MASTER

Making Sense of Hypnograms

van der Woerd, Caspar

Award date:
2021

Making Sense of Hypnograms

Master Thesis

Caspar van der Woerd

Eindhoven University of Technology
Philips Sleep & Respiratory Solutions
Kempenhaeghe Center for Sleep Medicine

Supervisors:
Dr. Pedro Fonseca
Prof. dr. Sebastiaan Overeem
Dr. ir. Stef van den Elzen
Humberto Garcia Caballero

Eindhoven, August 2021

Abstract

Good sleep is important for overall well-being and reduces the risk for many physical and mental health conditions. Polysomnography is the gold-standard for measuring sleep and is used to identify the sleep structure of a subject, which is depicted in a hypnogram. Even though the hypnogram plays an important role in diagnosing sleep disorders, interpretation by physicians typically relies on clinical intuition, experience and visual pattern recognition. Therefore, in this thesis we aimed to identify and interpret aspects within and across hypnograms that contribute to interpretation by physicians.

We describe how subjective evaluations for hundreds of hypnograms were obtained and how visualization and machine learning methods were applied to gain insight into interpretation of the hypnogram. Conventional machine learning models and convolutional neural networks were used to identify features of the hypnogram that are associated with interpretation by physicians. In addition, visualization methods were used to obtain more qualitative insights into the data and the models. Our results show how fragmentation and distribution of sleep stages in the hypnogram is strongly associated with interpretation by physicians. The implications of our work in the area of sleep research are discussed.

Acknowledgements

A year ago, I would have described my desired master thesis as a project that would involve a combination of data visualization and machine learning, applied in an interesting domain and conducted at an inspiring company. Therefore, I am incredibly grateful that I got the opportunity to do this project at Philips that ticks all of these boxes.

First of all, I would like to thank my supervisors Pedro Fonseca, Stef van den Elzen, Sebastiaan Overeem and Humberto Garcia Caballero for their guidance and valuable feedback throughout the project. The freedom that you gave me to shape the research in my own way provided a great learning opportunity. During our weekly meetings you often challenged me to dig deeper and think broader, which I appreciate and which resulted into a work that I personally feel proud of.

Furthermore, I would like to thank Fokke van Meulen for helping me hands-on with all practicalities at Kempenhaeghe. I want to thank the physicians that were involved in the project for their interest in the study and for their help in obtaining the required data. Also, I want to thank all those at the visualization research group, sleep research group and at Philips who attended my presentations, the feedback that you gave and the resulting discussions motivated me to continue.

Lastly, I would like to thank my girlfriend Senna for being there for me during these past months and for providing a temporary, sunny change of home in "working from home". And finally, I would like to thank my parents for supporting me in pursuing my own path and in the decisions that I took during my studies.

Contents

List of Acronyms

AUC Area Under the Curve.

CAM Class Activation Maps.
CNN Convolutional Neural Network.

DL Deep Learning.
DTW Dynamic Time Warping.

FCN Fully-Convolutional Network.
FPR False Positive Rate.

GAP Global Average Pooling.

LCM Least Common Multiple.
LIME Local Interpretable Model-Agnostic Explanations.

ML Machine Learning.
MLP Multilayer Perceptron.

N1 Non-Rapid Eye Movement 1.
N1 Non-Rapid Eye Movement 3.
N2 Non-Rapid Eye Movement 2.

NREM Non-Rapid Eye Movement.

PSG Polysomnography.

R&K Rechtschaffen & Kales.
ReLU Rectified Linear Unit.
REM Rapid Eye Movement.
ResNet Residual Network.
RNN Recurrent Neural Network.
ROC Receiver Operating Characteristic.

SDB Sleep Disordered Breathing.
SHAP SHapley Additive exPlanations.

T-SNE T-distributed Stochastic Neighbor Embedding.
TPR True Positive Rate.
TSC Time-Series Classification.

W Wake.

Chapter 1

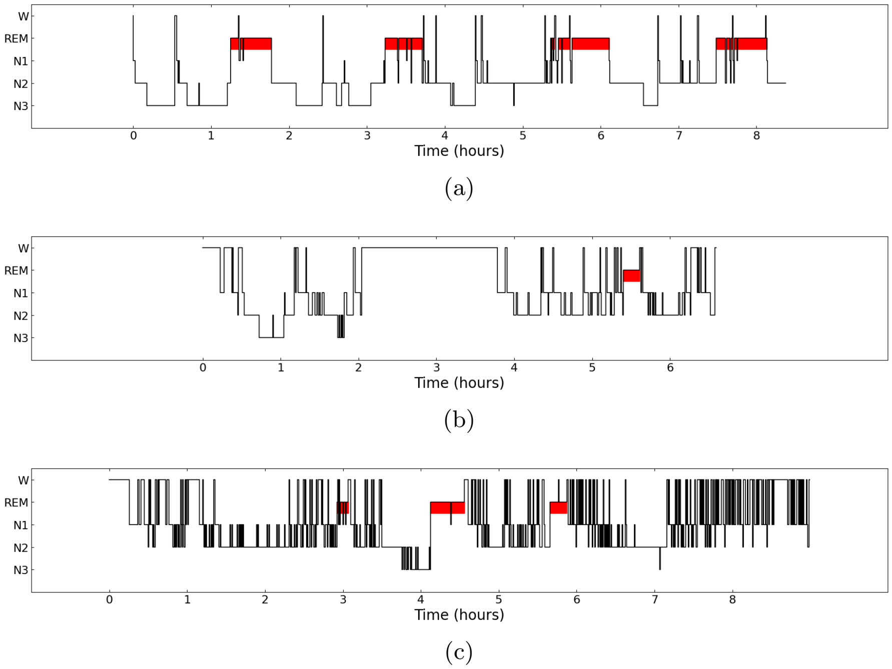
Introduction

Humans spend almost one-third of their life asleep, nevertheless there are still a lot of mysteries surrounding sleep. From an evolutionary perspective, it seems weird that our bodies spend large amounts of time asleep, a state in which we are vulnerable to predators. Nevertheless, the importance of sleep is indisputable. Good sleep has been shown to be important for overall well-being and to reduce the risk for many physical and mental health conditions [49].

The field of sleep medicine is specialized in diagnosis and therapy of disturbed and disordered sleep. Sleep is typically measured using polysomnography (PSG), which remains the current gold-standard for assessing sleep. PSG focuses on the measurement and analysis of brain activity, eye movements and muscle activity to determine whether a subject is awake or asleep, and in which sleep phase someone is. The signals obtained from PSG are split into epochs (20-30 second frames). Using a set of scoring rules, each epoch is visually inspected and classified as wake (W), non-rapid eye movement (NREM, subdivided into N1, N2 and N3) or rapid eye movement (REM) sleep. The resulting sequence of sleep stages can be visualized in a so-called **hypnogram**, which displays the sleep structure of a patient throughout the night.

