EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

V-awake

Citation for published version (APA):
Garcia Caballero, H., Westenberg, M., Gebre, B., & van Wijk, J. (2019). V-awake: a visual analytics approach for correcting sleep predictions from deep learning models. *Computer Graphics Forum*, *38*(3), 1-12. https://doi.org/10.1111/cgf.13667

DOI:
10.1111/cgf.13667

Document status and date:
Published: 21/03/2019

Document Version:
Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 04. Oct. 2023

Eurographics Conference on Visualization (EuroVis) 2019
M. Gleicher, H. Leitte, and I. Viola
(Guest Editors)

Volume 38 (2019), Number 3

V-Awake: A Visual Analytics Approach for Correcting Sleep Predictions from Deep Learning Models

Humberto S. Garcia Caballero[1], Michel A. Westenberg[1], Binyam Gebre[2] and Jarke J. van Wijk[1]

[1]Eindhoven University of Technology, The Netherlands
[2]Philips Research, The Netherlands

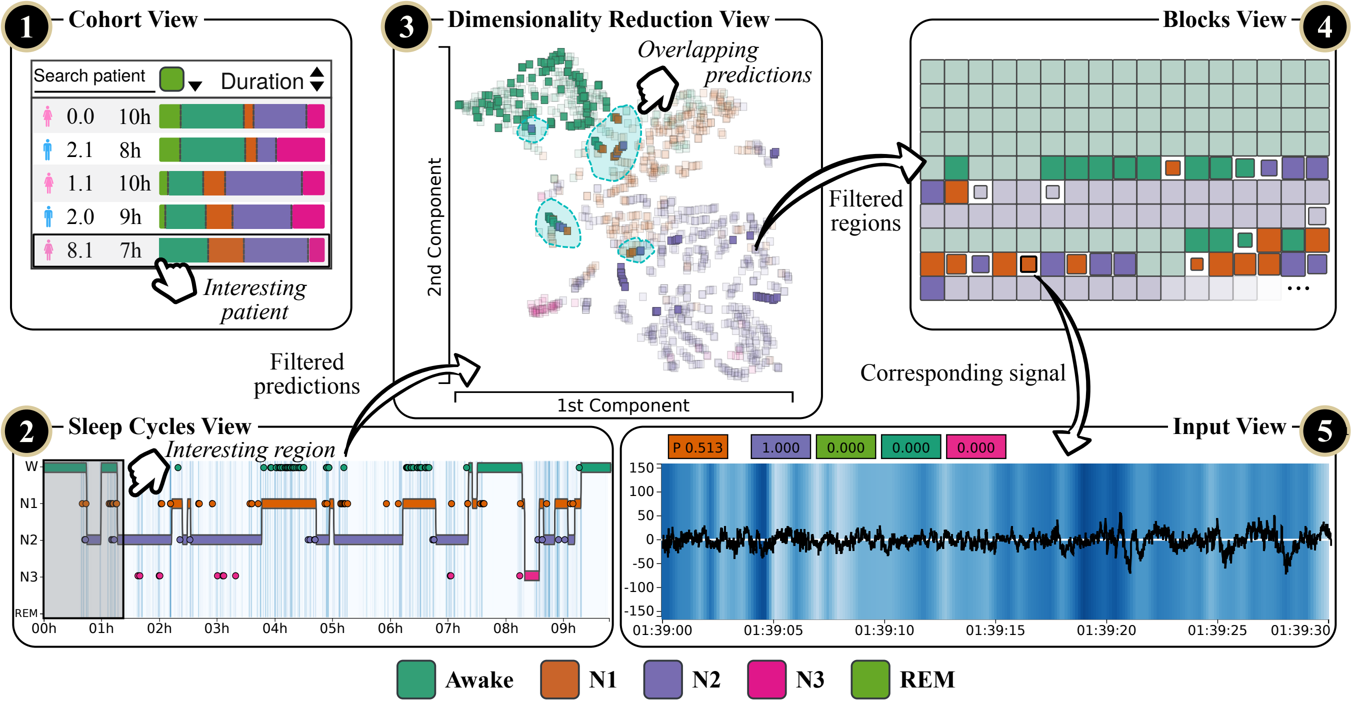


Figure 1: *Depiction of the main components of V-Awake. First, a patient is selected (1) and the predictions from the deep learning model are displayed in 2, 3 and 4. Next, some of the predictions are selected (2) and the data in the dimensionality reduction plot is highlighted (3). Some regions in the scatter plot are selected and the corresponding predictions are marked in the blocks view (4). Finally, selecting a prediction block makes the input view display the corresponding input (5), which can be analyzed to determine if the prediction is correct.*

Abstract

The usage of deep learning models for tagging input data has increased over the past years because of their accuracy and high-performance. A successful application is to score sleep stages. In this scenario, models are trained to predict the sleep stages of individuals. Although their predictive accuracy is high, there are still misclassifications that prevent doctors from properly diagnosing sleep-related disorders. This paper presents a system that allows users to explore the output of deep learning models in a real-life scenario to spot and analyze faulty predictions. These can be corrected by users to generate a sequence of sleep stages to be examined by doctors. Our approach addresses a real-life scenario with absence of ground truth. It differs from others in that our goal is not to improve the model itself, but to correct the predictions it provides. We demonstrate that our approach is effective in identifying faulty predictions and helping users to fix them in the proposed use case.

CCS Concepts

• ***Human-centered computing*** → ***Visual analytics;***

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

1. Introduction

The usage of deep learning (DL) has notably increased in the past years due to its effectiveness to solve problems of different nature. The applications of DL models are many and can be found in a wide range of contexts. For example, they have proved to be effective for many image-analysis tasks: object recognition [ZF14], image captioning [CLZ15, FGI*15, VTBE15], image segmentation [NDL*05] or image classification [KSH12], to name a few. Another successful domain is the medical field, wherein DL models were developed to help practitioners with their daily tasks: lung nodules detection and classification [HHH*15], or nuclei detection and classification [SRT*16, CROMO13, ABA*16].

One important field within the medical domain is the study of sleep. In this context, individuals are subject to polysomnography (PSG) tests when they are believed to be suffering from sleep disorders. The PSG involves measuring brain signals, which are recorded by electroencephalography (EEG) and analyzed afterwards by an expert. This expert is in charge of scoring the PSG by tagging pieces of the whole recording as sleep stages. This manual approach is time consuming and labor intensive [BKS*17], making it hard to apply at a large scale. To overcome this limitation, a lot of research has been performed to automate sleep scoring tasks, for example, by using DL models [SDWG17, BKS*17, LKL12, ZWBC16, TMGZ16]. The automation of the scoring process has obvious benefits regarding time and effort. However, it also brings drawbacks in terms of reliability and accuracy of results.

In the medical field, it is even more important than in other fields that models produce correct outputs to solve other complex, human dependent tasks (e.g., diagnosis). Although models provide certainty for the predictions, it does not depict actual validity that can be used to ensure any correctness. In addition, in real-life scenarios there is a lack of ground truth, making it nearly impossible to ascertain whether a prediction is correct or not. As a result, a reviewing process is necessary to ensure a certain degree of correctness. This reviewing process eliminates all the benefits of automation because it requires an inspection of the whole output space.

In state-of-the-art work, there are many tools that support the development of DL models [KAKC18, PHVG*18, SGPR18, MCZ*17, LSL*17, LSC*18, WSW*18, KJFF15]. Generally, they enable users to see whether a model performs correctly in terms of predictive accuracy, for example, by finding superfluous layers or deficiencies in the training data. Nevertheless, all these tools are applied in a development stage with the aim of improving a model. In contrast, our work focuses on a real-life scenario in which we have an imperfect model and no ground truth any longer. Therefore, we aid users in an exploratory process to find the potentially misclassified predictions. Our approach does not aim to discover the cause of the misclassification.

To tackle the difficult task of finding misclassifications in a real-life scenario, we present *V-Awake*, a visual analytics approach that aids users to find, store, analyze and correct faulty predictions from DL models. Our contributions are: **1)** We present the first visual analytics system for deep-learning based sleep staging, and **2)** We apply visualization in the absence of ground truth, i.e., real-life data, to accelerate detection of misclassification in deep-learning based sleep staging.

