the five preceding IMPC outputs are re-used. Similarly, where the five preceding IMPC outputs are not available (e.g. at the start of a sleep cycle or period over which a subject is monitored) the five succeeding IMPC outputs are re-used. By "re-use" it is intended that the relevant succeeding or preceding IMPC outputs are assumed to be reflective of the IMPC of the absent or unavailable epochs. The IMPC values for the re-used epochs may be mirrored around the epoch in question. As an example, for an epoch E_n having no succeeding epochs, the IMPCs of epochs E_{n-1} , E_{n-2} , E_{n-3} , E_{n-4} and E_{n-5} may be assumed to be representative of the IMPCs of what would otherwise have been epochs E_{n+1} , E_{n+2} , E_{n+3} , E_{n+4} and E_{n+5} . Where the IMPCs are mirrored, the IMPC for epoch E_{n-1} will be assumed to be the IMPC for epoch E_{n+1} , the IMPC for epoch E_{n-2} will be assumed to be the

IMPC for epoch E_{n+2} and so on. Moreover, where epochs E_{n-5} , E_{n-4} , E_{n-3} , E_{n-2} , E_{n-1} , E_n , E_{n+1} and E_{n+2} are present, then the IMPCs for epochs E_{n-5} , E_{n-4} , E_{n-3} may be assumed to be the IMPCs for epochs E_{n+3} , E_{n+4} , E_{n+5} – in other words, succeeding and preceding IMPCs can be used to fill in data around an epoch in question, where that data is otherwise unavailable in the input data.

[90] Classification module 2 800, shown in FIG. 8, consists of a multi-layer perceptron (MLP) with a plurality of hidden units. In the present embodiment, the IMPCs of five preceding and five succeeding epochs are used to adapt the IMPC of the epoch in question, by adapting the probabilities that the epoch in question is representative of any particular sleep stage. Accordingly, the module 800 receives as its input (802) the IMPCs of the five preceding and succeeding epochs, the IMPC of the epoch in question and the sleep stage probabilities of the epoch in question. These inputs (802) form input nodes 804 of the block classifier 800.

[91] Each input is mapped to each hidden units or nodes (806), of which there are presently 20. The hidden units 806 enable processes to be performed on the data that cannot be performed other than in sequence – i.e. there is no function directly mapping the input to the output but instead a plurality of sequential operations must be performed in order to map the input to the output.

[92] The hidden layer (i.e. the collection of nodes 806) is fully connected with the input layer (comprising nodes 804) – in other words, every node 804 forms an input for every node 806. The hidden layer is also fully connected with the output layer 808 – in other words, every

node 806 forms an input for every node 808. Each node 806 in the hidden layer comprises two functions 810, 812. The function 810 computes a weighted sum of data, along with a bias and sends it to an activation function 812. The activation layer 812 for the hidden unit 806 is a
 5 hyperbolic tangent sigmoid (tansig) function defined as:

$$f(x) = \frac{2}{e^{-2x} + 1} - 1$$

[93] This function is used in place of Tanh, the application of which will be understood by the skilled person, due to the faster computation afforded by the tansig function as realised by the inventors. Finally, the nodes of the output layer 808 create a weighted sum of the results from
 10 hidden layer nodes 806. Once computed, the weighted sums from the respective output layer nodes 808 are sent to a softmax activation function 814, which generates class probabilities by normalising the outputs from the output layer nodes 808 such that their total sums to 1. A MPC block 816 is used to identify, for each epoch, the sleep stage that has
 15 the highest probability of being associated with that epoch. This sleep stage class is the final most probable class (FMPC).

[94] While the process may stop at this point, the present second classifier block 800 provides the further processing step of determining a confidence score r . The confidence score r indicates how confident the
 20 computing system is in the result (i.e. the determined sleep stage for a particular epoch) it has obtained after the softmax function 814 has been applied. The confidence score can be used to distinguish epochs that are more likely to have been correctly assessed – in other words, those

epochs for which the determined sleep stage is likely to be correct – from those epochs for which the determined sleep stage may be incorrect.

[95] The relative confidence score r is computed as:

$$r = \min\left(\frac{p^{max}}{p^{secondmax}} - 1, 10\right),$$

where, p^{max} is the probability of most probable class and $p^{secondmax}$ is the probability of second most probable class. The score varies between 0 and 10, with 0 signifying very low confidence and 10 signifying very high confidence.

[96] The confidence score calculated for any particular epoch is compared to a threshold confidence score (if provided) to determine whether the confidence score for the epoch is sufficiently high to accept the sleep stage determination made in respect of that epoch. If the confidence score for a particular epoch is at least as high as the threshold confidence score, then the determination in respect of that epoch is accepted. If the confidence score for a particular epoch is below the threshold confidence score, then that particular epoch is compared to historical data to determine the sleep stage most closely approximated by the epoch. The sleep stage most closely approximated by that epoch is then assumed to be the sleep stage represented by the epoch. In some cases, the epoch is marked for review by an expert sleep scorer.

[97] By thresholding this confidence score, a portion of the overall data can be marked for review by, for example, a remote system using different analytical methods. The remote system may be substituted for an expert scorer.

[98] Using a confidence threshold, higher accuracy can theoretically be achieved at minimal cost. For example, by raising the threshold confidence score the number of sleep stage determinations that will be marked for review (i.e. assessment against historical data) increases, but the likely accuracy of the determinations that are accepted is also higher. Conversely, by lowering the threshold confidence score the number of sleep stage determinations that will be marked for review, and the likely accuracy of the determinations that are accepted, similarly decreases.

[99] The classified sleep stage along with associated relative confidence for each epoch is sent to the client in any of the open data interchange formats like comma separated value (CSV), Extensible Markup Language (XML), JavaScript Object Notation (JSON) or EDF amongst others (718).

[100] The multi-layer perceptron (MLP) with 16 inputs and 1 bias fully connected to 20 hidden nodes has $(16+1) \times 20 = 340$ weights in the input side. The 20 hidden nodes along with a bias fully connected with 5 output nodes has $(20+1) \times 5 = 105$ weights on the output side. The network therefore has a total of 445 trainable weights. The MLP classification module 2, 800 is trained only after the CNN classification module 1, 600 is trained using historic data. The training process for 800 is similar to that of 600. The overall classification modules of the present embodiments thus have a total of $177,669 + 445 = 178,114$ trainable weights. These trained weights, when laid out as a long column vector, constitute the model specification 1000 as shown in FIG. 10. The model specification 1000 is that used to provide the model 1002 of the first classification module and the model 1004 of the multi-layer perceptron (the second

classification module). The model specification 1000 may have weights for the artefact detection module and can provide the model 1005 of the artefact detection module.

[101] Additional metadata 1006 can be appended to the model specification. The metadata 1006 may hold information pertaining to the training process or the manner in which the weights are initialised. The metadata may also comprise information that enables a plurality of models to be used, where each model is tailored for a subject having particular characteristics – for example, age, weight and gender.

[102] FIG. 11 shows sample metadata 1100 for accompanying a particular model specification of FIG. 10. The metadata 1100 shows demographic data on which the model was trained, the expected accuracy and recommended relative threshold levels. The metadata 1100 includes the size 1102 of the dataset used for training (648,451 epochs $\approx$ 5400 hours); demographics 1104 of the subjects; stage-wise classification accuracy on training, testing and validation sets 1112, 1108, 1110; overall stage-wise accuracy 1106, and accuracy for different levels of confidence thresholds and corresponding amount of data that is marked for review 1112. The graph 1114 shows performance in terms of receiver operating characteristic (ROC) for the artefact detection module. The graph 1212 in FIG. 12, shows that an increase in confidence threshold increases the accuracy of the outputs. However, the increased accuracy is accompanied by an increased number of epoch classifications sent for further review.

[103] As mentioned above, multiple different models may be developed depending on demographics and characteristics of the subject. An end

user of the system (e.g. a sleep technician, physician or, for in-home monitoring, the user) can select the most suitable model based on its specifications' metadata, which can be sent over to the client side.

[104] For interoperability and ease of transport of processed data from client to the server side, a compressed feature set (CFS) format is specified (FIG. 13). The three-dimensional spectrogram data is vectorized by reading the data column-wise (each column corresponds to one time-point) starting from the first channel. Data from all available epochs are concatenated into a single very long column vector. The spectrogram data is stored in single precision 32-bit floating point number as per IEEE-754 standard. The 20 byte long cryptographic hash of this data is obtained using secure hash algorithm-1 (SHA-1). This hash uniquely identifies the spectrogram data without relying on any subject identifiable markers. Additionally, the hash ensures the consistency of spectrogram data as it is sent across the transport channel from client to server. The raw data stream is passed through deflate compression algorithm, such as for example is described in RFC-1951 specifications (The Internet Engineering Task Force, Request for Comments – document number 1951, published in 1996, the entire contents of which is incorporated herein by reference).

[105] This compressed stream constitutes the data stream for the CFS format. The first 11 bytes constitute the header of the file, off which first 3 bytes carry the signature for the file. The signature in HEX is 43, 46 and 53 which reads as 'CFS' in ASCII. The next 1 byte carries the file version number. The following 5 bytes carry the dimension of the spectrogram in

frequency (1 byte) X time (1 byte) X channel (1 byte) X epochs (2 bytes)
format. The last 2 bytes set compression mode and hash set binary flags.
When the compression mode is set to 0 the data stream is not
compressed. The hash set byte when set to 0 indicates that the SHA-1
hash is not computed and is not included in the file. Under normal
conditions both these bytes will be set. But for online processing, when
only one epoch of data is sent to the server at a time, the last 2 bytes in
the header can be set to 0 and data can be sent without compression and
hash. This reduces both computational and data overhead.

[106] FIG. 14 shows a schematic of a network-based system 1400 for
automatically determining sleep stages according to an embodiment of
the invention. The system 1400 comprises a computer 1402, one or more
databases 1404a...1404n, a user input module 1406 and a user output
module 1408. Each of the one or more databases 1404a...1404n are
communicatively coupled with the computer 1402. The user input
module 1406 and a user output module 1408 may be separate and
distinct modules communicatively coupled with the computer 1402.
Alternatively, the user input module 1406 and a user output module 1408
may be integrated within a single mobile electronic device (e.g. a mobile
phone, a tablet computer, etc.). The mobile electronic device may have
appropriate communication modules for wireless communication with
the computer 1402 via existing communication protocols.

[107] The computer 1402 may comprise: at least one processor; and at
least one memory including computer program code; the at least one
memory and the computer program code configured to, with at least one

processor, cause the computer at least to: (A) receive input data representing a continuous stream of data measured from a subject and representative of a continuous sequence of sleep epochs; (B) extract one or more features for each of a plurality of sleep epochs and apply a plurality of weightings for emphasising at least one feature of each sleep epoch of the plurality of sleep epochs; (C) determine, from the weighted features of the respective sleep epoch, a sleep stage probability for each of a plurality of sleep stages, each sleep stage probability defining a likelihood the human subject was in the respective sleep stage at the time the respective sleep epoch was measured; (D) adapt the sleep stage probabilities of each sleep epoch based on one or more sleep stage probabilities of at least one neighbouring sleep epoch in the plurality of sleep epochs; and (E) determine the final most probable class (FMPC) of the respective sleep epoch, the FMPC being the sleep stage with the highest adapted probability.

[108] The computer program code may be configured to, with the at least one processor, cause the computer to further (F) determine an intermediate most probable class (IMPC) for the respective sleep epoch from the output of C, the IMPC being the sleep stage with highest sleep stage probability. In relation to step (F), step (D) may further involve adapting the sleep stage probabilities of the respective sleep epoch based on the IMPC of at least one respective neighbouring sleep epoch in the plurality of sleep epochs. Step (E) may further involve passing the IMPC of the at least one neighbouring sleep epoch, along with the IMPC of the current epoch as well as the class probabilities, to a MLP.

[109] FIG. 15 depicts an exemplary computer / computing device 1500, hereinafter interchangeably referred to as a computer system 1500, where one or more such computing devices 1500 may be used to facilitate execution of the above-described method automatically determining sleep stages. In addition, one or more components of the computer system 1500 may be used to realize the computer 1402. The following description of the computing device 1500 is provided by way of example only and is not intended to be limiting.