Three examples of hypnograms are shown in figure 1.1. There are strong differences in patterns that can be observed in these hypnograms. From a clinical perspective, the hypnogram contains a wealth of information. It is one of the most important aspects of the PSG report and is used in combination with a patient's background, experienced symptoms and other information to determine if and which sleep disorder is present and how this should be treated.

Despite being an important aspect of the PSG report, assessment of the hypnogram by physicians is largely done visually in combination with "clinical intuition". There is only a limited set of quantitative variables extracted from the hypnogram, examples are the total sleep time and the percentage of REM sleep. Each parameter summarizes the hypnogram as a single number. These parameters might be unable to explain the full complexity that is conveyed by the hypnogram. Instead, the most important information conveyed by the hypnogram is retrieved based on visual pattern recognition by the physician.



(a)

(b)

(c)

Figure 1.1: Three example hypnograms showing the sleep structure (sequence of sleep stages) throughout the night obtained from PSG.

Research on interpretation and clinical relevance of the hypnogram is limited and the exact role of the hypnogram in diagnosing sleep disorders is not well understood. Some works have considered automatic detection of specific aspects (e.g. insomnia, sleep-disordered breathing) from the hypnogram [44, 9]. To the best of our knowledge, there is only one study that tried to relate features of the hypnogram with subjective evaluations by physicians. In a small experiment, it was shown that interpretation of a hypnogram in terms of normal or abnormal could be predicted accurately from the distribution of sleep stages in the hypnogram [4].

In this thesis, we aim to provide a more thorough exploration of the role of the hypnogram and how it is used by physicians. For this purpose, a large amount of hypnograms was assessed by a small group of physicians. Combining machine learning and visualization methods can yield solutions that are highly effective at gaining insight in complex data that contains a temporal dimension [2]. Therefore, we aim to use a combination of visualization and machine learning techniques to explore patterns in and across hypnograms. More specifically, we are interested in patterns in hypnograms that are associated with interpretation by physicians and agreement between physicians. These goals will be formalized in the next section.

1.1 Thesis Objectives

Before elaborating on the specific objective of this thesis, it should be noted that this project is by nature exploratory. We are unaware of any work that explored the role of the hypnogram and the relations between hypnogram, interpretation and diagnosis in a similar way. Therefore, the objective of this thesis is quite broad. Overall, our main objective can be described as:

> *To use visualization and (explainable) machine learning to identify and interpret aspects within and across hypnograms that contribute to interpretation by physicians.*

Moreover, four research questions were formulated to make this objective more concrete and to help us in exploring the patterns within and across hypnograms:

1. How can visualization and machine learning be used to gain insight into a large number of hypnograms?
2. Which features of the hypnogram drive interpretation by physicians?
3. Are there previously unknown features of the hypnogram that are associated with interpretation? Can these features be used for analysis and/or assessment of hypnograms?
4. Which factors determine and influence disagreement between physicians and certainty within physicians?

1.2 Thesis Scope

Normally, a physician would use a hypnogram in a clinical setting and take into consideration experienced symptoms, background of a patient and other PSG outcomes. In this research, we are emphasizing the structure of the hypnogram itself, therefore the hypnograms are considered in an artificial research setting rather than a clinical setting. Activities and information before the hypnogram is obtained (e.g. raw PSG signals) and other PSG outcomes (e.g. apneas) are considered out of scope for the current study. The hypnograms that are used were obtained from PSGs that were conducted and scored by experienced sleep technicians at Kempenhaeghe as part of the SOMNIA and Healthbed projects [48]. These PSG recordings were scored using the American Academy of Sleep Medicine (AASM) scoring rules [7]. Only subjects between 18 and 80 years are considered.

1.3 Thesis Contributions

In this thesis, we describe how we used visualization and machine learning to identify and interpret aspects within and across hypnograms that contribute to interpretation by physicians. We collected subjective evaluations for hypnograms and conducted several experiments to obtain insights in the data.

In chapter 2, we discuss relevant background literature and identify methods from visualization and machine learning that are suitable for dealing with hypnograms. Afterwards, in chapter 3, we describe the methods that were used to obtain and analyse hypnograms and evaluations. The results are presented in chapter 4. Finally, in chapter 5, we provide a discussion on these results, on the limitations of our work and on directions for future research.

Chapter 2

Background

This chapter will highlight some relevant background work. First, relevant work from the domain of sleep will be discussed in section 2.1, where we will also further elaborate on hypnograms. In section 2.2, methods from visualization and machine learning that are applicable to hypnograms are identified. Specific attention is paid to time-series since the time dimension of a hypnogram makes it a particularly challenging type of data. A summary is given in section 2.3

2.1 Sleep

As mentioned before, PSG is the gold-standard method for assessing sleep. The signals obtained from PSG are loaded into the computer and each epoch (20-30 second time frame) is visually inspected and manually classified as one of the sleep stages. There are different sets of rules for scoring these epochs; Rechtschaffen and Kales (R&K) which distinguishes 7 stages and the more novel AASM that distinguish 5 stages [29, 7]. As mentioned before, the stages distinguished by AASM are W, N1, N2, N3 and REM, a brief overview of these stages is presented in table 2.1. The resulting hypnogram h can be seen as a sequence of sleep stages, this is formalized in equation 2.1, here t is typically around 1000 assuming an 8-hour long recording.

$$h := \langle s_1, s_2, ..., s_t \rangle \text{ where each } s_i \in \{W, N1, N2, N3, REM\} \tag{2.1}$$

In a hypnogram showing a normal sleep structure we would expect to see a cyclical pattern, where cycles are approximately 90 minutes. Typically, four to six cycles are observed where NREM is followed by a period of REM sleep [8]. REM sleep is usually observed more frequently in the second half of the night, whereas deep sleep (N3) is more profound during the first half. The hypnogram in figure 1.1a illustrates these patterns, N3 is more profound at the start and the amount of REM increases towards the end, four two-hour cycles of NREM followed by REM sleep can be distinguished.

Name	**Type**	**Time**	**Description**
W	Wake		Person is awake.
N1	NREM	5%	Feeling drowsy and dozing off, light sleep, short transition state from W to N2. Easy to wake up.
N2	NREM	50%	Most frequent sleep stage during the night. Characterized by k-complexes and spindles in the EEG.
N3	NREM	20%	Deep sleep, person is difficult to wake up. Characterized by slow wave delta activity in the EEG.
REM	REM	25%	Characterized by rapid eye movements. High amounts of activity and irregularity, brain paralyzes the muscles.