We conducted our research with a sleep scoring model trained on raw, single EEG channel data [SDWG17]. We demonstrate the usability of our approach in a concrete, real-life use case together with two somnologists. We discuss the limitations of our approach, how it can be generalized to other domains that use DL models, and we give directions for future research.

2. Medical Background

The study of sleep is an important area in medical research. It can reveal disorders, such as apnea, narcolepsy, parasomnia or hypersomnia, which can also relate to other types of medical conditions such as psychiatric disorders, neurodegenerative diseases [WGWF10] or cardiovascular disorders [SWA*08]. Therefore, having a good understanding of sleep is crucial to provide better diagnoses of some diseases.

The current procedure to study sleep patterns in clinical settings consists of several steps. First, a PSG is performed to record brain signals, eye, chin and leg movements, blood oxygen level, heart rate and breathing of the patient. After the recordings have been obtained, a PSG technologist determines sleep stages by applying rules defined in one of the major sleep scoring guidelines such as the Rechtschaffen and Kales [RK68] or the American Academy of Sleep Medicine (AASM) [BBG*12]. The stages are usually tagged in 30 seconds segments of the PSG, which are called *epochs*. Subsequently, a sleep doctor uses this information to make a diagnosis.

The main drawback of such an approach is the amount of time needed to score sleep stages. For instance, a technologist may spend over one hour to score an 8-hour PSG [BKS*17] due to the labor-intensive nature of the scoring process that involves the analysis of several indicators. In Fig. 2 five examples of EEG signals are shown depicting the five sleep stages described in the AASM manual [BBG*12]. Stages *N1*, *N2* and *N3* represent the non-rem stages, and *REM* indicates the rapid-eye-movement phase of the sleep. Each stage is characterized for having different morphological characteristics going from lighter to deeper sleep respectively. As can be seen, the distinction between different sleep stage patterns can already be hard when analyzing the signals in isolation. In addition, technologists have many other parameters to be considered (e.g., movement sensors, oxygen level in blood, breathing rhythm, etc.), increasing the complexity even further.

Our approach aims to aid experts that score PSGs (i.e., technologists) at correcting the output of DL models for sleep stage scoring.

3. Related Work

Most of the visualization approaches available in the literature focus on the *development* of a DL model. Generally, the goal is to find issues in training/validation data, or architectural deficiencies like superfluous layers, non suitable activation functions, etc., that can be used to modify the initial model to create an improved version of it. In this section, we provide an overview of techniques that deal with understanding models, paying special attention to DL models. Our proposed approach distinguishes itself from previous work in that we work with a real context that lacks ground truth. This scenario has not yet been considered in the literature [HKPC18].

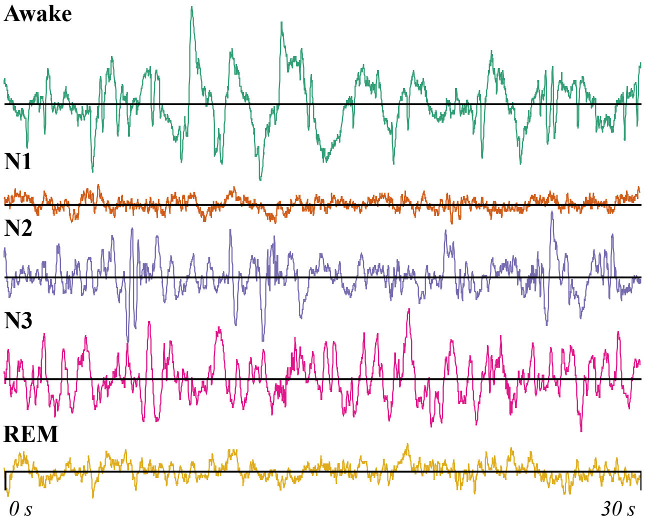


Figure 2: *Five 30-second epochs depicting the five different sleep stages described by the AASM. Stages from top to bottom represent respectively lighter and deeper sleep. Stages* N1 *to* N3 *depict the* non-rem *stages, while* REM *stage depicts the rapid-eye-movement phase of sleep. Signals are recordings from* Fpz-Cz *derivation [KZT*00].*

3.1. Performance Analysis

Work has been done on performance analysis of predictive models from a general perspective, focusing on the visual exploration of several performance indicators. ModelTracker [ACD*15] and Squares [RAL*17] are two systems that intend to provide insights into the performance of classifiers. Although they share a common goal, their approaches differ. The former system presents both training and test data, enabling users to label data as positive or negative, tag groups and link them through iterations of the model. Squares, for its part, strives to analyze the performance of multi-class classifiers. To this end, it visually presents the results of the validation data in a parallel coordinate plot fashion in which each column represents a class. This enables a comparison of the performance per class. The two systems differ from ours in that they address scenarios with ground truth available, and their goal is to analyze performance of models.

3.2. Neural Network Analysis

Much work has been done on understanding convolutional neural networks (CNN) [LSL*17, KAKC18, PHVG*18] and recurrent neural networks (RNN) [KJFF15, SGPR18, MCZ*17, SGB*19].

Their main goal is to provide insight into what networks learn. To this end, some techniques make use of 2D projections in combination with labeled data in order to find what Liu et al. [LSL*17] call *pure* and *impure* clusters. These cluster types indicate good or bad splitting of the data respectively. Therefore, they can be used to investigate how the model performs. ActiVis [KAKC18] uses 2D projections of the activations of several layers to determine whether the model learned how to properly split the input data into classes or not. Similarly, DeepEyes [PHVG*18] utilizes projections to identify stable layers. This system aims at helping during the training process, whereas ActiVis focuses its analysis in a post-training step. Interestingly, Rauber et al. [RFFT17] conducted several experiments on different datasets to demonstrate the usability of projections to evaluate how well the models learned to split the data. All these systems utilize 2D projections in conjunction with ground truth, that is, labeled data, whereas our approach is meant to use those projections solely due to the absence of ground truth. Therefore, we assume the network is able to produce good splittings which can be used for further analysis.

3.3. Model Interpretation

In some cases, experts use models to perform complex tasks like segmentation of anatomical structures or risk monitoring. In this context, models provide predictions or alarms based on given data. Inspection of model output is needed to ensure quality and be aware of possible misbehaviors. The work of Raidou et al. [RMB*16] presents a system that aims to help clinicians to understand segmentation models. It enables the exploration of errors in the segmentation to find patterns that can help evaluate the reliability of the model. The previous work uses labeled data to guide the exploration. In other scenarios, the only available data is the steps performed by the model. The work of Scheepens et al. [SMvdWvW15] aims at visualizing the rationale of a reasoning engine that is fed with possibly unreliable sources. Due to the nature of unreliability, experts require a support system to discard possible false alarms.

Both examples reflect an actual necessity to support users when using a model in a real-life scenario. Nevertheless, the concepts introduced in these works cannot be translated directly to the sleep staging problem nor to DL models.

3.4. Explanation Techniques

Explanation techniques are used in complex systems to provide a better understanding of DL models. This area has recently drawn attention due to the necessity for experts to explain how models work. For providing explanations on CNNs, a great amount of work has been done [SVZ13, SDBR14, MV16, ZBL*18]. We focus on the techniques that are most closely related to our approach.

Fong et al. [FV17] compute a perturbation mask that indicates ranges of the input space that were salient for the model when making a prediction. Other studies address the same problem with different approaches. For example, Grad-CAM [SCD*17] tries to find salient regions in the input space by means of gradients applied to the last convolutional layer of a CNN. It generalizes an earlier work that introduced a method to compute the so-called Class Activation Maps (CAM) [ZKL*16], which also depicts an approach to find saliency regions. The main drawback of CAM is that it is restricted to CNNs that do not include fully-connected layers. Another approach was introduced by Zeiler et al. [ZF14] to find salient regions by occluding parts of the input and attaching a *deconvolution* net to the model we want to analyze. All these techniques share a common goal, although their methods differ.