[110] As shown in FIG. 15, the example computing device 1500 includes a processor 1504 for executing software routines. Although a single processor is shown for the sake of clarity, the computing device 1500 may also include a multi-processor system. The processor 1504 is connected to a communication infrastructure 1506 for communication with other components of the computing device 1500. The communication infrastructure 1506 may include, for example, a communications bus, cross-bar or network.

[111] The computing device 1500 further includes a main memory 1508, such as a random access memory (RAM), and a secondary memory 1510. The secondary memory 1510 may include, for example, a storage drive 1512, which may be a hard disk drive, a solid state drive or a hybrid drive and/or a removable storage drive 1514, which may include a magnetic tape drive, an optical disk drive, a solid state storage drive (such as a USB flash drive, a flash memory device, a solid state drive or a memory card), or the like. The removable storage drive 1514 reads from and/or writes to a removable storage medium 1544 in a well-known manner. The

removable storage medium 1544 may include magnetic tape, optical disk, non-volatile memory storage medium, or the like, which is read by and written to by removable storage drive 1514. As will be appreciated by persons skilled in the relevant art(s), the removable storage medium
5 1544 includes a computer readable storage medium having stored therein computer executable program code instructions and/or data.

[112] In an alternative implementation, the secondary memory 1510 may additionally or alternatively include other similar means for allowing computer programs or other instructions to be loaded into the computing
10 device 1500. Such means can include, for example, a removable storage unit 1522 and an interface 1540. Examples of a removable storage unit 1522 and interface 1540 include a program cartridge and cartridge interface (such as that found in video game console devices), a removable memory chip (such as an EPROM or PROM) and associated socket, a
15 removable solid state storage drive (such as a USB flash drive, a flash memory device, a solid state drive or a memory card), and other removable storage units 1522 and interfaces 1540 which allow software and data to be transferred from the removable storage unit 1522 to the computer system 1500.

[113] The computing device 1500 also includes at least one communication interface 1524. The communication interface 1524 allows software and data to be transferred between computing device 1500 and external devices via a communication path 1526. In various embodiments of the inventions, the communication interface 1524 permits data to be
20 transferred between the computing device 1500 and a data
25

communication network, such as a public data or private data communication network. The communication interface 1524 may be used to exchange data between different computing devices 1500 where such computing devices 1500 form part of an interconnected computer network. Examples of a communication interface 1524 can include a modem, a network interface (such as an Ethernet card), a communication port (such as a serial, parallel, printer, GPIB, IEEE 1393, RJ45, USB), an antenna with associated circuitry and the like. The communication interface 1524 may be wired or may be wireless. Software and data transferred via the communication interface 1524 are in the form of signals which can be electronic, electromagnetic, optical or other signals capable of being received by communication interface 1524. These signals are provided to the communication interface via the communication path 1526.

[114] As shown in FIG. 15, the computing device 1500 further includes a display interface 1502 which performs operations for rendering images to an associated display 1530 and an audio interface 1532 for performing operations for playing audio content via associated speaker(s) 1534.

[115] As used herein, the term "computer program product" may refer, in part, to removable storage medium 1544, removable storage unit 1522, a hard disk installed in storage drive 1512, or a carrier wave carrying software over communication path 1526 (wireless link or cable) to communication interface 1524. Computer readable storage media refers to any non-transitory, non-volatile tangible storage medium that provides recorded instructions and/or data to the computing device 1500 for

execution and/or processing. Examples of such storage media include magnetic tape, CD-ROM, DVD, Blu-ray™ Disc, a hard disk drive, a ROM or integrated circuit, a solid state storage drive (such as a USB flash drive, a flash memory device, a solid state drive or a memory card), a hybrid drive, a magneto-optical disk, or a computer readable card such as a SD card and the like, whether or not such devices are internal or external of the computing device 1500. Examples of transitory or non-tangible computer readable transmission media that may also participate in the provision of software, application programs, instructions and/or data to the computing device 1500 include radio or infra-red transmission channels as well as a network connection to another computer or networked device, and the Internet or Intranets including e-mail transmissions and information recorded on Websites and the like.

[116] The computer programs (also called computer program code) are stored in main memory 1508 (which may comprise one or more memory modules) and/or secondary memory 1510. Computer programs can also be received via the communication interface 1524. Such computer programs, when executed, enable the computing device 1500 to perform one or more features of embodiments discussed herein. In various embodiments, the computer programs, when executed, enable the processor 1504 to perform features of the above-described embodiments. Accordingly, such computer programs represent controllers of the computer system 1500.

[117] Software may be stored in a computer program product and loaded into the computing device 1500 using the removable storage drive

1514, the storage drive 1512, or the interface 1540. Alternatively, the computer program product may be downloaded to the computer system 1500 over the communications path 1526. The software, when executed by the processor 1504, causes the computing device 1500 to perform functions of embodiments described herein.

[118] It is to be understood that the embodiment of FIG. 15 is presented merely by way of example. Therefore, in some embodiments one or more features of the computing device 1500 may be omitted. Also, in some embodiments, one or more features of the computing device 1500 may be combined together. Additionally, in some embodiments, one or more features of the computing device 1500 may be split into one or more component parts.

[119] It will be appreciated that the elements illustrated in FIG. 15 function to provide means for performing the computer implemented method as described with respect to FIG. 1. For example, the computing device 1500 provides an apparatus for performing a method for automatically determining sleep stages, the apparatus comprising: at least one processor 1504, at least one memory 1508 including computer program code and at least one communication interface 1524.

[120] The main memory 1508 and the computer program code are configured to, with at least one processor 1504, cause the apparatus at least to: receive input data, through the communication interface 1524, comprising a continuous sequence of sleep epochs measured from a subject, using at least one processor 1504.

[121] The at least one memory 1508 and the computer program code are further configured to cause the at least one processor 1504 to apply a model to a plurality of the sleep epochs, the model comprising a plurality of weightings for emphasising at least one feature of each epoch.

5 [122] The at least one memory 1508 and the computer program code are further configured to determine, from the weighted features of the respective sleep epoch, a sleep stage probability for each of a plurality of sleep stages, each sleep stage probability defining a likelihood the subject was in the respective sleep stage at the time the respective sleep epoch
10 was measured.

[123] The at least one memory 1508 and the computer program code are further configured to cause the at least one processor 1504 to adapt the sleep stage probabilities of each sleep epoch based on one or more sleep stage probabilities of at least one neighbouring sleep epoch in the
15 plurality of sleep epochs.

[124] The at least one memory 1508 and the computer program code are further configured to cause the at least one processor 1504 to determine a most probable class (FMPC) of the respective sleep epoch, the FMPC being the sleep stage with the highest adapted probability.

20 [125] The computing device 1500 of FIG. 15 may execute the process shown in FIG. 1 when the computing device 1500 executes instructions which may be stored in any one or more of the removable storage medium 1544, the removable storage unit 1522 and storage drive 1512. These components 1522, 1544 and 1512 provide a non-
25 transitory computer readable medium having stored thereon

executable instructions for controlling a computer to perform steps comprising: (A) receiving input data representing a continuous sequence of sleep epochs measured from a subject; (B) extracting one or more features for each of a plurality of sleep epochs and applying a plurality of weightings for emphasising at least one feature of each sleep epoch of the plurality of sleep epochs; (C) determining, from the weighted features of the respective sleep epoch, a sleep stage probability for each of a plurality of sleep stages, each sleep stage probability defining a likelihood the subject was in the respective sleep stage at the time the respective sleep epoch was measured; (D) adapting the sleep stage probabilities of each sleep epoch based on one or more sleep stage probabilities of at least one neighbouring sleep epoch in the plurality of sleep epochs; and (E) determining a final most probable class (FMPC) of the respective sleep epoch, the FMPC being the sleep stage with the highest adapted probability.

[126] It will be appreciated by a person skilled in the art that numerous variations and/or modifications may be made to the present invention as shown in the specific embodiments without departing from the spirit or scope of the invention as broadly described. The present embodiments are, therefore, to be considered in all respects to be illustrative and not restrictive.

CLAIMS

1. A method for automatically determining sleep stages, comprising:
 - receiving input data representing a continuous sequence of sleep epochs measured from a subject;
 - 5 extracting one or more features for each of a plurality of sleep epochs and applying a plurality of weightings for emphasising at least one feature of each sleep epoch of the plurality of sleep epochs;
 - determining, from the weighted features of the respective sleep epoch, a sleep stage probability for each of a plurality of sleep stages, each sleep stage
 - 10 probability defining a likelihood the subject was in the respective sleep stage at the time the respective sleep epoch was measured;
 - adapting one or more of the sleep stage probabilities of each sleep epoch based on one or more sleep stage probabilities of at least one neighbouring sleep epoch in the plurality of sleep epochs; and
 - 15 determining the final most probable class (FMPC) of the respective sleep epoch, the FMPC being the sleep stage with the highest adapted probability.
2. A method according to claim 1, wherein the adapting step comprises adapting the sleep stage probability of each epoch based on one or more sleep stage
- 20 probabilities of at least one earlier neighbouring epoch and at least one later neighbouring epoch.
3. A method according to claim 2, wherein the adapting step comprises adapting the sleep stage probability of each epoch based on one or more sleep stage

probabilities of five earlier neighbouring epochs and five later neighbouring epochs in the plurality of sleep epochs.

4. A method according to any preceding claim, wherein determining a sleep stage probability for each of a plurality of sleep stages for each respective epoch further comprises determining an intermediate most probable class (IMPC) for the respective sleep epoch, the IMPC being the sleep stage with highest sleep stage probability.
- 10 5. A method according to claim 4, wherein the adapting step comprises adapting the sleep stage probabilities of the respective sleep epoch based on the IMPC of at least one respective neighbouring sleep epoch in the plurality of sleep epochs.
- 15 6. A method according to claim 5, wherein the adapting step comprises applying a weighting to the IMPC of the at least one neighbouring sleep epoch, the weighting being based on the IMPC of the respective sleep epoch.
- 20 7. A method according to any preceding claim, wherein the plurality of weightings comprise a plurality of weighting stages each comprising a respective set of weightings, and applying the plurality of weightings comprises applying the weightings of each weighting stage to the features of the epoch to emphasise one or more features of the respective epoch relative to one or more other features of the respective epoch.

8. A method according to claim 7, wherein the weighting stages are applied in sequence.
9. A method according to claim 7 or 8, wherein applying a model comprises
- 5 applying a neural network to the features of the epoch.
10. A method according to any preceding claim, further comprising:
- determining a confidence level of the FMPC by dividing the highest adapted probability by a second highest adapted probability; and
- 10 accepting the FMPC as correct if the confidence is at least as high as a threshold confidence level.
11. A method according to claim 10, further comprising:
- comparing the epoch to historical data to determine the sleep stage most
- 15 closely approximated by the epoch, if the sleep stage probability is lower than the threshold confidence level.
12. A method according to claim 4, wherein adapting one or more of the sleep stage probabilities of each epoch comprises adapting the sleep stage probability
- 20 of the IMPC of each sleep epoch based on one or more sleep stage probabilities of at least one neighbouring sleep epoch in the plurality of sleep epochs.
13. A method according to claim 12, wherein determining a sleep stage probability for each of a plurality of sleep stages further comprises determining
- 25 the sleep stage with second highest sleep stage probability, and wherein

adapting one or more of the sleep stage probabilities of each epoch comprises
adapting the sleep stage probability of the IMPC of each sleep epoch, and the
respective second highest sleep stage probability, based on one or more sleep
stage probabilities of at least one neighbouring sleep epoch in the plurality of
5 sleep epochs.

14. A method according to claim 1, wherein adapting one or more of the sleep
stage probabilities of each sleep epoch comprises adapting all of the sleep stage
probabilities of each epoch based on one or more sleep stage probabilities of at
10 least one neighbouring sleep epoch in the plurality of sleep epochs.

15. A method according to any preceding claim, further comprising:
temporally splitting the extracted one or more features for one of the
plurality of sleep epochs into a plurality of subsections;
15 stacking the plurality of subsections into a plurality of stacks;
applying weightings to the plurality of stacks and the extracted one or
more features for the one of the plurality of sleep epochs; and
determining an artefact probability for each subsection based on the
weighted plurality of stacks and the weighted features, each artefact probability
20 defining a likelihood that artefacts are present in the respective subsection.