Table 2.1: Overview of different sleep stages distinguished by the AASM manual. Estimated times and descriptions are obtained from [10]. Note that the times per stage are approximate averages; there are large individual differences in how long a person spends in each stage during the night.

The times shown in table 2.1 are average estimates, in practice there are strong individual differences in sleep structure. A known factor that has an influence on sleep structure is age. The amount of N3 reduces with age, for elderly it is not surprising to see minimal amounts of deep sleep [8]. Moreover, it is important to be aware that an abnormal sleep structure does not imply that a person has a sleep disorder and vice versa. For example, in some cases abnormal sleep structures can be explained as a first-night effect (a person has a bad night due to the PSG setting).

The results of a PSG are presented in a report, which includes the hypnogram and a number of parameters that are computed from the hypnogram. These parameters quantify certain characteristics of the hypnogram. Typical parameters include, amongst others, distribution of stages, number of awakenings and sleep onset latency (minutes from lights off till first non-wake). In the PSG report typically only a small amount of parameters is presented, however for quantitative analysis more features can be included [4, 9].

There are some limitations to the hypnogram. First, the choice and definition of scoring rules is important. Parameters, such as sleep onset latency, were found to be significantly different for hypnograms obtained from the same PSG with different scoring rules (AASM vs R&K) [29]. Moreover, the scoring rules are not always easy to apply and leave room for interpretation for the scoring sleep technician. A comparison of scored stages by sleep technicians of eight European sleep laboratories in 2004, showed that the inter-rater agreement between technicians was limited [13]. The technicians scored a large sample of hypnograms with various disorders, the average agreement between the scorers was only 76.8%. Five years later, a similar study found that the inter-rater agreement between scorers was 82% for AASM and 80.6% for R&K [12]. This implies that for a given PSG, the resulting hypnogram can vary across scorers. Therefore, methods that automatically score the epochs of a PSG can be more robust. For example, it was shown that automatic insomnia detection from hypnograms was more accurate when the PSG was not scored by a sleep technician, but rather by an automatic sleep staging model [9]. Those results emphasize that the hypnogram is not an exact and perfectly accurate representation of sleep structure. Instead, it is subject to human choices and noise introduced by representing the sleep structure as a discrete sequence.

Consequently, novel approaches considered alternative representations for sleep structure. In [32], an automatic sleep staging model was proposed that is not limited to 30 second epochs but can provide predictions at higher frequencies. The authors illustrated that their method can provide additional diagnostic value by showing that higher frequency predictions can lead to a more accurate separation of OSA from healthy patients. Similarly, a hypnodensity plot can be used to represent the distribution of sleep stages at each timestamp rather than a single sleep stage per timestamp [43]. However, these methods are currently only of scientific interest. In practice the hypnogram is still the default method for visualizing the sleep stages discovered in a PSG.

2.2 Time-series

Time-series are a type of data where observations are recorded over time. Typical examples of time-series are ECG recordings and stock-market data, but also multimedia such as audio and video can be seen as time-series [30]. Since many real-world phenomena change over time and recent advances allow for large-scale storage of data, time-series has grown into a large research area. Hypnograms can also be seen as time-series, more specifically as a univariate discrete time-series since each timestamp describes a single, discrete value. Research on time-series considers many different tasks including (sub)sequence matching, anomaly detection, clustering, classification, visualization and forecasting [1]. In this section we will mostly focus on classification since the goal of identifying patterns in the hypnogram that are associated with interpretation can be formulated as a classification problem.

Methods for time-series problems focus on dealing with the temporal dimension of time-series. However, the structure of the individual observations at each timestamp can be different (as seen from the obvious differences between a video and a hypnogram). Therefore, it is challenging to evaluate algorithmic advances as their success might not generalize well across time-series problems [5]. As a solution, the UCR archive[1] was created in 2002 and has grown to a total of 128 time-series datasets since [14]. The archive contains a variety of time-series with different characteristics. In recent years, also multivariate time-series were introduced as part of the UEA archive [5]. Nowadays, new algorithmic advances for time-series are often tested on the entire archive.

Typically, time-series contain continuous values, in the UCR repository this is true for all datasets. In contrast, hypnograms have discrete values, therefore popular time-series algorithms such as dynamic time warping (DTW) cannot easily be applied on hypnograms [6]. Time-series are typically expensive to

[1] `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/`

collect as it requires longitudinal data collection [51]. This is also observed in the UCR repository most of the datasets contain only a few hundred instances, some exceptions contain more than 1000 instances but overall the datasets are relatively small.

2.2.1 Classification

Classification is a machine learning (ML) problem, where the goal is to learn a model that predicts a label based on some input. For this purpose a model is trained on labeled data, afterwards the model can be used to predict a label for previously unseen instances. For example, one could learn a model to predict whether an ECG signal is associated with a healthy or diseased patient. Here healthy and diseased are the labels and the sequence of ECG values is the input.

In many cases the raw input cannot easily be used to predict the labels. Therefore, traditional approaches compute features (e.g. amplitude of ECG signal) that serve as input for the model. More recently, advances in computing power led to an increasing interest in deep learning (DL). In this area of ML, deep neural networks are used that can automatically identify relevant features from raw input data [23].

In the remainder of this section, the traditional ML approach using handcrafted features and some examples of this in the context of sleep are discussed. Moreover, DL approaches for time-series and the challenge of explainability in DL are discussed.

2.2.1.1 Traditional Machine Learning

As explained in the previous section, traditional machine learning approaches consider handcrafted features that are computed from the raw input. In the context of time-series it is important to realize that computing features over the full series leads to a transformation of the original data to a feature space where there is no longer an explicit temporal dimension. Instead, the temporal aspect of the data is captured in the features themselves. Alternatively, a sliding-window approach can be employed to compute features over subsequences of the series [54, 2, 3]. In this case, several parameters can be tuned such as the size of the (adaptive) window and the overlap between subsequent windows, the result is a matrix of features per window.

The obtained features, either over the full series or using a sliding window, are used to learn a model predicting the target labels. Commonly used and fundamental ML classifier algorithms include (Logistic) Regression, Decision Trees, Naive Bayes, Nearest-Neighbor and Support Vector Machines. Many others or variants on these exist. Since the amount of feature engineering and machine learning approaches is extremely large, we consider this out of scope for this background section. Instead, we will discuss a few concrete examples that use feature engineering and machine learning in the context of hypnograms.