Regarding explanation techniques for RNNs, Van der Westhuizen et al. [vL17] apply existing saliency methods like the ones

© 2019 The Author(s)

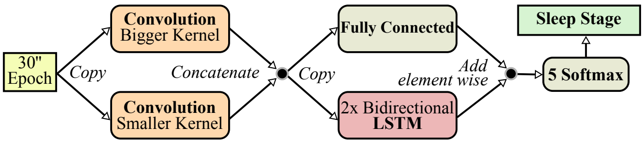


Figure 3: *Deep learning model for sleep stage scoring [SDWG17]. It comprises two convolutional branches with different kernel sizes, a shortcut connection and two bidirectional LSTM layers.*

described previously to temporal inputs (electrocardiogram). They found that deletion masks provided the best results and saliency regions matched medical concepts like types of waves that are used to recognize patterns. Regarding temporal inputs and hybrid models that combine convolutional and recurrent layers, recent work [GCWG18] analyzes saliency approaches and shows that they do not suffice to provide good explanations. Through visualization, they demonstrate that more research is needed to better understand how this type of model works with temporal input data.

4. Problem Definition

We define the problem of correcting predictions in a neural network model by means of tasks that depict the main goals in our system. Our goal is not to improve a given model in terms of predictive accuracy. Rather, it is to find incorrect predictions in environments in which ground truth is missing to enable users to correct and indicate what the prediction truly is. In this section, we firstly introduce a description of the model and data that we use in our approach. Next, we define a set of tasks and give a brief description of them.

4.1. Model Description and Dataset

To introduce our approach, we use a DL model [SDWG17] that scores sleep data. A graphical, high level description is shown in Fig. 3. It has two convolution branches with different kernel sizes: a smaller one to capture temporal information (i.e., EEG patterns) and a larger one to capture frequency information (i.e., frequency components). The derived features are then concatenated and fed to the recurrent part of the model, which is formed by two bidirectional long short-term memory (LSTM) layers. A residual learning approach is used, which is stated by the fully connected layer parallel to the bidirectional LSTMs, to keep track of the features extracted in the convolution step. Finally, all these activations are added up and fed to a fully connected layer with a 5-softmax activation function that serves to normalize the output into a probability distribution of five classes. The model is trained in two steps using data from a sleep study [KZT*00] available on PhysioNet [GAG*00]. In the first step the representation learning (i.e., convolutional layers) is done. Next, a residual learning approach is used to train the two LSTM layers as well as the shortcut connection (i.e., fully connected layer in Fig. 3). Once the model is trained, it can be used without necessity to retrain. In this model, convolutional layers act as feature extractors directly from the raw input signal, while LSTM layers learn transition rules between sleep stages. The model achieves an accuracy of 82.0% [SDWG17].

The data used to train the model represents the sleep recordings of 20 patients (from subject *SC4001E0* to subject *SC4192E0*) over two nights. A depiction is shown in Fig. 4. For each patient j and session k, a signal $f_{j,k}(t)$ is measured, where $t \in \mathbb{Z}$ indicates a point in time in seconds. The signal is sampled at a frequency of $100Hz$, resulting in 100 measurements per second. $E_{i,j,k} = [f_{j,k}(30i), f_{j,k}(30i+1), \cdots, f_{j,k}(30i+29)]$ represents the i-th epoch, which is a 30-value vector for patient j and session k. Each epoch $E_{i,j,k}$ is 30 seconds long, accumulating a total of 3000 values. Finally, $C_{i,j,k}$ indicates the corresponding classification for the i-th epoch of patient j in session k.

The signal f is gathered from sensors placed on the head of the patient during sleep. On average, there are 1075 epochs per patient and session, resulting in 1075 predictions over approximately 9 hours of sleep and above 3 million points. Examples of signal f for each sleep stage are shown in Fig. 2. Besides all this data, we retrieve the following information from the model:

Probabilities for each possible class are provided by DL models in classification. Thus, a function $P(E_{i,j,k})$ provides a vector $P_{E_{i,j,k}}$ of probabilities where $P^c_{E_{i,j,k}} \in [0,1]$ depicts the likelihood of epoch $E_{i,j,k}$ being classified as class c. Analogous, $P^c_{E_{i,j,k}} = P(C_{i,j,k})$ where c is the class predicted for epoch $E_{i,j,k}$, that is, the class with the highest probability.

Activation Maps, also named feature maps, depict the output produced by a certain layer l in a DL model after applying an internal function. This function depends on the type of layer. For instance, convolutional layers apply *convolution* over the input data to derive new features, i.e., produce an output. As evident, these features are used to determine the classification of unseen, new input data. Therefore, they can be used to discover similarities on predictions. The function $A(E_{i,j,k}, l)$ retrieves the activation map for epoch $E_{i,j,k}$ and layer l, containing a variable number of activations $u_{i,j,k,l}$.

Saliency Maps describe how important the attributes of the input

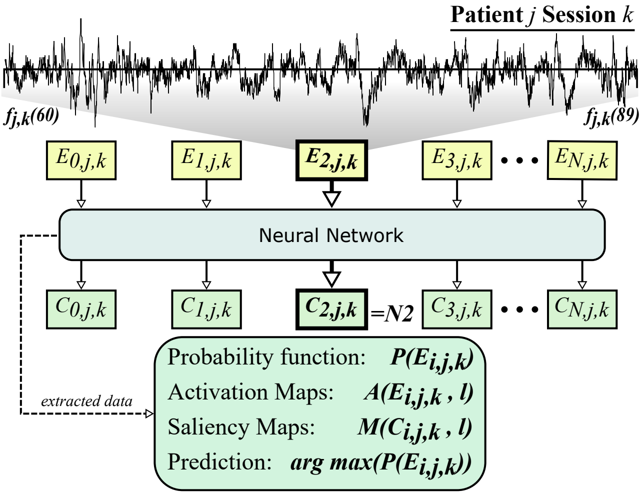


Figure 4: *Illustration of the data used in our approach for patient j and session k. An example of signal is given for epoch $E_{2,j,k}$ which is classified as stage* N2 *after being run through the model.*

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

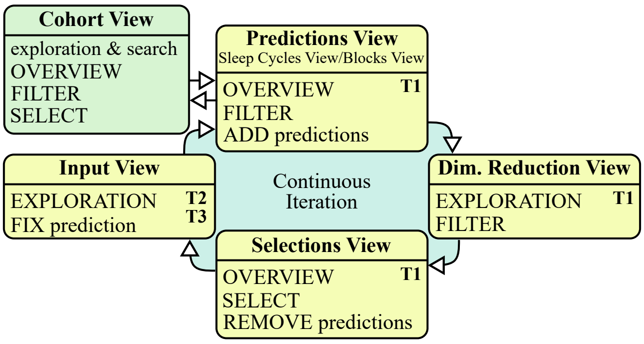


Figure 5: *Depiction of our workflow and the mapping to the views of our design. Arrows depict a common way of interaction, although other routes are possible. Upper case words summarize the most important actions performed in each view. The tasks that each view performs are also shown in the diagram.*

data are for a layer of the model to predict that input as a particular class. The function $M(C_{i,j,k},l)$ provides the saliency map for the epoch corresponding to classification $C_{i,j,k}$ and layer l. In our case, each saliency map contains a constant number of values $v^m_{i,j,k,l} \in [0,1]$ with $m \in [0,\cdots,2999]$, that indicates how important the m-th value is for layer l to classify $E_{i,j,k}$ as $C_{i,j,k}$. Our approach uses Grad-CAM [SCD*17] to compute saliency maps. The dimension of the map this method gives depends on the output size of the layer. However, to keep a constant size, the values of the output are rescaled with a linear interpolation to match the size of the input instances, i.e., 3000 values.

4.2. Tasks

Based on multiple interviews with two somnologists and four DL experts, we define a set of tasks that are considered relevant for the analysis of DL predictions for sleep scoring:

T1 Fix incorrect predictions. Incorrect predictions are a serious problem. Finding them is not a trivial task when there is lack of ground truth. Hence, users, independently of their expertise, should be enabled to explore the data in such a way that they can find potentially incorrect predictions and repair them by indicating the actual class.

T2 Understand why the model made a prediction. Once potentially incorrect predictions are found, it is necessary to understand why the model made such a prediction. This helps users to understand whether the prediction is correct or not.

T3 Re-tag predictions with basic support. The system must allow users to re-tag selected predictions. Hints must be provided to users to help them make a decision.

We designed a workflow to support the defined tasks (see Fig. 5). Users can perform actions in whichever order they decide.