16. A system for automatically determining sleep stages, comprising at least
one processor and at least one memory unit communicatively coupled to
each respective processor and comprising instructions that, when executed
25 by the processor cause the system to:

receive input data from one or more sensors, the data representing a continuous sequence of sleep epochs measured from a subject;

extract one or more features for each of a plurality of the sleep epochs and apply a plurality of weightings for emphasising at least one feature of each

5 sleep epoch of the plurality of sleep epochs;

determine, from the weighted features of the respective sleep epoch, a sleep stage probability for each of a plurality of sleep stages, each sleep stage probability defining a likelihood the subject was in the respective sleep stage at the time the respective sleep epoch was measured;

10 adapt the sleep stage probabilities of each sleep epoch based on one or more sleep stage probabilities of at least one neighbouring sleep epoch in the plurality of sleep epochs;

determine the final most probable class (FMPC) of the respective sleep epoch, the FMPC being the sleep stage with the highest adapted probability; and

15 output to a display the FMCP of the respective epoch.

17. A system according to claim 16, wherein the instructions, when executed by the at least one processor, cause the system to adapt the sleep stage probabilities of each epoch based on one or more sleep stage probabilities of at least one
20 earlier neighbouring epoch and at least one later neighbouring epoch.

18. A system according to claim 17, wherein the instructions, when executed by the at least one processor, cause the system to adapt the sleep stage probability of each epoch based on one or more sleep stage probabilities of five earlier

neighbouring epochs and five later neighbouring epochs in the plurality of sleep epochs.

19. A system according to any one of claims 16 to 18, wherein the instructions,
5 when executed by the at least one processor, cause the system to determine, from the weighted features of the respective sleep epoch, an intermediate most probable class (IMPC) for the respective sleep epoch, the IMPC being the sleep stage with highest sleep stage probability.
- 10 20. A system according to claim 19, wherein the instructions, when executed by the at least one processor, cause the system to adapt the sleep stage probabilities of the respective sleep epoch based on the IMPC of at least one respective neighbouring sleep epoch in the plurality of sleep epochs.
- 15 21. A system according to claim 20, wherein the instructions, when executed by the at least one processor, cause the system to adapt the sleep stage probabilities by applying a weighting to the IMPC of the at least one neighbouring sleep epoch, the weighting being based on the IMPC of the respective sleep epoch.
- 20 22. A system according to any one of claims 16 to 21, wherein the model comprises a plurality of weighting stages each comprising a plurality of weightings, and wherein the instructions, when executed by the at least one processor, cause the system to apply a weighting model by applying the weightings of each weighting stage to the features of the epoch to emphasise one

or more features of the respective epoch relative to one or more other features of the respective epoch.

23. A system according to claim 22, wherein the instructions, when executed by
5 the at least one processor, cause the system to apply the weighting in sequence.

24. A system according to claim 22 or 23, wherein the instructions, when
executed by the at least one processor, cause the system to apply a model by
applying a neural network to the features of the epoch to ascertain the sleep
10 stage the subject is most likely to be in at the time of measurement of the epoch.

25. A system according to any one of claims 16 to 24, wherein the instructions,
when executed by the at least one processor, cause the system to:

temporally split the extracted one or more features for one of the
15 plurality of sleep epochs into a plurality of subsections;

stack the plurality of subsections into a plurality of stacks;

apply weightings to the plurality of stacks and the extracted one or more
features for the one of the plurality of sleep epochs; and

determine an artefact probability for each subsection based on the
20 weighted plurality of stacks and the weighted features, each artefact probability
defining a likelihood that artefacts are present in the respective subsection.