Classifying abnormal hypnograms To the best of our knowledge, the only attempt at relating features of the hypnogram to subjective interpretation by physicians was done by Amouh (2011) as part of a PhD thesis [4]. In their research, 52 hypnograms were considered of which 29 were labelled *abnormal* and 23 were labelled *normal* by domain experts, it is unknown how the hypnograms were selected. The hypnograms were obtained from PSG using the R&K scoring rules. From the hypnogram the following features were computed:

- **Distribution of stages**. Proportion of the recording spent in each stage.
- **Minimum and maximum duration per stage**. For each stage a tuple (min, max) describes the minimum and maximum duration in minutes for the given stage.

The features were used as input for a special type of decision tree classifier that was created as part of their work. A regular decision tree is a hierarchical set of rules that, if followed, predicts one of the labels. A decision tree is typically learned top-down, at each phase a split is created that maximizes the information gain (i.e. split the data into parts using a specific feature such that the parts have minimal entropy with respect to the labels) [36]. The tree that is obtained is pruned to prevent overfitting (i.e. too strictly modelling the training data leading to rules that do not generalize outside the training data). The variation of Amouh involves a decision tree that can handle, what they refer to as *structured* data, this concretely means that the decision tree can handle multiple values at once during a split. For example, the distribution of sleep stages are all used simultaneously in a single split, instead of using a single sleep stage per split.

The non-overfitting model that was learned by Amouh is shown in figure 2.1, it can be observed that the model only considers a single split and uses the distribution of sleep stages for this split. From the model we can see that abnormal hypnograms are in general associated with larger proportions W and S2; on the other hand they have less REM, S1 and deep sleep (S3 and S4). Overall the model achieves a 0.90 accuracy which was evaluated using leave-one-out cross-validation.

Detecting insomnia from hypnogram Another study that used the hypnogram for learning a predictive model was done by Chaparro-Vargas et al. (2016), one of their goals was to detect which subjects were suffering from Insomnia given the hypnograms [9]. For this task they considered the sleep-onset periods, i.e. only the first W, N1 and N2 transitions, of 32 subjects.

In contrast to the work of Amouh, they did not compute features over the entire hypnogram, instead the hypnogram was represented as a transition diagram with stages W, N1 and N2 as shown in figure 2.2. From the hypnogram they learned the probabilities of all 9 pairwise transitions. These transition probabilities were used as input for a logistic regression model to predict whether a subject was suffering from insomnia. Using leave-one-out cross validation the

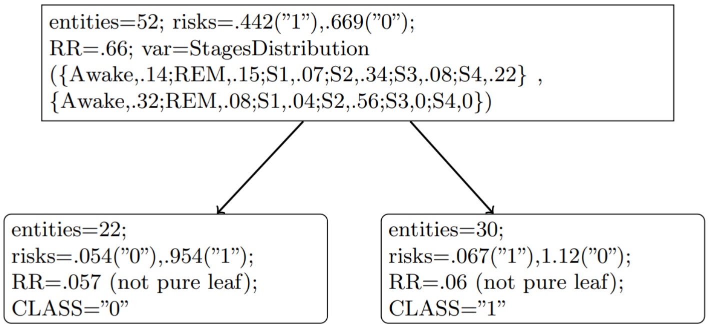

entities=52; risks=.442("1"),.669("0");
RR=.66; var=StagesDistribution
({Awake,.14;REM,.15;S1,.07;S2,.34;S3,.08;S4,.22} ,
{Awake,.32;REM,.08;S1,.04;S2,.56;S3,0;S4,0})
entities=22;
risks=.054("0"),.954("1");
RR=.057 (not pure leaf);
CLASS="0"
entities=30;
risks=.067("1"),1.12("0");
RR=.06 (not pure leaf);
CLASS="1"

Figure 2.1: Decision tree for classifying hypnograms as normal (0) or abnormal (1), directly taken from [4]. The pruned decision tree considers only a single split. The hypnogram is normal if the stage distribution of the hypnogram is closer to $P = \{W, .14; REM, .15; S1, .07; S2, .34; S3, .08; S4, .22\}$ than to $Q = \{W, .32; REM, .08; S1, .04; S2, .56; S3, 0; S4, 0\}$.

model was estimated to have an accuracy of 0.81, with errors equally distributed across the two classes. They concluded that the hypnogram can be used to derive whether a subject is likely to suffer from insomnia.

In a similar study, transitions systems were used to distinguish between subjects with and without sleep-disordered breathing (SDB) [44]. Instead of onset stages, the model considered the states W, REM and NREM. Significant differences in transition probabilities were found between the SDB and no-SDB groups, indicating that this disorder is related to changes in sleep structure.

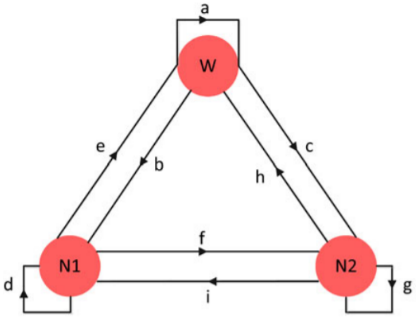


Figure 2.2: Chaparro-Vargas represented the sleep onset period of the hypnogram as a transition system. The probabilities associated with the transitions (edges in the diagram) were used as input for a logistic regression model. Directly taken from [9].

2.2.1.2 Deep Learning

Deep learning (DL) is a subset of machine learning where neural networks are used to automatically extract representations from raw data. In contrast to traditional ML, neural networks can handle raw and complex data and automatically learn suitable abstract representations using backpropagation. These methods have shown remarkable state-of-the-art performance in image recognition, speech processing and many other domains [23].

Fawaz et al. (2019) provide an overview of deep learning applications in time-series classification (TSC) [15]. They argue that deep learning is underutilized for TSC, existing DL methods have benefits over traditional ML approaches. Different types of deep learning models exist, most known are multilayer perceptron (MLP), convolutional neural networks (CNN) and recurrent neural networks (RNN). Despite RNN being known for its ability to handle sequences, CNN is the most successful and most applied model for TSC problems [15]. The reasons for this are that RNNs are mostly useful for sequence-to-sequence applications [22], they suffer from the vanishing gradient problem and are avoided because they are difficult and computationally expensive to train [31].