5. V-Awake

In this section we introduce the main components of our approach (see Fig. 6 for an overview). Although it was designed for sleep experts, the components are generic enough to be used by experts with different backgrounds (**T1**).

5.1. Cohort View

The primary goal of the cohort view (Fig. 6.1) is to provide a summary of the data for each patient j and session k. The summary displays information regarding the gender of the patient, identifier and length of the recording session, and distribution of the predictions $C_{i,j,k}$. This provides an overview that can be examined to spot interesting cases for experts.

The summary of the predictions plays an important role because it might show particularities that are of interest (e.g., absence of predictions of a particular class, or an abnormal class distribution). It is depicted with a horizontal stacked bar chart per patient. The width of each bar encodes the ratio of predictions of a class relative to the total number of predictions.

The ultimate goal from a usability perspective is that the user selects a case of interest. To facilitate it, *search*, *sort* and *layout* features are provided. Regarding the latter feature, users can select a stage of interest to be placed as the first element in the stacked bar chart to make comparisons between different subjects.

Below the panel that displays all the patients, two stacked bar charts are shown in the same fashion. The top one shows the distribution of predictions for the patient selected in the cohort view. The bottom one shows the aggregated distribution for all the patients. The location of both charts facilitates comparison of the local (i.e., selected patient) and global (i.e., all patients) distributions. Moreover, the summary view (Fig. 6.1.1) provides information regarding the results of the model for the validation dataset. A confusion matrix is shown, which can be used by users to determine the cases that the model fails more often. When a patient is selected, estimated values are provided. These values are computed by interpolating the global values from the validation set of the model.

5.2. Predictions View

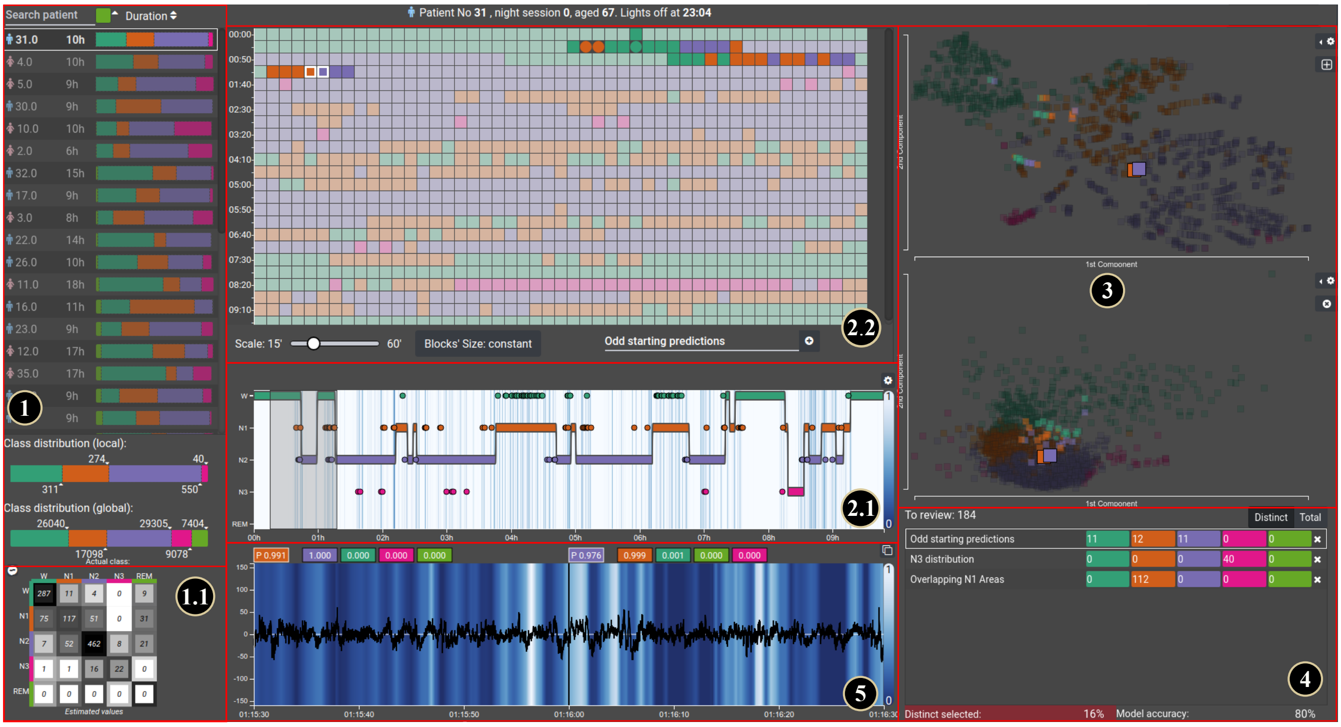
Predictions are the core items in our work. Our approach provides two different ways to directly interact with them. They are discussed in the following subsections.

5.2.1. Sleep Cycles View

The sleep cycles view is presented in Fig. 6.2.1 and provides an overview of the whole sleep session in a familiar manner to the expert. It emphasizes the transitions between sleep stages. This *piano roll* representation enables users to spot interesting patterns quickly. The view is based on a time series chart where x and y axes denote time relative to the beginning of the recording and sleep stage, respectively. Colors are used to encode stages. The background displays the fluctuations on the certainty of predictions (i.e., probabilities $P(C_{i,j,k})$), without interfering with the core part of the view. Fluctuations can be utilized by the expert to spot regions in which the model was less certain and therefore prone to misclassifications.

To prevent visual clutter in the global trend, a preprocessing step is applied to the data to extract possible outliers. It consists of extracting consecutive prediction sequences that belong to the same

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.



Search patient	Duration
31.0	10h
4.0	10h
5.0	9h
30.0	9h
10.0	10h
2.0	6h
32.0	15h
17.0	9h
3.0	8h
22.0	14h
26.0	10h
11.0	18h
16.0	11h
23.0	9h
12.0	17h
35.0	17h
	9h
	9h

Class distribution (local): 274, 40, 311, 550

Class distribution (global): 26040, 29305, 7404, 17098, 9078

Actual class:

	W	N1	N2	N3	REM
W	287	11	4	0	9
N1	75	117	51	0	31
N2	7	52	462	8	21
N3	1	1	16	22	0
REM	0	0	0	0	0

Estimated values

To review: 184

Distinct Total

Odd starting predictions	11	12	11	0	0
N3 distribution	0	0	0	40	0
Overlapping N1 Areas	0	112	0	0	0

Distinct selected: 16% Model accuracy: 80%

Figure 6: *User interface of V-Awake. Numbers depict the views and components of our approach: cohort view (1) and confusion matrix (1.1), predictions view with sleep cycles view (2.1) and blocks view (2.2), dimensionality reduction view (3), selections view (4) and input view (5).*

class and contain less predictions than a set threshold. The threshold can be adjusted by the user. For example, users can set a lower value to extract outliers that form very quick transitions. This helps to find cases in which the model rapidly changes stages, potentially indicating that there are misclassifications. The visual encoding in the sleep cycles view highlights transitions that should not occur in a normal context. In a normal sleep pattern, transitions should happen in a specified order. For instance, it is not possible to immediately move from *awake* to *REM*. Outliers are visually represented as dots visually disconnected from the main trend, which is represented as a *piano roll*. This particular visualization supports the task of spotting faulty predictions (**T1**).

The view relies on brushing to focus on a specific area of the predictions space. The other views are updated accordingly restricting further actions to the selected area. Furthermore, when an action is performed in other components, the sleep cycles view updates accordingly by visually de-emphasizing corresponding predictions.

5.2.2. Blocks View

Similar to the sleep cycles view, the blocks view (Fig. 6.2.2) depicts an overview of all the predictions generated for a selected patient and session, which are represented as blocks and are placed sequentially from the top-left corner to the bottom-right corner. The major difference with previous is the visual encoding and the interactions. While the sleep cycles view emphasizes transitions between stages, this focuses on the sequentiality of the predictions. The blocks view serves two main purposes:

1. **Give an overview of the predictions in constant time intervals**. Intervals directly connect with medical concepts (e.g., *N1* should last up to 7 minutes). To provide more flexibility, time intervals are adjustable by users, allowing the exploration of the predictions from different time perspectives.
2. **Highlight the predictions that are under consideration**. The location of this view is ideal for depicting the predictions that are filtered out from other components. This allows users to be aware of the time position of the predictions that are selected after performing brushing in other views (see Fig. 6). Moreover, users can directly add or remove elements by clicking them. This provides fine-grained control over the elements that are currently selected. This control is useful to incorporate or exclude predictions that are located nearby in time and that the expert considers to be interesting for a further analysis.