1/18

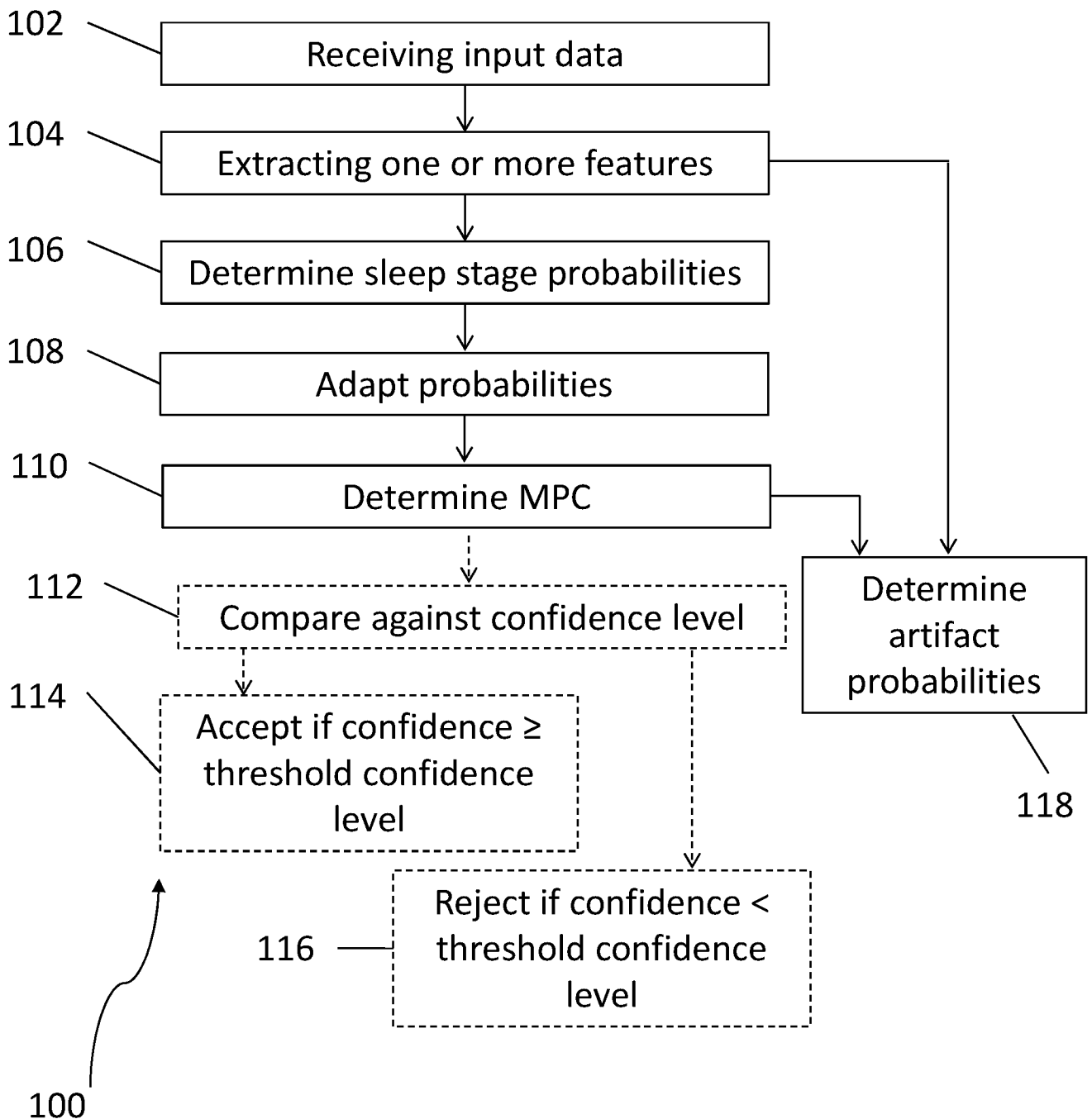


FIG. 1

2/18

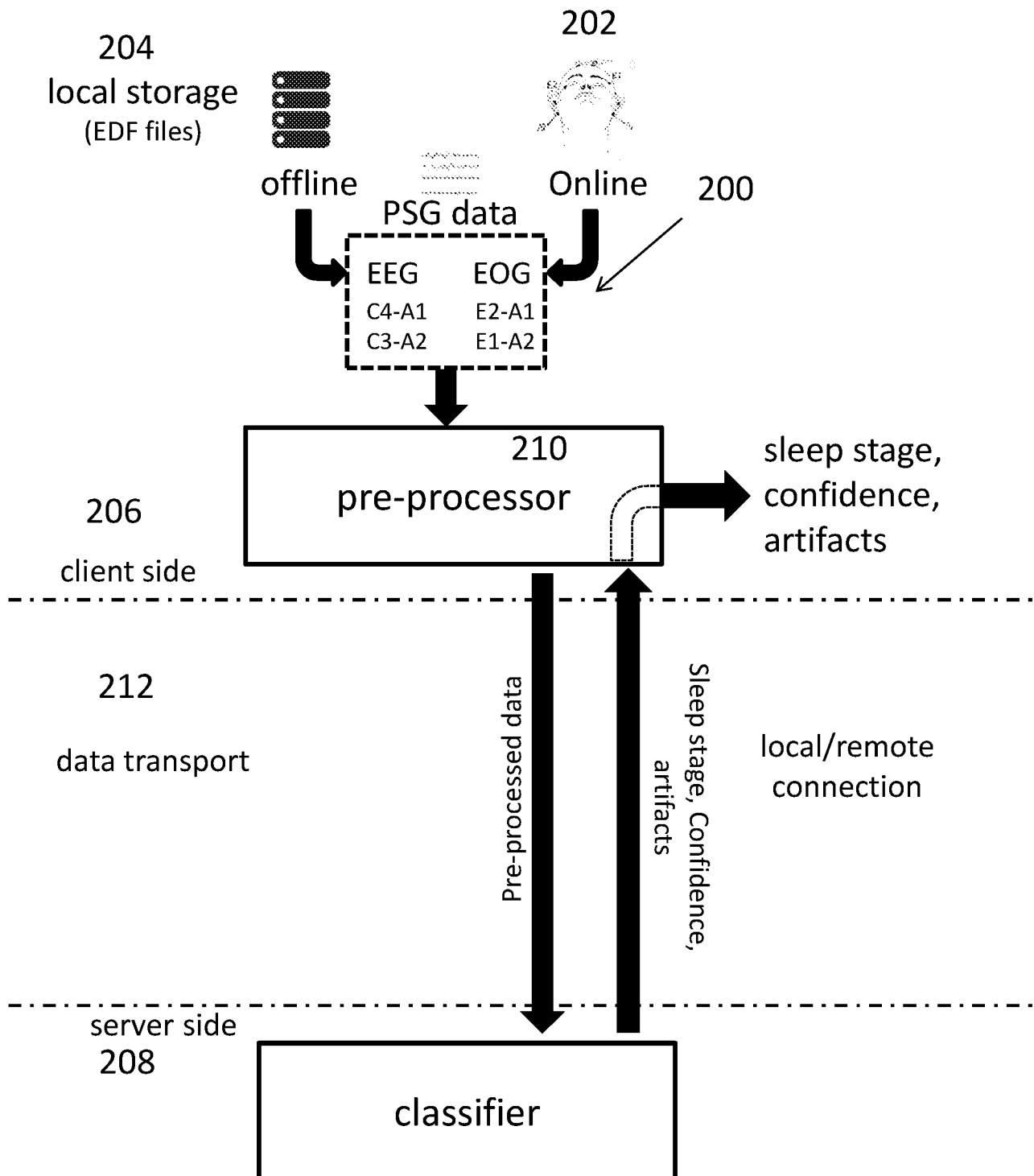


FIG. 2

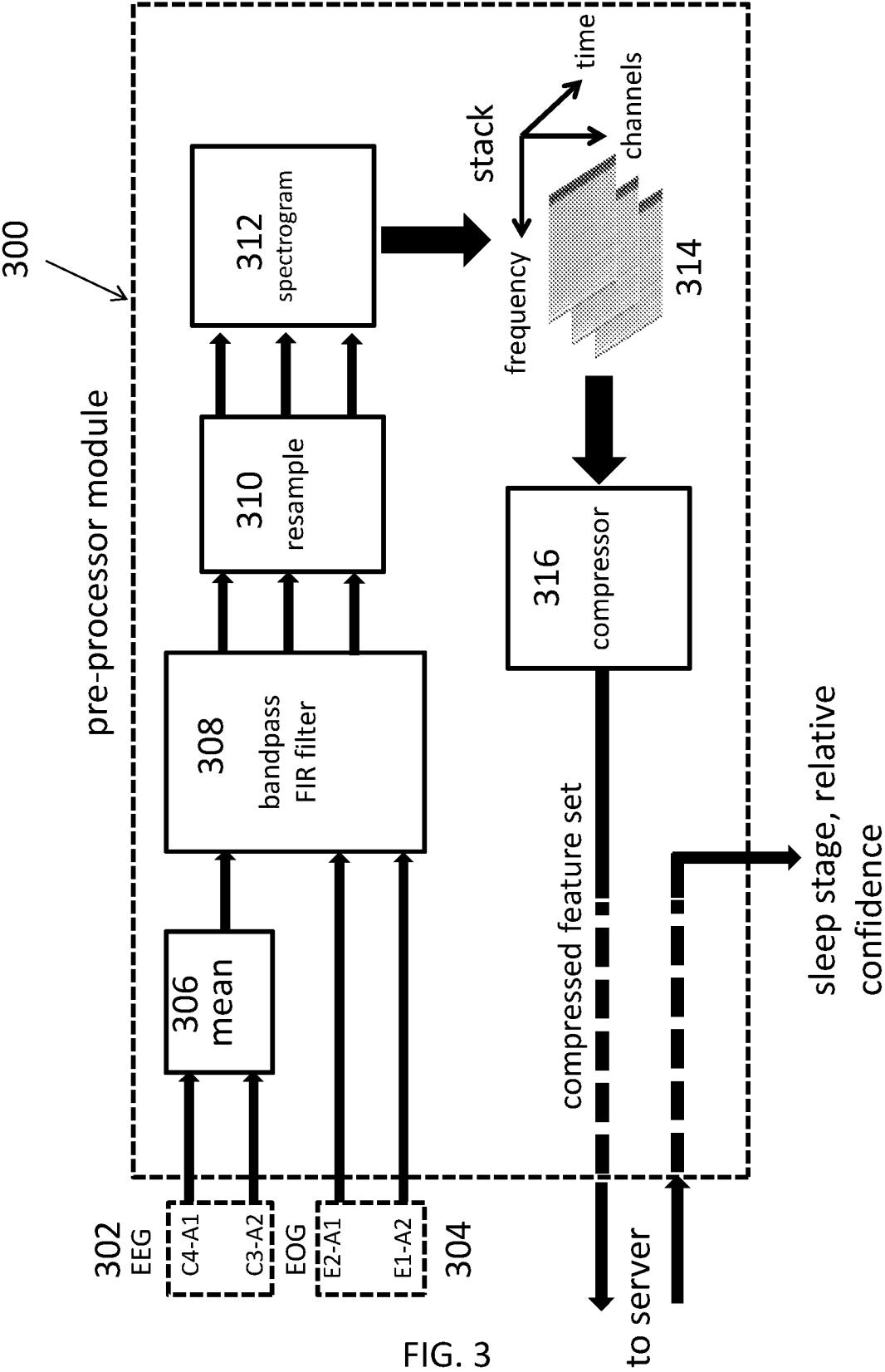


FIG. 3

4/18

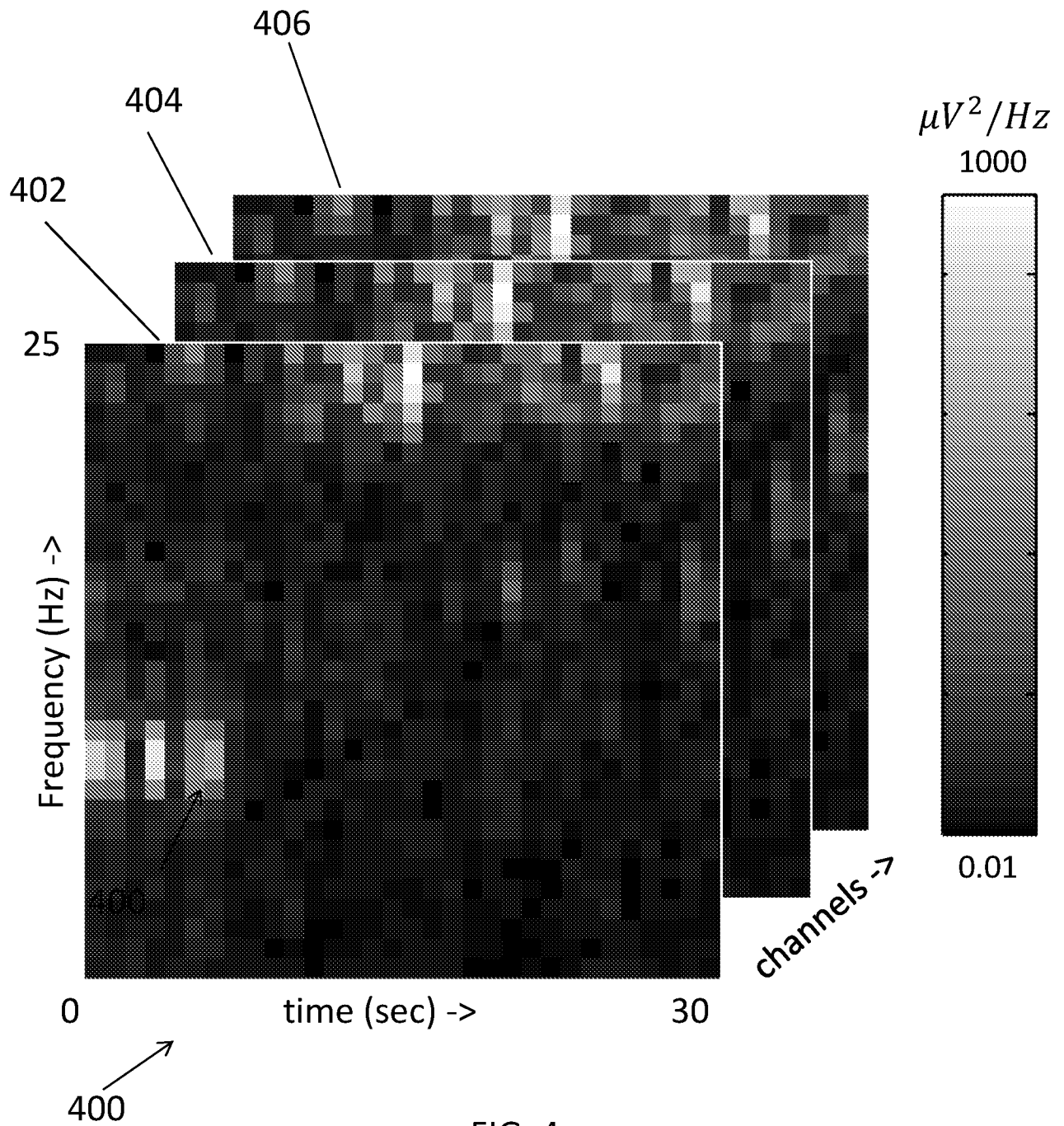


FIG. 4

5/18

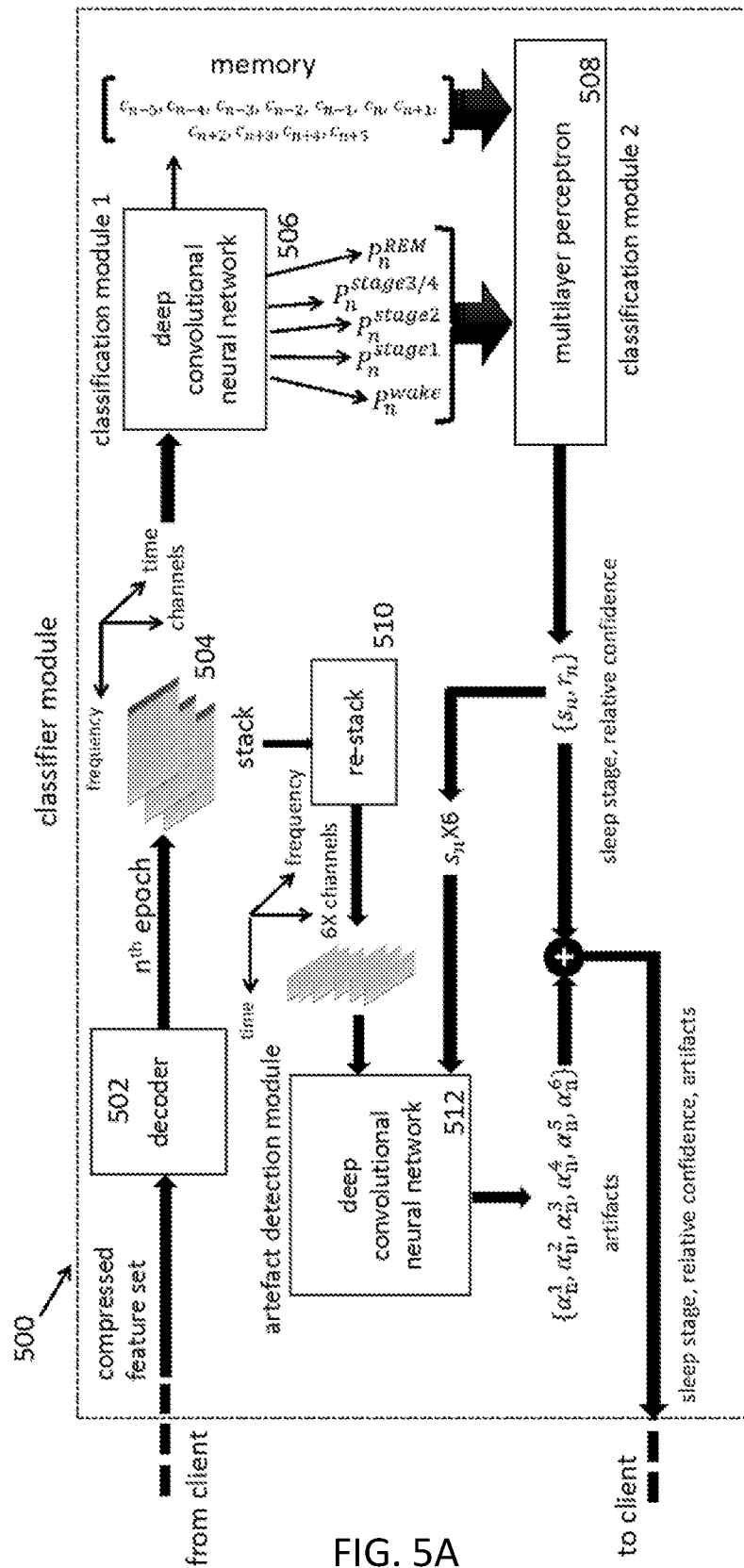


FIG. 5A

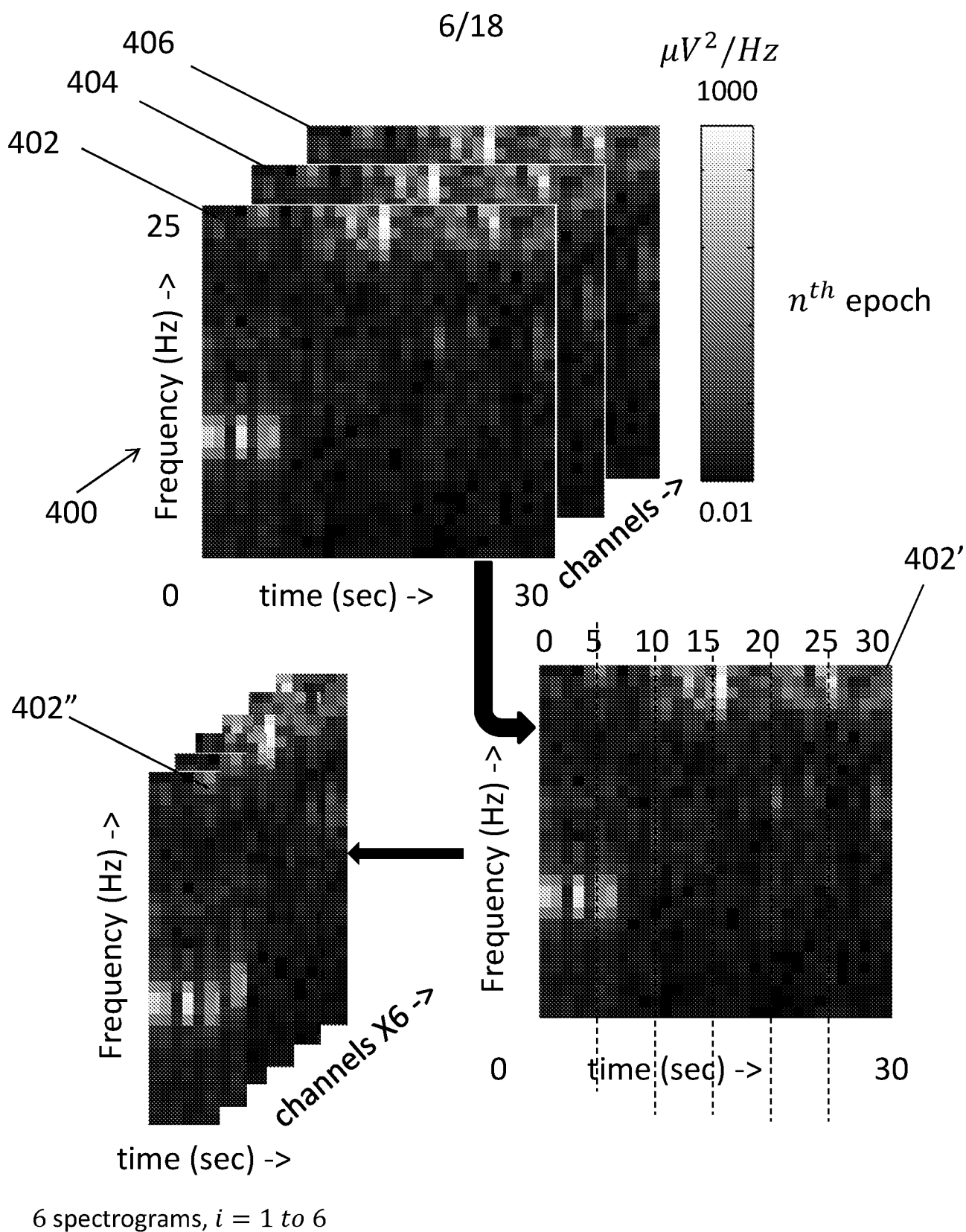


FIG. 5B

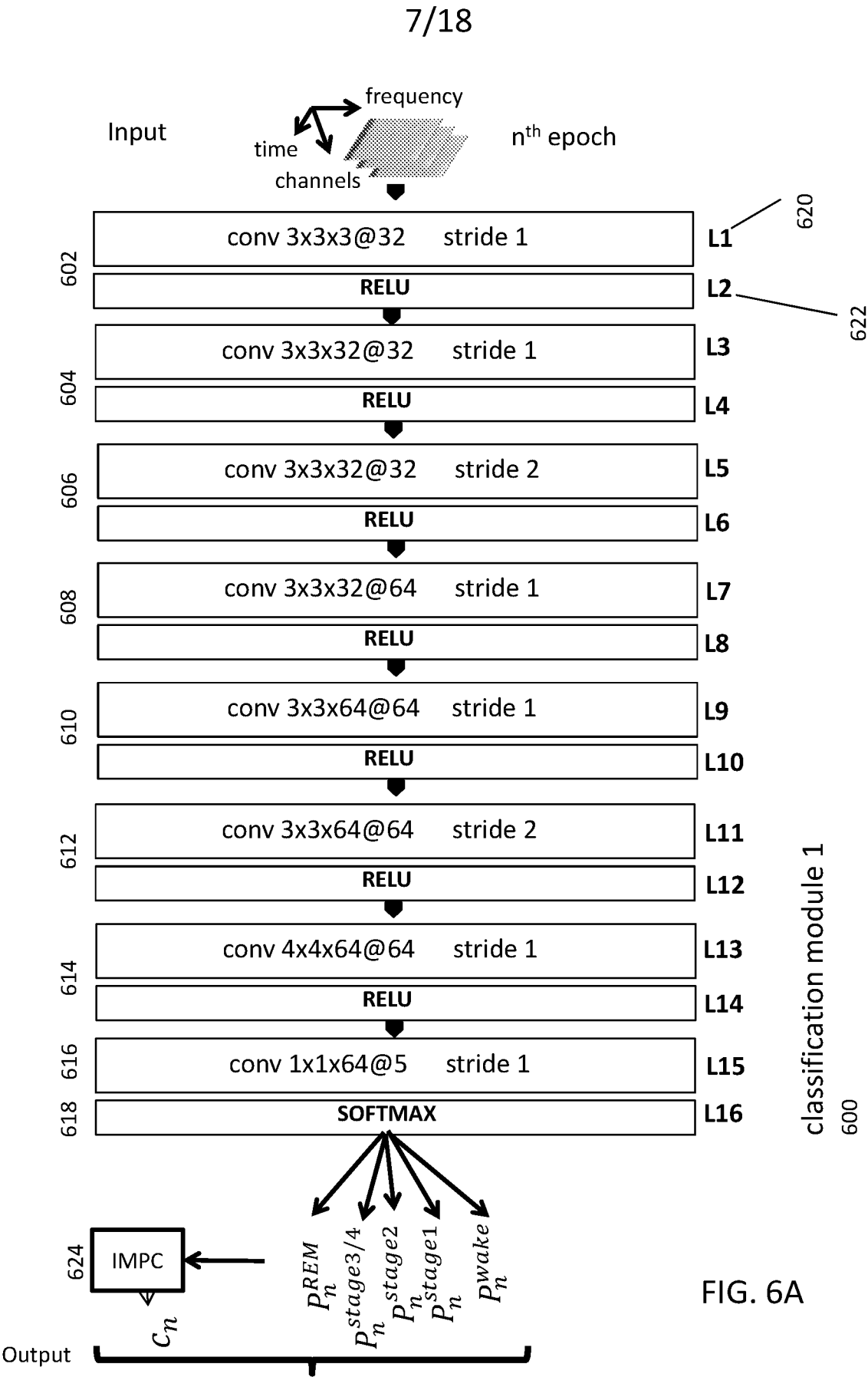
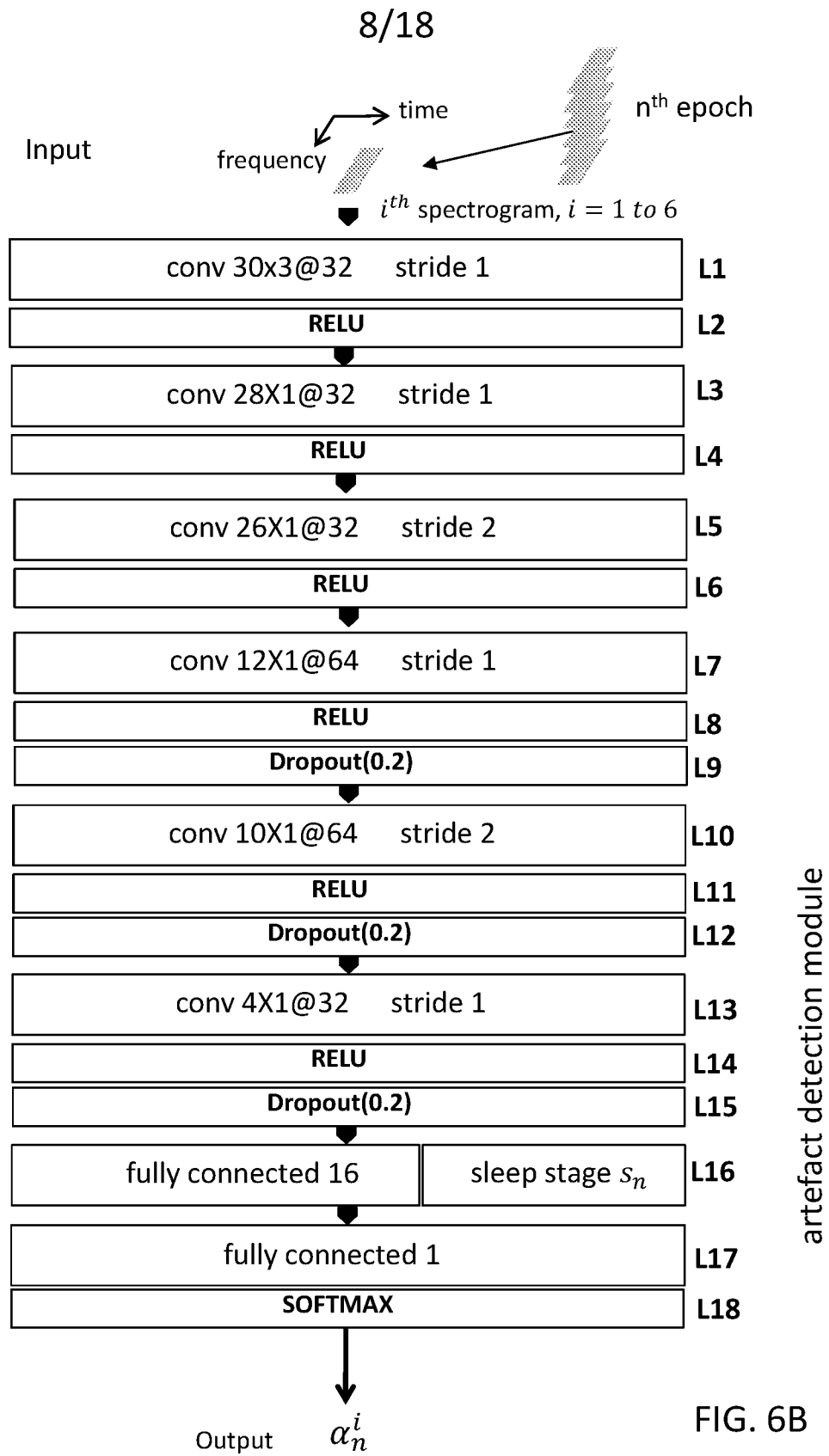


FIG. 6A



9/18

layer number	Type	filter support	filter dimension	no. of filters	stride	data size	weights
input	n/a	n/a	n/a	n/a	n/a	32x32x3	n/a
L1	conv	3X3	3	32	1	30X30X32	896
L2	RELU	1X1	n/a	n/a	1	30X30X32	n/a
L3	conv	3X3	32	32	1	28X28X32	9248
L4	RELU	1X1	n/a	n/a	1	28X28X32	n/a
L5	conv	3X3	32	32	2	13X13X32	9248
L6	RELU	1X1	n/a	n/a	1	13X13X32	n/a
L7	conv	3X3	32	64	1	11X11X64	18496
L8	RELU	1X1	n/a	n/a	1	11X11X64	n/a
L9	conv	3X3	64	64	1	9X9X64	36928
L10	RELU	1X1	n/a	n/a	1	9X9X64	n/a
L11	conv	3X3	64	64	2	4X4X64	36928
L12	RELU	1X1	n/a	n/a	1	4X4X64	n/a
L13	conv	4X4	64	64	1	64	65600
L14	RELU	1	n/a	n/a	1	64	n/a
L15	conv	1	64	5	1	5	325
L16	softmax	1	n/a	n/a	1	5	n/a
output	n/a	n/a	n/a	n/a	n/a	6	n/a