CNN Convolutional neural networks became popular after AlexNet won the Imagenet competition in 2012 [21]. The main building block of CNN is the convolutional layer which consists of convolution units, often referred to as *filters* or *kernels*. The filter, typically length 3, slides across the time-series, at each point the result of the filter are the input values multiplied by the weights of the filter [15]. Afterwards, this value is transformed using a non-linear activation function such as Rectified Linear Unit (ReLU). A convolutional layer consists of multiple filters that are slided across the input in the same fashion. This means that a convolutional layer with k filters of size w applied on an input of length n produces an output of size $(k, n - w + 1)$. By making use of a non-linear activation function and stacking multiple layers the network can learn complex representations of the input. Typically, the convolutional layers are followed by a number of fully-connected layers.

In a CNN one will also often encounter pooling and dropout layers. The former is used to reduce the dimensionality by reducing an input to local maximum or local average values, this is illustrated in figure 2.3. By downsampling the input with pooling, noise can be surpressed and computational complexity of the model can be reduced. Moreover, dropout layers are often used to prevent overfitting. Dropout is used to randomly ignore nodes during training, thereby approximating a large number of different architectures in parallel [42]. For a more extensive explanation of the workings of CNN in the context of time-series we recommend reading section 2.2.2 of [15].

CNN for TSC Applications of CNN are best-known on images, however the operations of a CNN are also defined on 1-dimensional data. Compared to 2D data such as images, applications of CNN on time-series (or other 1D data)

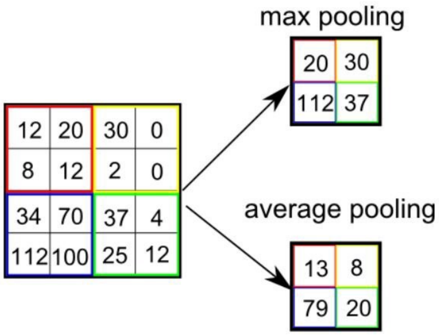
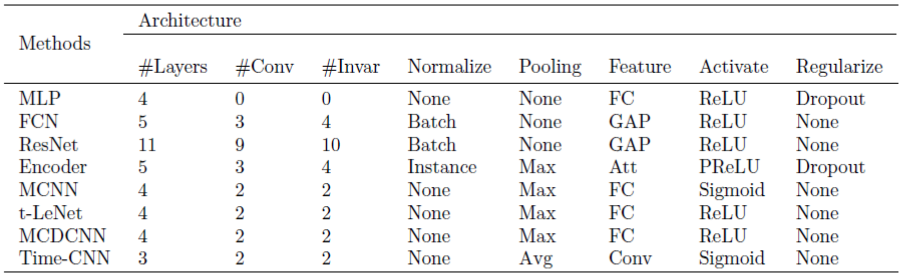


Figure 2.3: Pooling operation on a 2D input. Pooling reduces the size of the input by taking local maximum or average values, directly taken from [37].

are less computationally complex, contain less layers (often 2-3 CNN layers are enough), have much less parameters (typically less than 10.000) and consequently training is faster [19]. In their overview paper, Fawaz et al. compared 9 different deep learning architectures for time-series classification, they compared their performance on all 85 datasets in the UCR repository [15].

A summary of the architectures that were compared is shown in figure 2.4. It can be observed that most of the networks are fairly similar in the sense that they contain 3-5 layers of which 2-3 convolutional layers, only notable exception to this rule is the residual network (ResNet) model proposed by [50]. The differences between the other models concern whether pooling was used after convolutions and whether fully-connected or global average pooling (GAP) layers were used as the last layer(s) of the model.



Methods	Architecture							
	#Layers	#Conv	#Invar	Normalize	Pooling	Feature	Activate	Regularize
MLP	4	0	0	None	None	FC	ReLU	Dropout
FCN	5	3	4	Batch	None	GAP	ReLU	None
ResNet	11	9	10	Batch	None	GAP	ReLU	None
Encoder	5	3	4	Instance	Max	Att	PReLU	Dropout
MCNN	4	2	2	None	Max	FC	Sigmoid	None
t-LeNet	4	2	2	None	Max	FC	ReLU	None
MCDCNN	4	2	2	None	Max	FC	ReLU	None
Time-CNN	3	2	2	None	Avg	Conv	Sigmoid	None

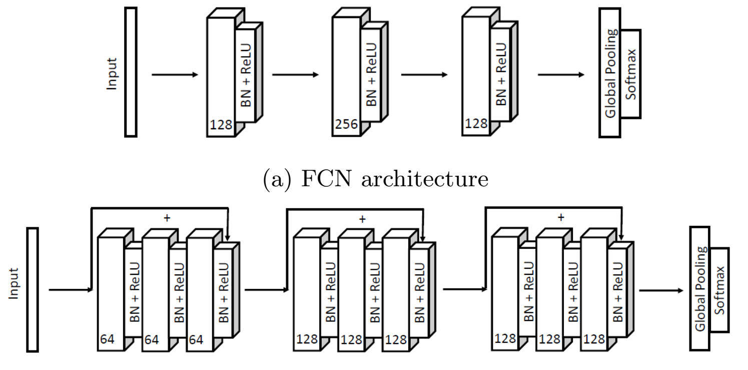
Figure 2.4: Summary of the 9 different DL architectures that were compared on the UCR repository, directly taken from [15].

The results of the comparison of the models on the UCR repository indicated that the fully-convolutional (FCN) and ResNet models were performing significantly better than the others [15]. Interestingly, the ResNet model is the deepest model in their comparison which indicates that deeper models can be more effective for TSC. Moreover, this suggests that the size of the datasets (recall that datasets in the UCR are typically less than 1000 instances) is not a limiting factor. Therefore, this confirms that 1D CNN applications require less data for training compared to their 2D counterparts [15, 50, 19]. Fawaz et al. argue that the success of the FCN and ResNet model can be attributed to the use of a GAP layer, rather than fully-connected layers at the end. Moreover, this adds the benefit of being able to compute class activation maps, on which we will further elaborate later [57].

The FCN and ResNet architecture were both proposed by Wang et al. (2017) [50]. As those models were outperforming all others on almost all datasets of the UCR database, we briefly describe their architectures. The architectures are also shown in figure 2.5.

- FCN stands for fully-convolutional network, this term is used as no pooling layers are used after convolution as seen in figure 2.5a. Instead, the 3 convolutional layers are all followed by batch normalization. The model uses ReLu activation function, GAP is used with softmax for obtaining the final predictions.

- ResNet stands for residual network and has achieved state-of-the art results for object detection problems [18]. The TSC variant of this model, seen in figure 2.5b, is characterized by 3 residual blocks that have shortcut connections to each other. These blocks are followed by a GAP layer and a softmax layer. The shortcuts enable the model to learn residual functions, which has been shown to potentially improve accuracy.