The size of the blocks can encode extra information such as the probability and the entropy of a prediction. The latter is defined as:

$$-\sum_{c=0}^{n} P^{c}_{E_{i,j,k}} \cdot \log_n P^{c}_{E_{i,j,k}},$$

where n is the number of classes. This encoding helps to visually identify predictions that deviate from others in terms of probability. The probability metric emphasizes high probability predictions, while entropy emphasizes extreme cases in which the probability values are very similar.

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

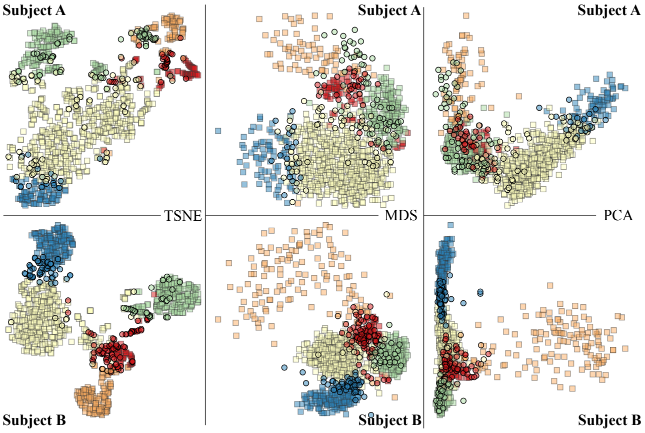


Figure 7: *Comparison of tSNE, MDS and PCA for two subjects in a setting with ground truth available. Each square represents a prediction, the color depicts the predicted sleep stage. Circles with a black border represent predictions from the model that do not match the label assigned as ground truth. tSNE works, in general, better than other methods to identify borderline cases.*

5.3. Dimensionality Reduction View

The dimensionality reduction view (Fig. 6.3) depicts a scatter plot with a dimensionality reduction computed over all the activations of a layer on the predictions of a given patient and session. It is used to find incorrect predictions by identifying visual overlaps (i.e., *impure clusters*). The data displayed in this view is defined as:

$$\sigma([A(E_{0,j,k},l),A(E_{1,j,k},l),\cdots,A(E_{N,j,k},l)],n_c),$$

where σ represents a dimensionality reduction function and n_c is the number of principal components that σ provides. The rationale for the election of a layer is that each layer in a DL model learns different features from data. The layers close to the output are believed to effectively separate the features linearly [DJV*14].

We considered three dimensionality reduction functions: Principal Component Analysis (**PCA**) [Pea01, Hot33, Jol11], Multidimensional Scaling (**MDS**) [Kru64] and t-distributed Stochastic Neighbor Embedding (**tSNE**) [MH08]. Figure 7 shows a comparison of the them for two different subjects in a context with ground truth. The same model as in our approach is used to compute the projections. Circles with black borders depict misclassified predictions. As can be seen, PCA tends to group cases belonging to the same class and does not discern misclassified cases, hence we discarded this method. On the other hand, MDS and tSNE appear to better divide the space so that boundary cases for incorrect predictions stand out more. tSNE is the default option in this view. The design of the this view addresses task **T1** since the location of the predictions in the plot might provide useful information.

Generally, dimensionality reduction techniques use heuristics to find the most optimal solution. They involve randomization, resulting in a different output every time the method is executed. Even though this randomization has benefits, it can worsen the exploratory process since the user would not get the same result in two different executions. To prevent this from happening, we decided to set the seed to a fixed value. Nevertheless, we provide users with options to set a random or a different value for the seed if desired.

We heavily rely on linking and brushing to help users spot suspicious elements, with the previously introduced components all coupled. Multiple dimensionality reduction plots can be visualized at the same time. They all are coupled, enabling an exploratory process in which we could compare different aspects:

1. **Different layers**. Users might be interested in exploring the dimensionality reduction output for activations of different layers. This is useful, for example, when the model has a hybrid architecture. Exploring the activations of convolution and recurrent layers side by side can lead to interesting findings. By default, the previous-to-last layer is selected because it has been shown that it performs the best for these methods.
2. **Number of principal components**. In most of the cases the use of two components enables a fairly sufficient exploration. However, when this is applied, the sequential nature of the predictions is lost. To tackle this, in our approach it is possible to switch from 2 to 1 principal component, used for the vertical axis, while the horizontal axis is used for the sequence number of the epoch. When this occurs, the plot adapts the axes to either show both components, or the only main component together with the sequence number of the prediction.

5.4. Selections View

To facilitate addressing task **T1**, we provide a mechanism to store selections of predictions. They are collected and presented in this view (see Fig. 6.4). Selections are depicted by indicating the occurrences of predictions for each class. To enable a quick identification, a textual label is displayed together with the summary. Labels are defined by the user at the creation of a selection.

An indicator depicting the number of selected predictions is also shown. It reflects either an absolute or distinct count of predictions. The latter count is also used to compute a ratio over the total. The ratio is used to visually inform the user in case it goes above a threshold. The threshold is determined by subtracting 100 and the model's accuracy percentage. This acts as a *warning* to keep the number of selected predictions low. The rationale for this threshold is that, assuming that the model's accuracy is similar to the one obtained for the evaluation data, then it should make around the same percentage of incorrect predictions in a real-life scenario.

5.5. Input View

Understanding why the DL model made a prediction (**T2**) is addressed with this view (Fig. 6.5). By visualizing an instance of input data (i.e., an epoch $E_{i,j,k}$), a resolution can be made to decide the correctness of a prediction (i.e., a classification $C_{i,j,k}$).

Our approach displays the values of signal f for a given epoch. Also, consecutive epochs can be displayed side-by-side at the same time to give a notion of context to the users. Furthermore, users can move forwards and backwards to retrieve the next or the previous prediction's input respectively. This can be performed by pressing the right or left arrow on the keyboard. We visualize epochs by means of traditional line plots, with a fixed scale for the y-axis to facilitate the comparison of signals that have different amplitudes. Although this information is enough for the expert to infer the stage that an epoch represents, we also provide some clues on what the model was *seeing* at the moment of making a decision. To this end, our approach provides the corresponding values of a saliency map. By default, the saliency maps $M(C_{i,j,k}, l)$ and $M(C_{i,j,k}, l')$, where l and l' represent two convolutional layers from two convolutional branches of our model, are averaged. The layer parameter can be adjusted to visualize any convolutional layer of the model, or a combination of them. Saliency maps are encoded as one-dimensional heat maps. This enables an easy exploration of the salient regions of the input, without disturbing the visualization of the input data itself. This design addresses task **T2**.

Finally, the re-tagging task (**T3**) can be performed in this view. To quickly change the class of a prediction, glyphs are presented on the top of the view. Colors encode the class that a glyph represents. We also use position and text to encode some other information:

- The glyph at the left-most position indicates the current value associated to the prediction. In case it is the original prediction, it is marked with the label *P*, which stands for *Prediction*. If the class was changed by the user, the label *F* is used, which stands for *Fixed*. Moreover, the glyph also shows the probability that the model produced by a text label.
- The other glyphs are slightly separated from the previous ones. This emphasizes that the probabilities they depict are normalized with respect to their values. This is helpful because the model we use tends to produce high-probability predictions. As a result, the probabilities for the rest of stages are extremely low, making it hard to enable a comparison between them. By normalizing their values, an easier comparison can be made by the user.

When a prediction is corrected, the visual encoding in all the other displays changes accordingly to indicate so. This is helpful to avoid the user from revisiting corrected predictions.

6. Use Case

Sleep is a natural process consisting of transitions between sleep stages that tend to follow rules. Alterations in the transitions can be an indicator of a sleep disorder. These alterations can be reflected in different manners: a longer duration of a particular stage (e.g., stage *N1* should last up to 7 minutes), continuous changes between stages (e.g., from *N2* to *awake* and vice versa) or the absence of some stage (e.g., *REM*). These patterns might represent either the effect of a sleep disorder, or problems in our predictive model to correctly classify sleep stages. The main goal of the exploration is to find out possible misclassifications and fix them.