700



FIG. 7A

10/18

layer number	Type	filter support	filter dimension	no. of filters	stride	data size	weights
input	n/a	n/a	n/a	n/a	n/a	32x5x1	n/a
L1	conv	3X3	1	32	1	30X3X32	320
L2	RELU	1X1	n/a	n/a	1	30X3X32	n/a
L3	conv	3X1	32	32	1	28X1X32	9248
L4	RELU	1X1	n/a	n/a	1	28X1X32	n/a
L5	conv	3X1	32	32	2x1	26X1X32	3104
L6	RELU	1X1	n/a	n/a	1	26X1X32	n/a
L7	conv	3X1	32	64	1	12X1X64	6208
L8	RELU	1X1	n/a	n/a	1	12X1X64	n/a
L9	dropout(0.2)	1X1	n/a	n/a	1	12X1X64	n/a
L10	conv	3X1	64	64	2x1	10X1X64	12352
L11	RELU	1X1	n/a	n/a	1	10X1X64	n/a
L12	dropout(0.2)	1X1	n/a	n/a	1	10X1X64	n/a
L13	conv	3X1	32	32	2x1	4X1X32	6176
L14	RELU	1X1	n/a	n/a	1	4X1X32	n/a
L15	dropout(0.2)	1X1	n/a	n/a	1	4X1X32	n/a
L16	fully connected	16 + 1 sleep stage	n/a	n/a	n/a	17	1935
L17	fully connected	1	n/a	n/a	n/a	1	17
L18	softmax	1	n/a	n/a	1	1	n/a
output	n/a	n/a	n/a	n/a	n/a	1	n/a

FIG. 7B

11/18

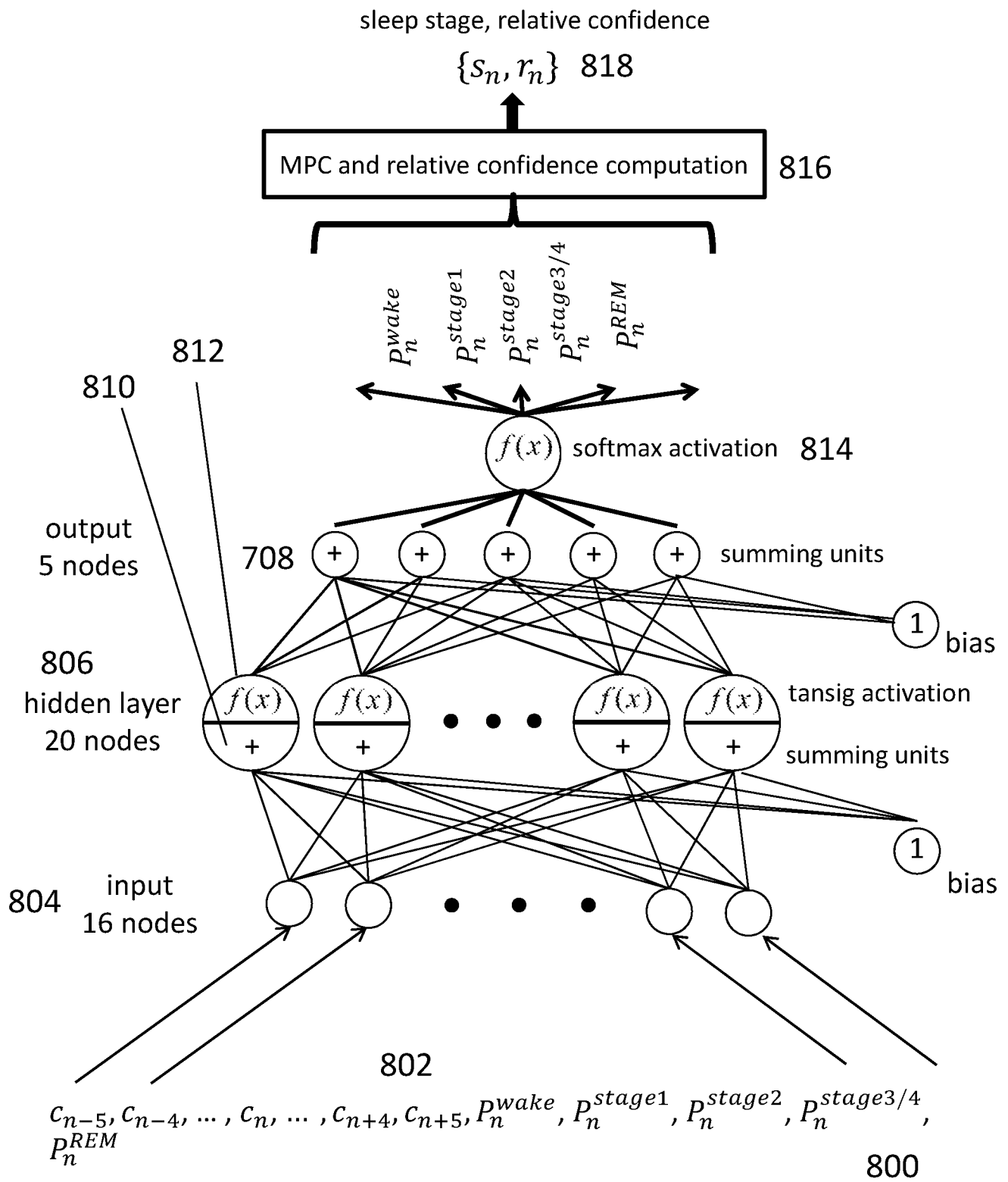


FIG. 8

classification module 2

12/18

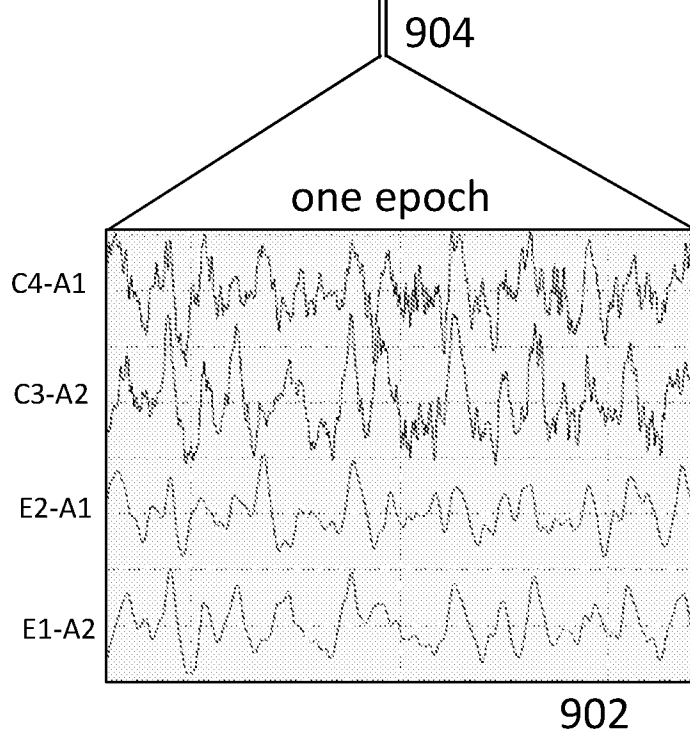
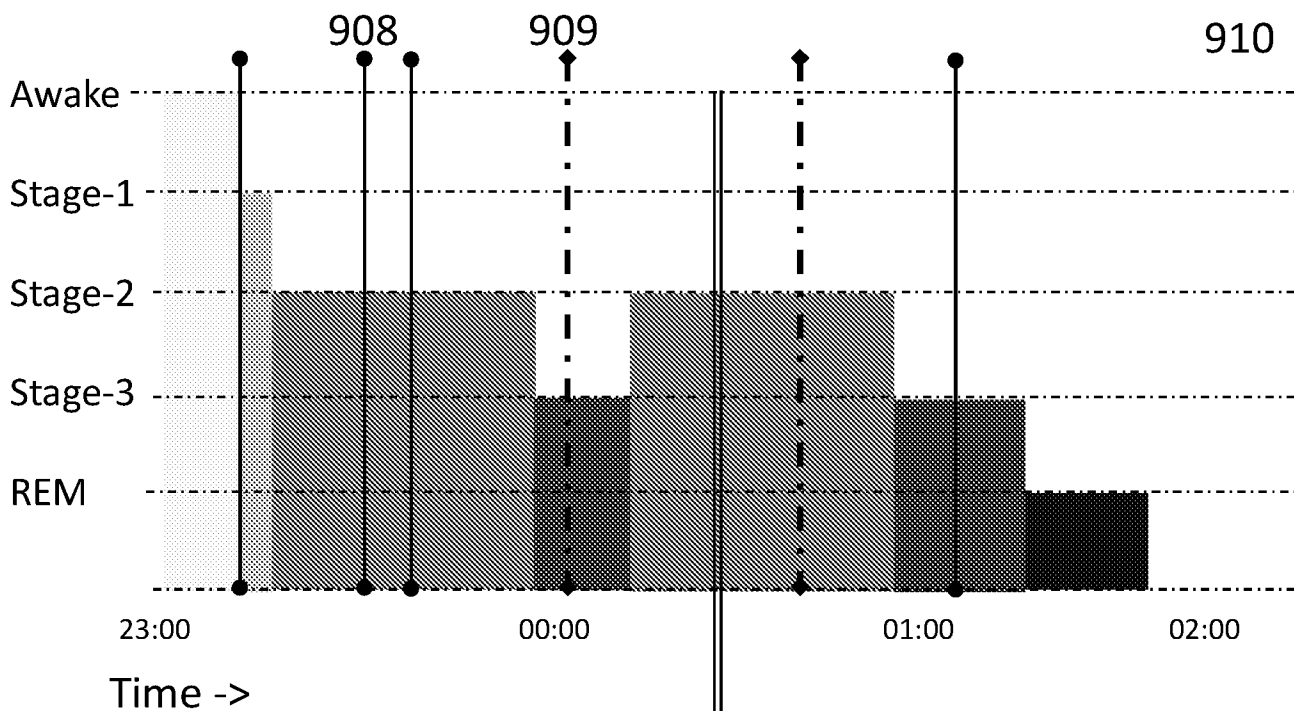


FIG. 9

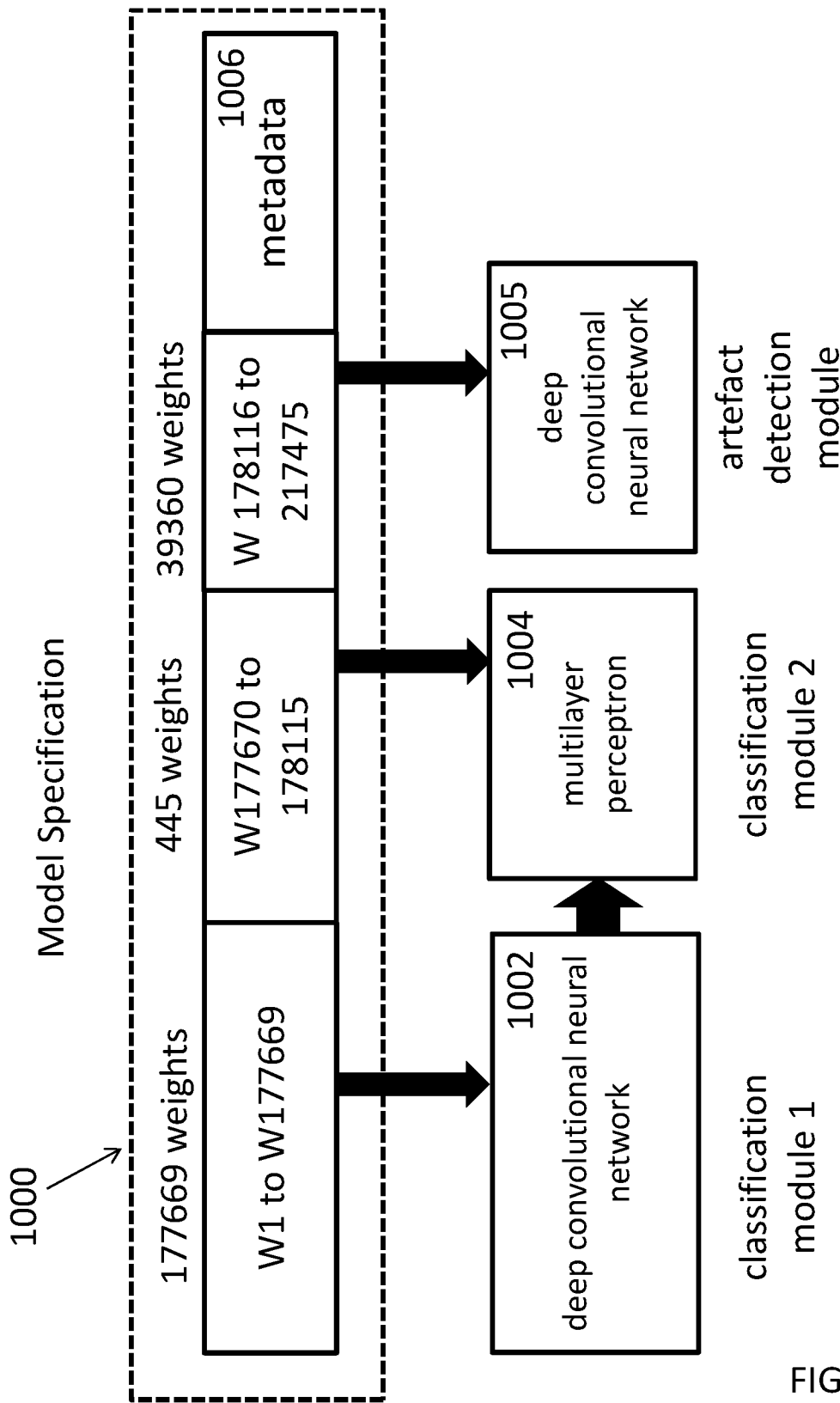


FIG. 10

14/18

1100

Model Name: NFS1-CSL1-CSL2**Date:** 22-Feb-2016**Dataset Size:** 648,451 epochs from three independent datasets**Training Format:** Randomly shuffled; 70% training, 15% testing and 15% validation**Demographics:** Healthy young subjects**Recommended threshold:** 1.5