(a) FCN architecture

(b) ResNet architecture

Figure 2.5: The FCN and ResNet architectures for TSC as proposed by [50]. These models were found to be the best performing TSC models on the UCR database [15]. Directly taken from [50].

2.2.1.3 CNN Explainability

The effectiveness of CNN and other deep learning models comes with a downside; as the functions that can be learned are complex and non-linear the models are difficult to interpret [25]. For many applications it is important to understand why a model makes a certain prediction, therefore a large amount of work has been done on explaining and interpreting neural networks. Xie et al. (2019) distinguish three different classes of explainability methods for neural networks: visualization, model distillation and intrinsic methods [53]. For CNN, the first two classes are most relevant, applications of intrinsic methods are mostly used on RNN or more complex network structures [53].

Visualization-based explainability Visualization based methods express explanations by highlighting characteristics of the input that strongly influence the output of a neural network [53]. The result, typically demonstrated for images, is a map that can be placed on top of the image to highlight the relevant characteristics. Most of these methods are based on computing gradients of the score that maximizes the class probabilities for a given image [56]. Several methods exists, making use of slightly different techniques such as deconvolution or guided-backpropagation [40, 41, 55]. In contrast to using gradients, another approach is to perturb parts of the input and see how the output of the model changes [53].

In 2016, Zhou et al. proposed class activation maps (CAM), which is a method to visualize which parts of an input are discriminative for an output class and are attended by the CNN [57]. Their method assumes FCN models that use GAP followed by a softmax layer for classification. Because of this, CAM is simple and intuitive. As seen in figure 2.6, the CNN feature maps correspond to samples of the input image, the GAP layer outputs a simple average of each feature map and the predictions for a given class can be obtained by simply taking the weighted sum according to the last layer. A more formal mathematical description can be found in [57]. The presence of a GAP layer is quite a strong limitation, however models with GAP can provide competitive results [57]. Also note that the previously seen models that were most successful for TSC both used a GAP layer [15, 50].

A generalization of CAM is Grad-CAM (gradient-CAM) proposed in [39], it can be used for a broader range of models including CNN models followed by fully-connected layers. In this case, class discriminative maps are obtained by first computing the gradient of the target class score with respect to the activations of a convolutional layer. A variant that uses guided backpropagation can be used to obtain more specific and high resolution maps [41].

Model distillation Model distillation methods for explanation develop a separate explainable model that is trained to mimic the input-output behavior of a neural network [53]. The explainable model can identify important rules and/or input features and provide hypotheses for the prediction of the network.

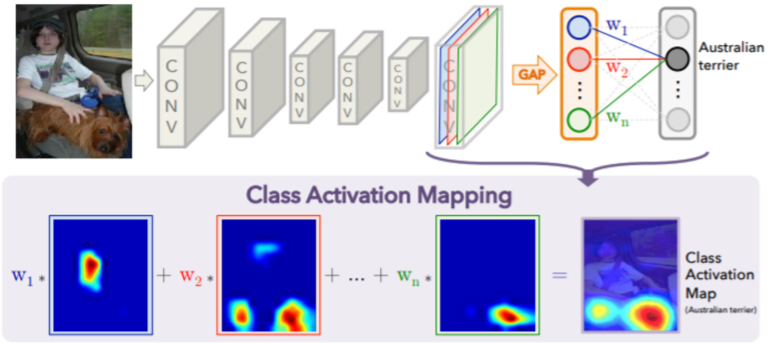


Figure 2.6: Class activation maps can be computed to visualize which parts of an input are discriminative of a specific class. By using GAP the importance for each input sample on the predicted class can be constructed. Directly taken from [57].

One of the most famous methods for model distillation is LIME (Local Interpretable Model-agnostic Explanations), which is a framework that can provide explanations for any type of black-box classification model (i.e. it is model-agnostic) [34]. LIME works by perturbing parts of the input and evaluating how the output prediction of the model changes, it uses an intrepretable model to relate local parts of the input to the output of the model. LIME can be used to provide explanations at an instance level, moreover the LIME framework is publicly available as Python code[2]. Other model distillation methods that use local approximations are mostly extensions of LIME [53]. There exists a method specifically designed for time-series, however this method only considers time-series with continuous values [17].

Another popular model-agnostic explanation method is SHAP (SHapley Additive exPlanations). In contrast to LIME, SHAP provides guarantees on accuracy and consistency [26]. In fact, the authors argue that LIME is actually a subset of SHAP. The main component of SHAP are Shapley values, which quantify feature contributions under consideration of all possible combinations of inputs. As a downside, this is an exhaustive approach which is computationally much more expensive compared to LIME. In contrast to LIME, SHAP provides explanations over the full feature space rather than at an instance level. The code for SHAP is publicly available as a Python module[3]. There is no single best explanation method, instead it depends on characteristics of the model, the data and the goals of the user [38, 11].

[2] https://lime-ml.readthedocs.io/en/latest/

[3] https://shap.readthedocs.io/en/latest/

2.2.2 Visualization

Visualization of time-series can generally speaking take two directions: either time is represented by space (e.g. a time-axis in the visualization) or by time (animation). Animation provides a solution to displaying many and large time-series, however for analysis purposes static representations of time are more effective [20]. Time-series data is often high-dimensional, continuous and large in size. These are typical challenges of time-series data, therefore simplified representations are used that can be visualized in a static manner [16]. In particular, when considering multiple time-series simultaneously these representations can be effective. This is also true for hypnograms, which can be seen as a simplified, interpretable representation of a PSG. However, when the number of hypnograms grows large, analysis and pattern observation is a challenging task that could benefit from alternative representations and visualizations.