We demonstrate our approach by means of a use case on a single dataset [KZT*00]. The subjects considered for our use case range from *SC4201E0* to *SC4822GC*. These subjects were not considered in the training phase of the DL model. Note that the original source of the dataset also provides the ground truth, which we use in a posterior stage to calculate the percentage of *actual* misclassifications found in the exploratory use cases. We show how components in our approach together with domain knowledge from experts enable the identification of possible misclassifications as well as interesting patterns from the cohort.

6.1. Exploration Patient 1

The exploration (see Fig. 6 for an overview) was conducted with the help of the somnologists. We picked a patient that seemed interesting because of the deviations in the distribution of sleep stages. The expert remarked that it was unusual to not have a single prediction of *REM* stage. This could be because the patient has some disorder, or because the model was wrong when making predictions.

REM stage usually happens after *N3*, or after a short period of *N2*. In Fig. 2, we can see that *N1* and *REM* are somewhat similar in shape. Therefore, it may be that the model had difficulties to distinguish them. We started the exploration by narrowing the analysis to consider *awake*, *N1* and *N2*. We noticed some interesting patterns in the sleep cycles view. For instance, we saw that slightly two hours after the start of sleep, there was a noticeable drop in the probability prediction and there were some outliers from *N1* (see Fig. 8 S2). We selected that region for further analysis. After moving forward in the sleep cycles view, we saw two other interesting patterns: quick alternations between *awake* and *N1*. This pattern was seen between four and five hours after the beginning of sleep (see Fig. 8 S3). The pattern looked suspicious because of the quick changes between stages. We selected and saved them for analysis.

At the beginning of the sleep cycles view, there were two periods of *awake* followed by some predictions of *N1* and *N2* (see Fig. 8 S1). We noticed that the probability of the model dropped in that region considerably. When we selected slightly more than the first hour of sleep, the dimensionality reduction plot showed some overlapping areas that looked suspicious. We selected and saved those areas. Nearly at the end of the sleep record, there was a period of *N3* predictions (see Fig. 8 S4). According to the expert, this was suspicious because *N3* tends to shrink during the night. We selected all these predictions for further analysis. Figure 9 shows some of the overlapping areas for selections *S1*, *S2*, *S3* and *S4*. The brushed predictions are potential misclassifications.

At that point, we focused only on the dimensionality reduction plot to observe the whole picture. We observed sparse predictions of class *N1* and *N3* in an area principally covered by predictions of class *N2*. Therefore, we selected and saved predictions belonging to classes *N1* and *N3* in the overlapping area (see Fig. 9 S5).

After performing the selections, we ended up with 311 predictions. While reviewing the input data for each prediction, we an-

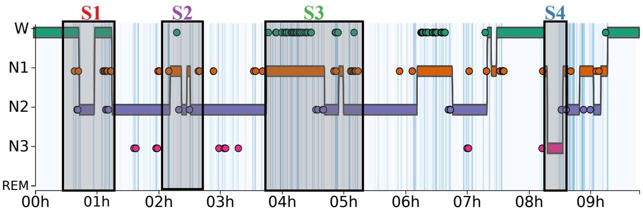


Figure 8: *The four selections made in the exploratory use case.*

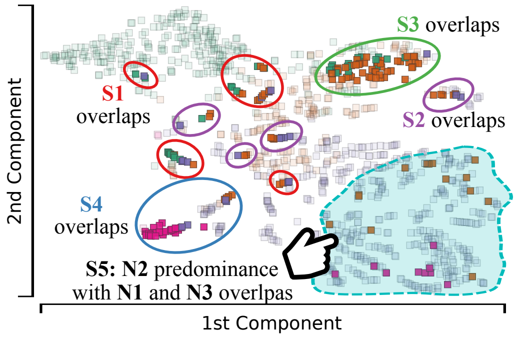


Figure 9: *Selection of possible misclassifications. For instance,* S5 *depicts an area of* N2 *predominance, but* N1 *and* N3 *predictions are found.*

notated 217 as misclassified, which represented the 69% of the whole selection. Posterior analysis using the ground truth showed that there were 249 predictions that were actually misclassifications. Therefore, we found 87% of the misclassifications with our approach. It is important to remark that the information about the number of misclassifications was not available during the exploration of the data. During annotation of the block of *N3* predictions nearly at the end of sleep, we discovered that they were misclassified because the input signals seemed to contain an artifact (see Fig. 10) following a very regular pattern. The expert indicated that this might be due to a problem with the location of the sensors that were interfering with the movements of the eyes.

The needed time to analyze the plots and to select pieces of data was about 5 minutes. We do not measure the time to analyze each individual epoch. However, if we consider the time proposed by the sleep scoring manual [BBG*12], which recommends to invest up to 2 seconds per epoch, it would result in above 10 minutes. If we apply the same time per epoch for the whole output space, it would result in investing 39 minutes. Therefore, we would roughly save 24 minutes for this particular subject.

6.2. Exploration Patient 2

For the second exploration, we used the same patient and session as shown in the video demonstrating the usage of our approach. This session contained 775 predictions. We immediately observed that the sleep cycles seemed to be more uniform. It could also be seen in the locations of predictions in the dimensionality reduction view, which were better localized forming clusters-like structures.

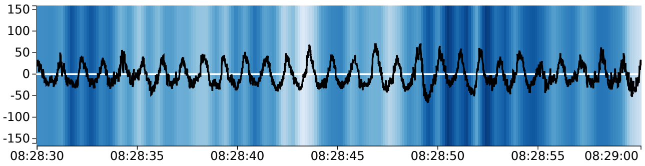


Figure 10: *Strange artifact found during exploratory process. It is too regular and free of noise to depict a bio-signal.*

In this case, we took advantage of the outliers from the sleep cycles view and the location of predictions in the dimensionality reduction plot. The sleep cycles view showed some interesting areas that contained outliers. They were seen after one hour and a half (a *N2* sequence with *N1* and *N3* outliers), slightly before two hours (a *REM* sequence with *N1* and *N2* outliers), around three hours (a *N2* sequence with *N3* and *REM* outliers) and so on. When selecting each of these areas individually, we observed the corresponding predictions in the dimensionality reduction view. We observed some overlapping areas in this plot. They could be an indicator of misclassifications, thus we selected and saved them.

After repeating this process iteratively, we created a global selection with 57 predictions. This represented the 7% of the total output space. Out of those predictions, we found 35 misclassifications. Posterior analysis using the ground truth showed that there were 63 misclassifications. Therefore, we were able to find the 55% of misclassifications. As for the previous patient case, this information was not available beforehand.

7. Discussion and Limitations

The use case depicted in Section 6 shows how our approach can be used by experts to utilize domain knowledge to guide exploration towards finding misclassifications.

Finding misclassifications when there is no ground truth is an unsolvable problem per se. Visually exploring and analyzing predictions can shed light and ensure a certain degree of correctness. The ability to select parts of data based on observations enables experts to incrementally cover most of the misclassified predictions. The tight interaction between components helps to verify earlier assumptions (e.g., quick transitions between stages might represent misclassifications).

As for other approaches, ours has limitations. For instance, the usage of our system does not guarantee the discovery of all the misclassifications. The interaction and exploration of predictions cannot always lead to finding all the faulty predictions, and in some cases it might become difficult to understand the dimensionality reduction. Moreover, the dimensionality reduction might be ineffective if the model produced poor separations of the feature space. This limitation is not specific to our approach but inherent to the dimensionality reduction approach.

We performed an informal qualitative evaluation of our approach with both somnologists and DL experts. They found our approach useful and helpful to fill the gap in current settings in which a DL model is used. They expressed that being able to see the context of the predictions in different forms was very helpful. For instance, linking the time-location of a prediction with its location in the dimensionality reduction plot was useful to better analyze incorrect predictions. They also found views and visualizations of our approach appropriate for sleep experts. They stated that, after some explanation, the dimensionality reduction view was understandable and useful for spotting incorrect predictions. Also, the ability to create selections that matched hypotheses was an interesting way of addressing the problem. Finally, they also remarked that visualizing the input for a particular prediction in conjunction with the saliency map was useful to understand what the model was recognizing in signals. Having a smaller version of the sleep cycles view in the cohort view was proposed by one of the somnologists. This could help spot interesting patients from an overview of the transitions between sleep stages. However, it might be difficult to visualize due to the size of the components.