1102

1104

1106

Training Confusion Matrix 1112

Output Class	W	S1	S2	S3	R	
	8105 1.8%	2262 0.5%	115 0.0%	1738 0.4%	1113 0.2%	50.8% 39.2%
	5133 1.1%	156311 34.7%	6285 1.4%	1484 0.3%	4502 1.0%	90.0% 10.0%
	222 0.0%	5170 1.1%	87634 19.4%	369 0.1%	14 0.0%	93.8% 6.2%
	2491 0.6%	1543 0.3%	312 0.1%	77098 17.1%	803 0.2%	93.7% 6.3%
	3663 0.8%	5899 1.3%	25 0.0%	1295 0.3%	77229 17.1%	87.6% 12.4%
						41.3% 58.7%
						91.3% 8.7%
						92.9% 7.1%
						94.0% 6.0%
						92.3% 7.7%
						90.1% 9.9%
						W S1 S2 S3 R
						Target Class

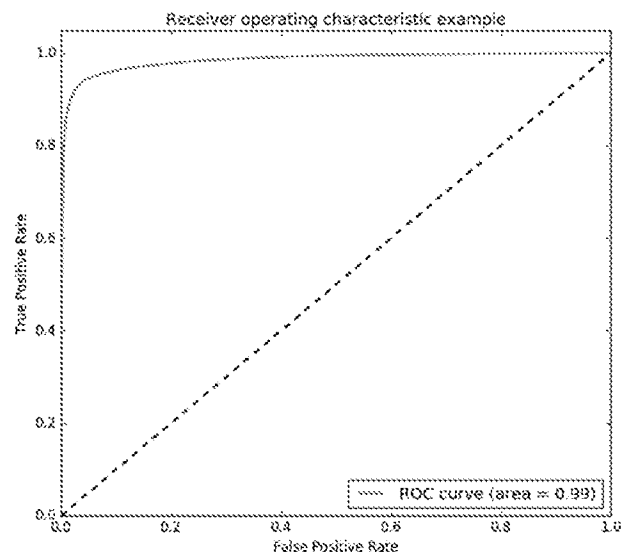
Validation Confusion Matrix 1110

Output Class	W	S1	S2	S3	R	
	1726 1.8%	540 0.6%	24 0.0%	406 0.4%	239 0.2%	58.8% 41.2%
	1134 1.2%	33728 34.9%	1362 1.4%	323 0.3%	969 1.0%	88.9% 10.1%
	45 0.0%	1078 1.1%	18540 19.2%	69 0.1%	3 0.0%	93.9% 6.1%
	583 0.6%	337 0.3%	83 0.1%	16575 17.2%	182 0.2%	93.3% 6.7%
	751 0.8%	1276 1.3%	2 0.0%	265 0.3%	16363 16.9%	87.7% 12.3%
						40.7% 59.3%
						91.3% 8.7%
						92.6% 7.4%
						94.0% 6.0%
						92.2% 7.8%
						90.0% 10.0%
						W S1 S2 S3 R
						Target Class

Test Confusion Matrix 1108

Output Class	W	S1	S2	S3	R	
	1765 1.8%	436 0.5%	18 0.0%	400 0.4%	223 0.2%	62.1% 37.9%
	1111 1.2%	33520 34.7%	1384 1.4%	288 0.3%	989 1.0%	88.9% 10.1%
	50 0.1%	1123 1.2%	18887 19.6%	80 0.1%	6 0.0%	93.8% 6.2%
	530 0.5%	313 0.3%	53 0.1%	16529 17.1%	187 0.2%	93.9% 6.1%
	778 0.8%	1236 1.3%	7 0.0%	281 0.3%	16409 17.0%	87.7% 12.3%
						41.7% 58.3%
						91.5% 8.5%
						92.8% 7.2%
						94.0% 6.0%
						92.1% 7.9%
						90.2% 9.8%
						W S1 S2 S3 R
						Target Class