An influential time-series representation is SAX, which represents a time-series as a sequence of symbols by taking discrete time intervals and assigning each interval to a group [24]. Since the representations of a hypnogram and SAX time-series are both categorical sequences, applications that assume SAX are also applicable to hypnograms. One such application is SAX-navigator, which enables exploration of patterns across many time-series [35]. In figure 2.7 it is shown how the tool can help to understand and compare clusters of sequences. A group of sequences can be represented as a heatmap, the difference between heatmaps can be visualized with a diverging color scheme to highlight

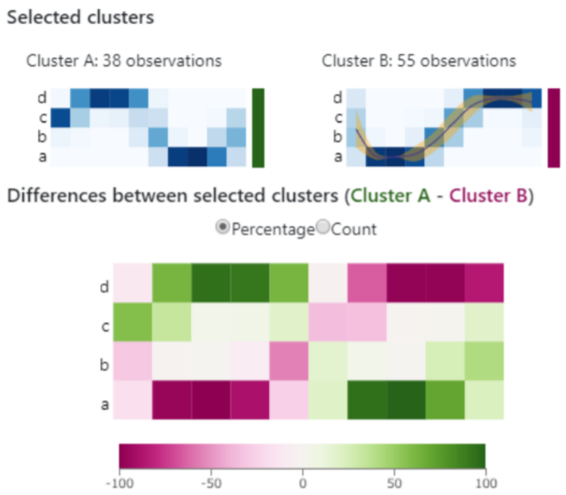

Selected clusters
Cluster A: 38 observations
Cluster B: 55 observations
d
c
b
a
Differences between selected clusters (Cluster A - Cluster B)
● Percentage ○ Count
-100
-50
0
50
100

Figure 2.7: SAX-navigator is a tool that enables interactive clustering and exploration of categorical sequences [35]. By visualizing a cluster as a heatmap it can easily be observed which areas define the cluster. Clusters can be compared by taking the difference of the heatmaps. Directly taken from [35].

differences. Combined with hierarchical clustering their tool can be used to explore SAX represented data.

Another solution involves using small multiples, in this case multiple small plots are shown in the same axes with the same scale to enable easy comparison between the plots [46]. For categorical sequences, this can be achieved by mapping values of a time-series to colors. The resulting color sequence can be visualized with a fixed height and by stacking and reordering them, patterns can be revealed. This method was demonstrated to be effective in identifying EEG patterns that are associated with sleep-disordered breathing [45].

Both SAX navigator and color sequences as small multiples are methods that preserve the time-dimension in the resulting plot. Another option would be to project multiple time-series to a representation without a time dimension and use more conventional visualization methods. High dimensional data can be projected to 2D by using linear methods such as principal component analysis (PCA) or non-linear methods such as t-distributed stochastic neighbor embedding (t-SNE) or UMAP [28]. T-SNE measures the distance between instances in high-dimensional space and maps these to a low-dimensional space such that, with high probability, similar instances are close and dissimilar instances are far in the low-dimensional space [47]. These methods can take any representation of the original data as input, for the hypnogram this can be the raw form, features (over a sliding window), transition probabilities or it can even be represented by neural network activations. Variants on these methods exist that are specifically aimed at addressing time-series [33, 52]. Nevertheless, regular dimensionality reduction methods can also be effective in gaining insights into time-series that are represented by (sliding window) features, in particular when combined with an interactive visual analytics environment [3].

2.3 Summary

- A hypnogram shows the sleep structure of a person throughout the night described by the stages wake, N1, N2, N3 and REM. An abnormal sleep structure does not imply a sleep disorder and vice versa.

- A hypnogram is obtained by (manual) visual inspection and classification of epochs of a PSG. This process is done following AASM (or R&K) scoring rules. However, these rules leave room for interpretation, therefore the hypnogram is not a perfectly accurate representation of sleep structure.

- A hypnogram can be seen as a discrete univariate time-series. An entire research area is dedicated to dealing with the temporal characteristics of time-series. Typically, time-series data is continuous, which is not true for hypnograms as they describe discrete sleep stages.

- Hypnograms can be represented by features computed over the full length or over a sliding window or by transition probabilities. These representations can serve as input for machine learning models in classification tasks.

- Deep learning models can automatically detect suitable abstract representations from time-series. In time-series classification CNN is the most suitable and effective type of deep learning model.

- Time-series datasets are typically rather small (less than 1000 instances). Nevertheless, simple models with a relatively small amount of parameters can be effective. In general, 1D CNN models require less training data than their 2D counterparts.

- Fully-convolutional networks with GAP and ResNet were shown to be effective deep learning methods for time-series. If the network is fully-convolutional and GAP is used, class activation maps can be used for explanations. Otherwise, more advanced visualization methods such as Grad-CAM can be used to provide explanations. Alternatively, model agnostic distillation methods, such as LIME and SHAP can be used to explain complex neural networks.

- Methods exist that can visualize clusters of discrete time-series. Moreover visualization methods such as t-SNE and UMAP can be used to gain insight in high-dimensional data.

Chapter 3

Methods

The goal of our thesis is to use visualization and (explainable) machine learning to identify and interpret aspects within and across hypnograms that contribute to interpretation by physicians. The methods that were used to address this goal are described in this chapter.

In section 3.1, we describe the data that was used and a data collection that was conducted to obtain subjective hypnogram evaluations by physicians. In section 3.2, we describe how t-SNE was used to visualize patterns across hypnograms to identify relations between hypnograms, collected evaluations and other attributes (e.g. diagnosis). Moreover, these visualizations were used to gain insight into disagreement between physicians. Finally, the goal of identifying features within the hypnograms that are associated with evaluation by physicians was addressed using classification models, which is described in section 3.3.

3.1 Data

Throughout this study we used hypnograms of PSG recordings that were done at Kempenhaeghe Center for Sleep Medicine as part of the SOMNIA project [48]. The hypnograms are associated with a diverse set of disorders. In addition, hypnograms of healthy subjects, that slept at Kempenhaeghe as part of the HealthBed project, were included. Subjects younger than 18 or older than 80 years were excluded. If the hypnogram of the subject contained stages that were scored with values other than one of the five AASM stages, it was excluded. The original PSG recordings were scored according to the AASM rules by experienced sleep technicians of Kempenhaeghe, which enabled us to reconstruct the hypnogram.

3.1.1 Annotations

Recall that the goal of our thesis is to identify aspects of the hypnogram that are associated with interpretation by physicians. For each hypnogram one or more associated diagnoses were available, but a specific evaluation of the hypnogram itself is missing. Therefore, we set up a data collection to obtain subjective evaluations from a small group of physicians for a large number of hypnograms. For this purpose, a web-app was created that enabled the physicians to login and rate hypnograms at their own times and pace. Four physicians participated in the data collection, the physicians are working at Kempenhaeghe, CIRO[1] and Maxima MC[2].

For each hypnogram the physician was asked to rate it as normal or abnormal (w.r.t sleep structure) and to indicate how certain they were about their assessment. The hypnograms were presented in a restricted research setting with little clinical information. This was purposely done to emphasize the structure of the hypnogram itself. Physicians were instructed to go with their first impression of the hypnogram rather than trying to clinically diagnose the subject.