The somnologists also pointed at the lack of a mechanism to filter cases with some interesting properties. Filtering cases in which the input signal is mostly 0 volts would be desirable because it might reflect a misplacement or disconnection of the electrodes. The experts would also like to use the system to better understand the model. The interest of the experts arose when they observed the input of incorrect predictions. Even though this is not the goal of our system, we certainly believe this would greatly improve the system.

When the somnologists were asked whether they could use data corrected with our approach despite the fact it cannot be guaranteed to be fully correct, one stated the following: *"It really, really depends on where the errors lie. If there is some misclassification between N1 and N2, it probably will not affect the clinical interpretation too much. But if all errors converge to misclassifying REM as something else, it will affect it. As another example; if epochs are misclassified as wake sparsely through the hypnogram, it may result in a very fragmented looking hypnogram, which may be classified as abnormal; while if the same amount of (mistaken) wake epochs are grouped into 2 or 3 a little bit longer periods of consolidated wake, it may look normal to a doctor"*. We believe these issues are addressed by our approach since users can spot suspicious patterns, analyze them in terms of activations from the DL model and inspect corresponding input data. From the comments of the somnologist we can argue that the number of found misclassifications is not crucial as long those that could affect the diagnosis of sleep disorders are analyzed.

7.1. Approach Generalization

Deep learning models can be used to detect cancer tissue in video frames (e.g., colonoscopy [UTA*18]). Generally, the traditional approach works by examining a patient with a tubular camera that explores the colon while the doctor analyzes the video in real time. Deep learning can be incorporated in this process to aid doctors during examination. Our approach can be generalized to be applied in this scenario. Sleep staging and cancer detection share the characteristic of having sequences of predictions of a recording session. The sleep staging model uses epochs, while the cancer detection model uses video frames. Although the nature of the data is different, they have the temporal aspect in common. Both tasks aim to classify input sequences in a small set of classes and both scenarios are divided by patients and sessions. To generalize our approach, it would need to handle video frames keeping the rest of the system intact.

7.2. Scalability

The bottleneck of our approach is the dimensionality reduction. Our implementation is able to compute tSNE over one thousand of records with thousands of dimensions in a few seconds. If the number of dimensions was higher, a pre-process step could be applied to randomly project dimensions to a lower space, feeding the result to the dimensionality reduction technique. This approach seems to be effective for tSNE as Donahue et al. stated [DJV*14]. In our approach, we handle over 1000 samples per patient on average. In case this number was too high to be handled, the analysis could be performed by examining chunks of thousands of samples per time. The major drawback of this approach would be to ensure that we have enough representatives of each class in each step. On the visual aspect, the main concern is the number of classes that our approach is able to present. In the case of sleep scoring, there are only five different classes. However, in other domains this number might be substantially higher.

8. Conclusions and Future Work

In this work, we have presented *V-Awake*: a visual analytics approach to find and correct faulty predictions in real-life scenarios. It is a novel visual analytics approach that combines different visual and interactive components to enable users to effectively find and correct predictions in a sleep staging context.

We have demonstrated the usability of our approach in a use case. It shows that our approach can be used to find suspicious patterns that can represent misclassifications. Besides, a generalization of our approach has also been proposed, stating that it can be transferred to other domains with minor modifications. The discussion with experts reflected a real interest from them in our approach as well as ways to improve it. Although our tool does not guarantee a perfect correction, it does enable experts to analyze interesting patterns to make sure that a proper diagnosis can be performed afterwards.

As future work, we would like to investigate how to incorporate active learning such that users could reinforce the model with our approach. This idea requires more research to ensure that the learning process provides benefits instead of creating a bias in the model that deteriorates the predictive accuracy. We also plan to extend the approach such that users can explore the dimensionality reduction plot from a higher level, that is, focusing on the entire cohort population rather than a single patient. We expect this to provide some further insight into patients that have more faulty predictions. Our idea is that we could apply a dimensionality reduction method over the entire population to find groups of patients that share similar peculiarities in terms of activations of layers. With this, users would only have to analyze some representative subjects from a particular cluster and apply the learned facts to the rest (e.g., majority of faulty predictions in stage *REM*, similar sleep patterns, etc.).

Acknowledgments

The authors would like to thank Dr. Sebastiaan Overeem from Kempenhaeghe Hospital in The Netherlands and Dr. Gary Garcia-Molina, from Philips Research North-America, for their valuable feedback, comments and time. This research was performed within the framework of the strategic joint research program on Data Science between TU/e and Philips Electronics Nederland B.V.

References

[ABA*16] ALBARQOUNI S., BAUR C., ACHILLES F., BELAGIANNIS V., DEMIRCI S., NAVAB N.: Aggnet: deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE transactions on medical imaging 35*, 5 (2016), 1313–1321. 2

[ACD*15] AMERSHI S., CHICKERING M., DRUCKER S. M., LEE B., SIMARD P., SUH J.: Modeltracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015), ACM, pp. 337–346. 3

[BBG*12] BERRY R. B., BROOKS R., GAMALDO C. E., HARDING S. M., MARCUS C., VAUGHN B., ET AL.: The aasm manual for the scoring of sleep and associated events. *Rules, Terminology and Technical Specifications, Darien, Illinois, American Academy of Sleep Medicine* (2012). 2, 9

[BKS*17] BISWAL S., KULAS J., SUN H., GOPARAJU B., WESTOVER M. B., BIANCHI M. T., SUN J.: Sleepnet: Automated sleep staging system via deep learning. *arXiv preprint arXiv:1707.08262* (2017). 2

[CLZ15] CHEN X., LAWRENCE ZITNICK C.: Mind's eye: A recurrent visual representation for image caption generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 2422–2431. 2

[CROMO13] CRUZ-ROA A. A., OVALLE J. E. A., MADABHUSHI A., OSORIO F. A. G.: A deep learning architecture for image representation, visual interpretability and automated basal-cell carcinoma cancer detection. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (2013), Springer, pp. 403–410. 2

[DJV*14] DONAHUE J., JIA Y., VINYALS O., HOFFMAN J., ZHANG N., TZENG E., DARRELL T.: Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning* (2014), pp. 647–655. 7, 10

[FGI*15] FANG H., GUPTA S., IANDOLA F., SRIVASTAVA R. K., DENG L., DOLLÁR P., GAO J., HE X., MITCHELL M., PLATT J. C.: From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 1473–1482. 2

[FV17] FONG R. C., VEDALDI A.: Interpretable explanations of black boxes by meaningful perturbation. *arXiv preprint arXiv:1704.03296* (2017). 3

[GAG*00] GOLDBERGER A. L., AMARAL L. A., GLASS L., HAUSDORFF J. M., IVANOV P. C., MARK R. G., MIETUS J. E., MOODY G. B., PENG C.-K., STANLEY H. E.: Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation 101*, 23 (2000), e215–e220. 4

[GCWG18] GARCIA CABALLERO H. S., WESTENBERG M. A., GEBRE B.: Plainability: Explainability for one dimensional temporal inputs of deep learning models. *Demo at the Workshop on Visualization for AI explainability (VISxAI)* (2018). Advance online publication. 4

[HHH*15] HUA K.-L., HSU C.-H., HIDAYATI S. C., CHENG W.-H., CHEN Y.-J.: Computer-aided classification of lung nodules on computed tomography images via deep learning technique. *OncoTargets and therapy 8* (2015). 2

[HKPC18] HOHMAN F. M., KAHNG M., PIENTA R., CHAU D. H.: Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Transactions on Visualization and Computer Graphics* (2018). 2

[Hot33] HOTELLING H.: Analysis of a complex of statistical variables into principal components. *Journal of educational psychology 24*, 6 (1933), 417. 7

[Jol11] JOLLIFFE I.: Principal component analysis. In *International encyclopedia of statistical science*. Springer, 2011, pp. 1094–1096. 7