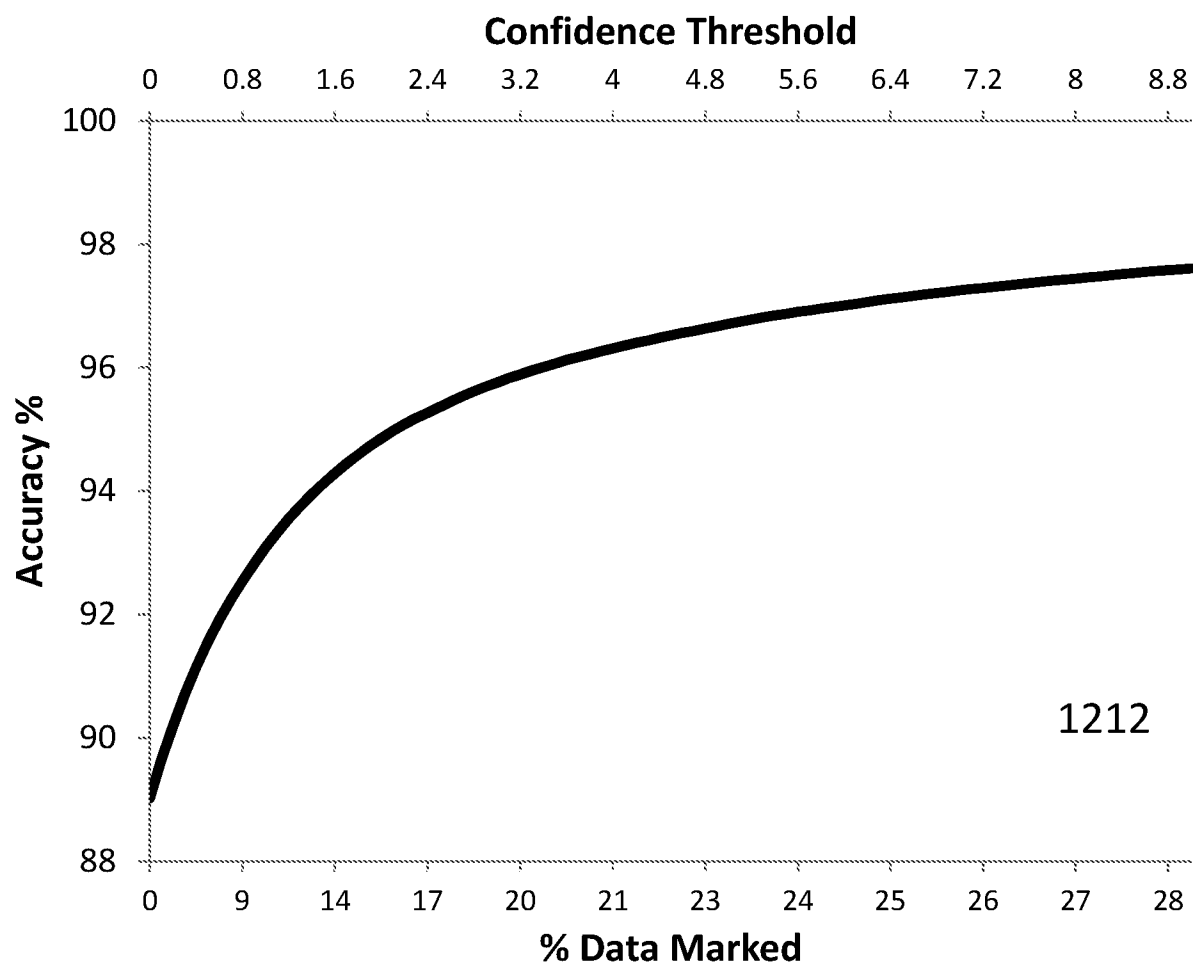
1114



W: wake, S1: stage-1, S2: stage-2, S3: stage-3/4, R: REM

FIG. 11

15/18



1212

FIG. 12

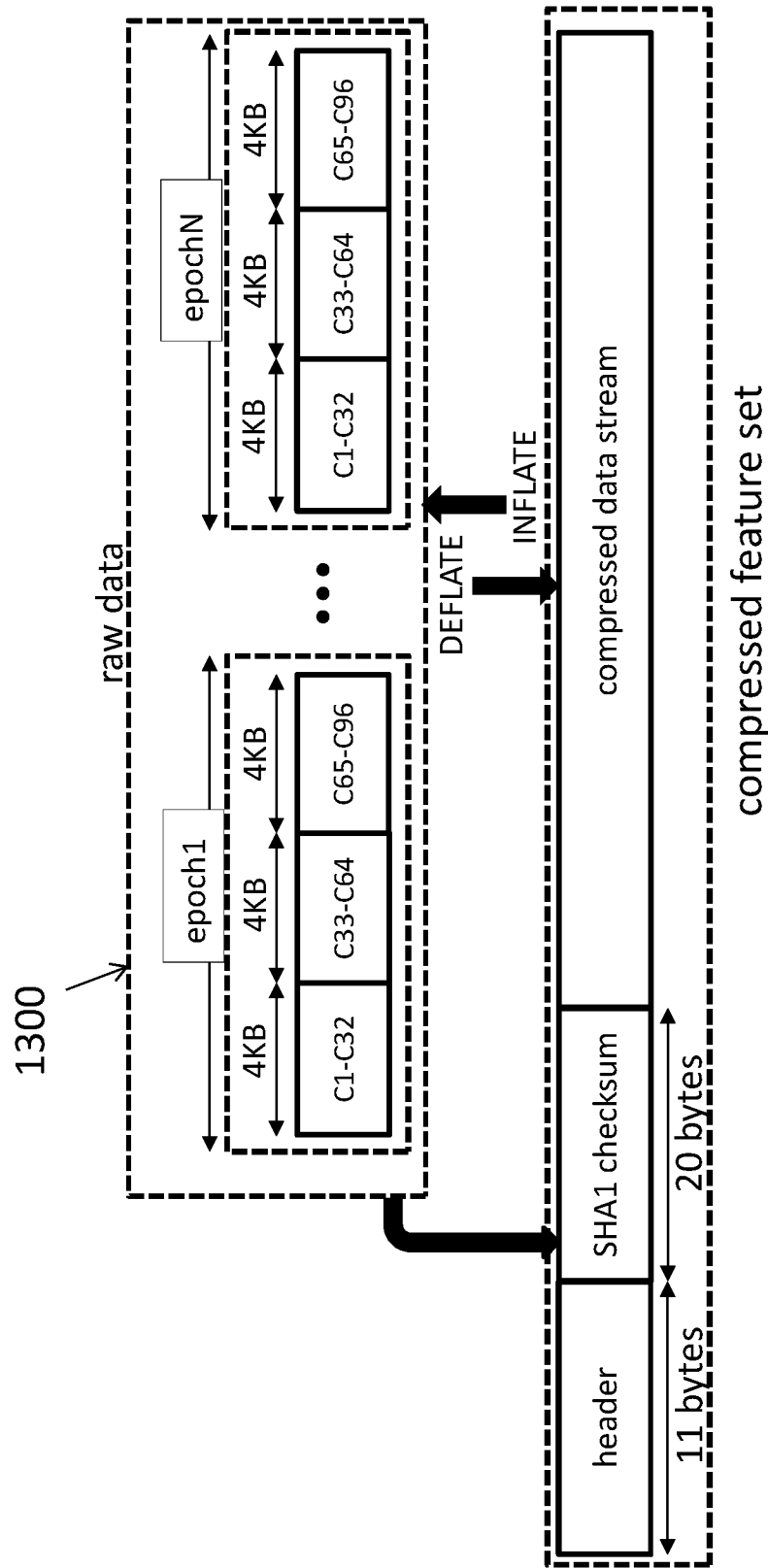


FIG. 13

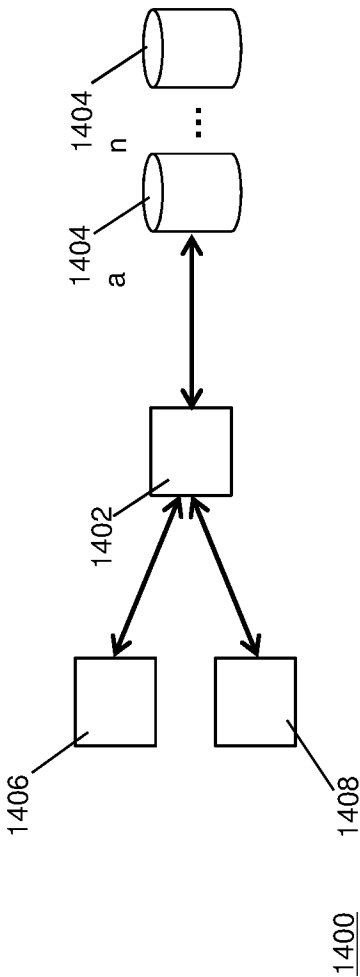
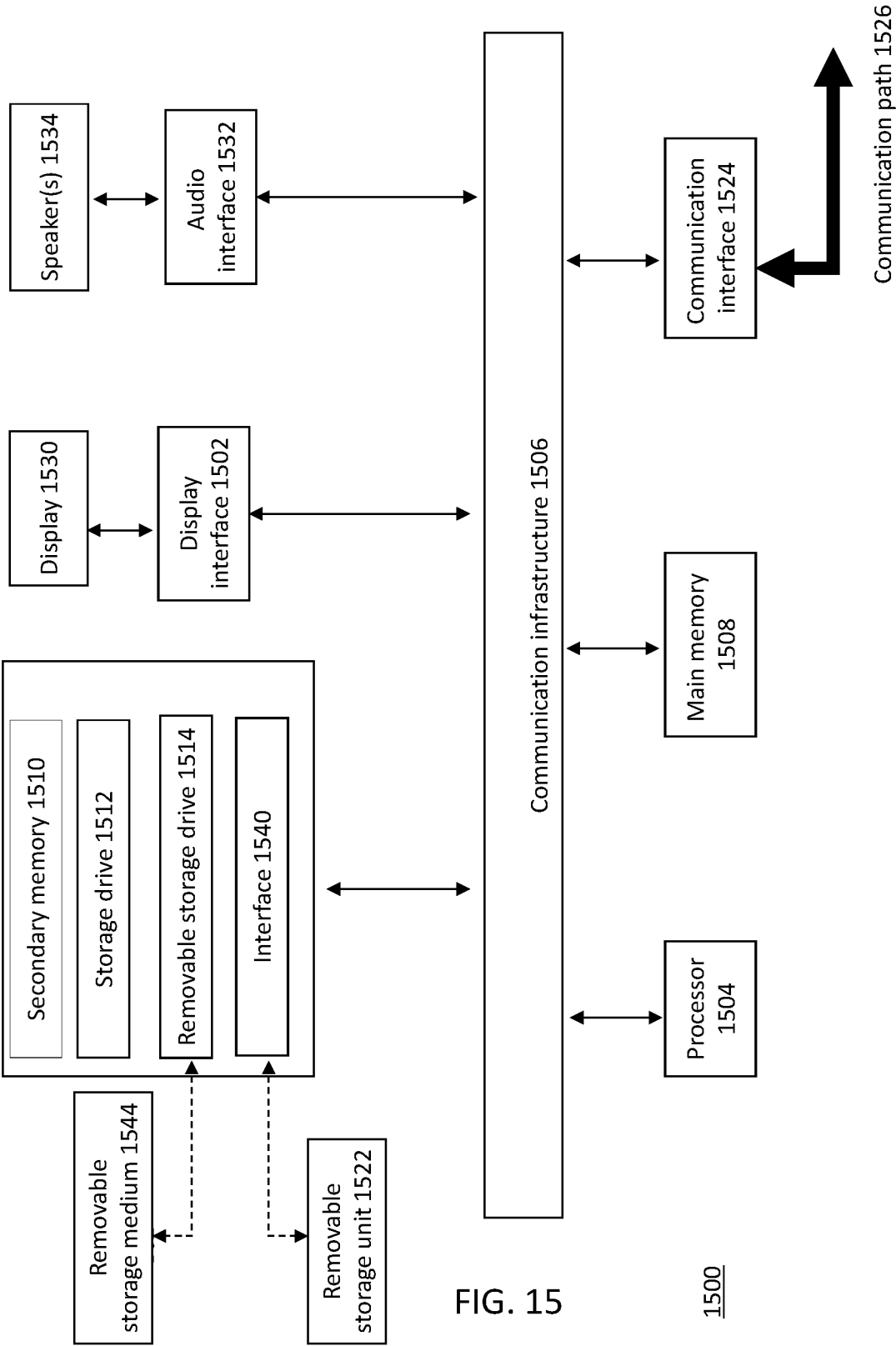


FIG. 14



INTERNATIONAL SEARCH REPORT

International application No.

PCT/SG2017/050508

A. CLASSIFICATION OF SUBJECT MATTER

A61B 5/04 (2006.01) G06F 19/00 (2018.01)

According to International Patent Classification (IPC)

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

A61B, G06F

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

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

FAMPAT/Internet: sleep staging, epoch, smoothing, adapt, modify, weighting, adjust, neighbouring, adjacent, artifact, sub segment, probability, likelihood, confidence, chance, 睡眠, 状态, 调整 and related terms

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5732696 A (RAPOPORT D. M. & NORMAN R. G.) 31 March 1998 Column 2 lines 49-55, Column 4 lines 36-67, column 5 lines 5-30, lines 50-59, column 6 lines 34-67, column 7 line 36 - column 8 line 2, column 9 lines 1-59, column 10 lines 1-15, Figures 7A & 7B	1-25
X	US 2011/0124979 A1 (HENEGHAN C. ET AL.) 26 May 2011 Paragraphs [0027]-[0028], [0031], [0045]-[0046]	1-25
X	US 2015/0190086 A1 (CHAN A. & NARASIMHAN R.) 9 July 2015 Paragraphs [0014], [0018], [0027], [0034]-[0035], [0039]-[0042], [0048]	1-25
A	US 6805668 B1 (CADWELL J. A.) 19 October 2004 Column 6 lines 5-18, column 8 lines 2-30, column 8 line 48 - column 9 line 51	-

☒ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

*Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

20/11/2017

(day/month/year)

Date of mailing of the international search report

08/12/2017

(day/month/year)

Name and mailing address of the ISA/SG



Intellectual Property Office of Singapore

51 Bras Basah Road
#01-01 Manulife Centre
Singapore 189554

Email: pct@ippos.gov.sg

Authorized officer

Lai Tan Lei Tanny (Dr)

IPOS Customer Service Tel. No.: (+65) 6339 8616

INTERNATIONAL SEARCH REPORT

International application No.

PCT/SG2017/050508

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	TIAN J. Y. & LIU J. Q., Automated Sleep Staging by a Hybrid System Comprising Neural Network and Fuzzy Rule-based Reasoning. <i>Proceedings of the 2005 IEEE 27th Annual Conference Engineering in Medicine and Biology</i> , 18 January 2006, pages 4115-4118 [Retrieved on 2017-11-20] <DOI: 10.1109/IEMBS.2005.1615368> Section II	-
A	US 2007/0016095 A1 (LOW P. S. & SEJNOWSKI T. J.) 18 January 2007 Paragraph [0185]	-
A	PARK H.J., Automated Sleep Stage Analysis Using Hybrid Rule-Based and Case-Based Reasoning, Doctoral dissertation submitted to Graduate School of Seoul National University. 31 August 2000 [Retrieved on 2017-11-20 from https://pdfs.semanticscholar.org/353c/8b762609156421885ad688dd63ca03ce6829.pdf] Chapter 5	-
A	US 2011/0295142 A1 (CHAKRAVARTHY N. ET AL.) 1 December 2011 Paragraphs [0088]-[0094]	-
A	US 2011/0224569 A1 (ISENHART R. ET AL.) 15 September 2011 Paragraphs [0031]-[0033]	-
A	GUNNARSDOTTIR K.M., Towards automating sleep stage scoring to diagnose sleep disorders, Thesis submitted to John Hopkins University. 31 May 2016 [Retrieved on 2017-11-20 from https://jscholarship.library.jhu.edu/bitstream/handle/1774.2/39495/GUNNARSDOTTIR-THESIS-2016.pdf?sequence=1&isAllowed=y] Section 2.3.2	-
A	HERRERA L. J. ET AL., Combination of heterogeneous EEG feature extraction methods and stacked sequential learning for sleep stage classification. <i>International journal of neural systems</i> , 19 April 2013, Vol. 23, No. 3, pages 1350012:1-20 [Retrieved on 2017-11-20] <DOI: 10.1142/S0129065713500123> Section 2	-
P,X	WO 2017/040331 A1 (AWARABLES, INC.) 9 March 2017 Paragraphs [0086]-[0089], [0100]-[0115], [0121]-[0123], [0147]-[0148], [0167], Figure 1	1-25
P,X	WO 2017/136352 A1 (VERILY LIFE SCIENCES LLC) 10 August 2017 Paragraphs [043]-[053], [084]-[093], Figures 3, 4, 7 & 8	1-25
P,X	CN 106419893 A (GUANGZHOU CVTE ELECTRONIC TECH) 22 February 2017 Paragraphs [0045]-[0048], [0056], [0060]-[0064] of the original non-English language document (a machine translation is enclosed only for your reference)	1-25

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/SG2017/050508

Note: This Annex lists known patent family members relating to the patent documents cited in this International Search Report. This Authority is in no way liable for these particulars which are merely given for the purpose of information.

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5732696 A	31/03/1998	NONE	
US 2011/0124979 A1	26/05/2011	EP 2185063 A1	19/05/2010
		WO 2009/024273 A1	26/02/2009
US 2015/0190086 A1	09/07/2015	WO 2015/103558 A1	09/07/2015
US 6805668 B1	19/10/2004	NONE	
US 2007/0016095 A1	18/01/2007	CA 2607049 A1	16/11/2006
		CN 101272732 A	24/09/2008
		EP 1885237 B1	22/04/2015
		JP 2014176743 A	25/09/2014
		JP 2008544772 A	11/12/2008
		KR 20080025673 A	21/03/2008
		US 2012/0029378 A1	02/02/2012
		WO 2006/121455 A1	16/11/2006
		WO 2006/122201 A1	16/11/2006
US 2011/0295142 A1	01/12/2011	EP 2575608 A1	10/04/2013
		WO 2011/149752 A1	01/12/2011
US 2011/0224569 A1	15/09/2011	CA2792607 A1	15/09/2011
		EP 2544588 A1	16/01/2013
		US 2013/0211224 A1	15/08/2013
		WO 2011/112652 A1	15/09/2011
WO 2017/040331 A1	09/03/2017	US 2017/0055898 A1	02/03/2017
		US 2017/0055899 A1	02/03/2017
		WO 2017/040333 A1	09/03/2017
WO 2017/136352 A1	10/08/2017	US 2017/0215808 A1	03/08/2017
		US2017/0220752 A1	03/08/2017
		WO 2017/136339 A1	10/08/2017
CN 106419893 A	22/02/2017	NONE	