[KAKC18] KAHNG M., ANDREWS P. Y., KALRO A., CHAU D. H. P.: Activis: Visual exploration of industry-scale deep neural network models. *IEEE transactions on visualization and computer graphics 24*, 1 (2018), 88–97. 2, 3

[KJFF15] KARPATHY A., JOHNSON J., FEI-FEI L.: Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078* (2015). 2, 3

[Kru64] KRUSKAL J. B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika 29*, 1 (1964), 1–27. 7

[KSH12] KRIZHEVSKY A., SUTSKEVER I., HINTON G. E.: Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (2012), pp. 1097–1105. 2

[KZT*00] KEMP B., ZWINDERMAN A. H., TUK B., KAMPHUISEN H. A., OBERYE J. J.: Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg. *IEEE Transactions on Biomedical Engineering 47*, 9 (2000), 1185–1194. 3, 4, 8

[LKL12] LÄNGKVIST M., KARLSSON L., LOUTFI A.: Sleep stage classification using unsupervised feature learning. *Advances in Artificial Neural Systems 2012* (2012), 5. 2

[LSC*18] LIU M., SHI J., CAO K., ZHU J., LIU S.: Analyzing the training processes of deep generative models. *IEEE transactions on visualization and computer graphics 24*, 1 (2018), 77–87. 2

[LSL*17] LIU M., SHI J., LI Z., LI C., ZHU J., LIU S.: Towards better analysis of deep convolutional neural networks. *IEEE transactions on visualization and computer graphics 23*, 1 (2017), 91–100. 2, 3

[MCZ*17] MING Y., CAO S., ZHANG R., LI Z., CHEN Y., SONG Y., QU H.: Understanding hidden memories of recurrent neural networks. *arXiv preprint arXiv:1710.10777* (2017). 2, 3

[MH08] MAATEN L. V. D., HINTON G.: Visualizing data using t-sne. *Journal of machine learning research 9*, Nov (2008), 2579–2605. 7

[MV16] MAHENDRAN A., VEDALDI A.: Salient deconvolutional networks. In *European Conference on Computer Vision* (2016), Springer, pp. 120–135. 3

[NDL*05] NING F., DELHOMME D., LECUN Y., PIANO F., BOTTOU L., BARBANO P. E.: Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing 14*, 9 (2005), 1360–1371. 2

[Pea01] PEARSON K.: Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2*, 11 (1901), 559–572. 7

[PHVG*18] PEZZOTTI N., HÖLLT T., VAN GEMERT J., LELIEVELDT B. P., EISEMANN E., VILANOVA A.: Deepeyes: Progressive visual analytics for designing deep neural networks. *IEEE transactions on visualization and computer graphics 24*, 1 (2018), 98–108. 2, 3

[RAL*17] REN D., AMERSHI S., LEE B., SUH J., WILLIAMS J. D.: Squares: Supporting interactive performance analysis for multiclass classifiers. *IEEE transactions on visualization and computer graphics 23*, 1 (2017), 61–70. 3

[RFFT17] RAUBER P. E., FADEL S. G., FALCAO A. X., TELEA A. C.: Visualizing the hidden activity of artificial neural networks. *IEEE transactions on visualization and computer graphics 23*, 1 (2017), 101–110. 3

[RK68] RECHTSCHAFFEN A., KALES A.: A manual for standardized terminology, techniques and scoring system for sleep stages in human subjects. *Brain information service* (1968). 2

[RMB*16] RAIDOU R. G., MARCELIS F. J., BREEUWER M., GRÖLLER M. E., VILANOVA A., VAN DE WETERING H. M.: Visual analytics for the exploration and assessment of segmentation errors. *VCBM 16* (2016), 7–9. 3

[SCD*17] SELVARAJU R. R., COGSWELL M., DAS A., VEDANTAM R., PARIKH D., BATRA D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV* (2017), pp. 618–626. 3, 5

[SDBR14] SPRINGENBERG J. T., DOSOVITSKIY A., BROX T., RIEDMILLER M.: Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806* (2014). 3

[SDWG17] SUPRATAK A., DONG H., WU C., GUO Y.: Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg. *IEEE Transactions on Neural Systems and Rehabilitation Engineering 25*, 11 (2017), 1998–2008. 2, 4

[SGB*19] STROBELT H., GEHRMANN S., BEHRISCH M., PERER A., PFISTER H., RUSH A. M.: Seq2seq-vis: A visual debugging tool for sequence-to-sequence models. *IEEE transactions on visualization and computer graphics 25*, 1 (2019), 353–363. 3

[SGPR18] STROBELT H., GEHRMANN S., PFISTER H., RUSH A. M.: Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE transactions on visualization and computer graphics 24*, 1 (2018), 667–676. 2, 3

[SMvdWvW15] SCHEEPENS R., MICHELS S., VAN DE WETERING H., VAN WIJK J. J.: Rationale visualization for safety and security. *Computer Graphics Forum 34*, 3 (2015), 191–200. 3

[SRT*16] SIRINUKUNWATTANA K., RAZA S. E. A., TSANG Y.-W., SNEAD D. R., CREE I. A., RAJPOOT N. M.: Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE transactions on medical imaging 35*, 5 (2016), 1196–1206. 2

[SVZ13] SIMONYAN K., VEDALDI A., ZISSERMAN A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013). 3

[SWA*08] SOMERS V. K., WHITE D. P., AMIN R., ABRAHAM W. T., COSTA F., CULEBRAS A., DANIELS S., FLORAS J. S., HUNT C. E., OLSON L. J., ET AL.: Sleep apnea and cardiovascular disease: An american heart association/american college of cardiology foundation scientific statement from the american heart association council for high blood pressure research professional education committee, council on clinical cardiology, stroke council, and council on cardiovascular nursing in collaboration with the national heart, lung, and blood institute national center on sleep disorders research (national institutes of health). *Journal of the American College of Cardiology 52*, 8 (2008), 686–717. 2

[TMGZ16] TSINALIS O., MATTHEWS P. M., GUO Y., ZAFEIRIOU S.: Automatic sleep stage scoring with single-channel eeg using convolutional neural networks. *arXiv preprint arXiv:1610.01683* (2016). 2

[UTA*18] URBAN G., TRIPATHI P., ALKAYALI T., MITTAL M., JALALI F., KARNES W., BALDI P.: Deep learning localizes and identifies polyps in real time with 96% accuracy in screening colonoscopy. *Gastroenterology* (2018). 10

[vL17] VAN DER WESTHUIZEN J., LASENBY J.: Techniques for visualizing LSTMs applied to electrocardiograms. *ArXiv e-prints* (May 2017). 3

[VTBE15] VINYALS O., TOSHEV A., BENGIO S., ERHAN D.: Show and tell: A neural image caption generator. In *Computer Vision and Pattern Recognition (CVPR)* (2015), IEEE, pp. 3156–3164. 2

[WGWF10] WULFF K., GATTI S., WETTSTEIN J. G., FOSTER R. G.: Sleep and circadian rhythm disruption in psychiatric and neurodegenerative disease. *Nature Reviews Neuroscience 11*, 8 (2010), 589. 2

[WSW*18] WONGSUPHASAWAT K., SMILKOV D., WEXLER J., WILSON J., MANÉ D., FRITZ D., KRISHNAN D., VIÉGAS F. B., WATTENBERG M.: Visualizing dataflow graphs of deep learning models in tensorflow. *IEEE transactions on visualization and computer graphics 24*, 1 (2018), 1–12. 2

[ZBL*18] ZHANG J., BARGAL S. A., LIN Z., BRANDT J., SHEN X., SCLAROFF S.: Top-down neural attention by excitation backprop. *International Journal of Computer Vision 126*, 10 (2018), 1084–1102. 3

[ZF14] ZEILER M. D., FERGUS R.: Visualizing and understanding convolutional networks. In *European conference on computer vision* (2014), Springer, pp. 818–833. 2, 3

[ZKL*16] ZHOU B., KHOSLA A., LAPEDRIZA A., OLIVA A., TORRALBA A.: Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 2921–2929. 3

[ZWBC16] ZHANG J., WU Y., BAI J., CHEN F.: Automatic sleep stage classification based on sparse deep belief net and combination of multiple classifiers. *Transactions of the Institute of Measurement and Control 38*, 4 (2016), 435–451. 2