During a pilot of the web-app with a physician, it was found to be difficult to objectively assess the sleep structure without considering whether a subject has a disorder because the physicians are used to diagnosing patients. Therefore, an extra option was introduced in case the sleep structure was considered abnormal; the physician was asked whether they suspected the subject to be disordered or healthy. An abnormal sleep structure in a healthy subject can for example occur because of a first-night effect. By explicitly asking whether a subject has a disorder when the hypnogram is seen as abnormal, thinking of sleep structure and disorder separately was promoted. Moreover, the extra option increases the richness of the resulting dataset and can help in discovering novel patterns. Certainty scores were collected on a 5-point Likert scale ranging from very uncertain (1) to very certain (5).

The main interface of the web-app that was used to collect these evaluations and confidence scores is shown in figure 3.1. Each hypnogram is presented in the top-center, the age of the subject was placed to the lower-left of the hypnogram. In the area below the hypnogram the evaluation could be selected by clicking the round buttons or by using the keyboard shortcuts that are provided (`[Q]`, `[W]` or `[E]` for sleep structure and `[1]` to `[5]` buttons for certainty). The result could be submitted if an evaluation and certainty were selected. A bar on top marks the progress that was made, using the previous and next buttons the user could scroll back and forth between previously evaluated hypnograms. Another page of the web-app listed the instructions. The question mark button on the evaluation page provided a shortcut to the instruction screen in case extra help was needed. Time taken per hypnogram was logged.

Each physician was assigned a personalized sequence of hypnograms with a fixed and random part. In this sequence it was ensured that each physician would rate 200 hypnograms that were evaluated by all in the same fixed order,

[1] `https://www.ciro-horn.nl/nl`

[2] `https://www.mmc.nl/`

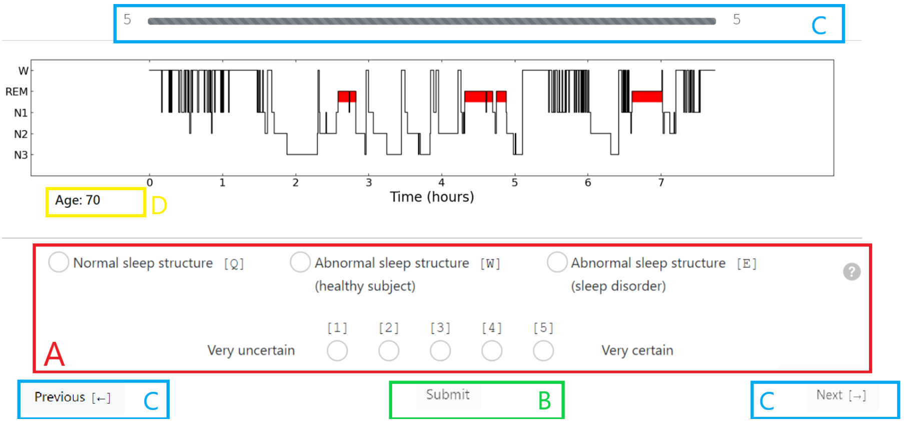


Figure 3.1: Web-app scoring interface, the hypnogram is shown in the top-center of the screen. **(A)** The evaluation and certainty options can be selected below the hypnogram, keyboard shortcuts can be used to select the buttons (e.g. `[Q]` for normal sleep structure). **(B)** When evaluation and certainty are selected the result can be submitted to continue to the next hypnogram. **(C)** Using the previous and next button the user can scroll back and forth between previously evaluated hypnograms. The current hypnogram and total evaluated are respectively shown on the left and right of the progress-bar on top. **(D)** Age is the only clinical information that is presented (other than the hypnogram).

this enables inter-rater comparison. During a pilot it was found that the first few hypnograms were more difficult as the task was novel and it required some time to get used to the diversity of the hypnograms. Therefore the first 50 hypnograms were repeated later in the sequence for each physician to circumvent this effect, the initial evaluations were discarded for the analysis. A priority sample of hypnograms was created where a larger proportion of healthy subjects was included to guarantee increased diversity in the first set of hypnograms. These hypnograms were prioritized in the order of the sequence.

A meeting was hosted to introduce the study and the scoring application. The physicians were instructed that there were no right or wrong answers and were instructed to go with their first impression of the hypnogram.

Technical Details The web-app was created using `Flask`[3] which is a `Python`[4] web-framework. Login functionality of `Flask` was used to ensure that only authorized users had access. Results were stored in a `SQlite3`[5] database, which communicated with the web-app using `SQLalchemy`[6]. The web-app was hosted on a protected server of the TU Eindhoven.

[3] `https://flask.palletsprojects.com/en/2.0.x/`
[4] `https://www.python.org/`
[5] `https://www.sqlite.org`
[6] `https://www.sqlalchemy.org/`

3.1.2 Preprocessing

Each subject was associated with one or more diagnoses (except for the healthy subjects). The number of unique diagnoses was large and many of these diagnoses were scarce, as shown in appendix A.1. Therefore, it was decided to group similar diagnoses with the mapping that is included in appendix A.2.

Apnea-hypopnea index (AHI) is a parameter that quantifies the number of apneas (i.e. breathing stops) per hour. This is an important indicator for sleep disordered breathing, one of the most common disorders in our data. AHI values were grouped as normal (less than 5), mild (5 to 15), severe (15 to 30) and extreme (more than 30).

From the obtained evaluations, disagreement was computed for each hypnogram that was annotated by multiple physicians, this was done in two ways:

1. The first measure comprises a simple binary measure indicating whether the evaluations for a given hypnogram are all the same across physicians, as shown in 3.1.

$$d(e_1, e_2, ..., e_n) = \begin{cases} 0, & \text{if } e_1 == e_2 == ... == e_n \\ 1, & \text{otherwise} \end{cases}$$
$$\text{where each } e_i \text{ is a physician's evaluation of a given hypnogram.} \tag{3.1}$$

2. The other measure also takes into account the confidence scores on the original 1 to 5 scale. For a pair of evaluations of a given hypnogram, the disagreement is the absolute difference of confidence scores if the evaluations are the same and the sum of confidence scores otherwise. For example, if two physicians evaluate the hypnogram the same with confidence 3 and 5 respectively, the disagreement is $|3-5| = 2$. If the evaluation is different for the same confidence scores, the disagreement is $3 + 5 = 8$. The disagreement for a given hypnogram is obtained by computing the mean over all pairwise evaluations as shown in equation 3.2. The resulting disagreement is in range $[0, 10]$ since each of the terms in the sum is in this range.

$$d((e_1, c_1), ..., (e_n, c_n)) = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \begin{cases} 0, & \text{if } i == j \\ |c_i - c_j|, & \text{if } e_i == e_j \\ c_i + c_j, & \text{otherwise} \end{cases} \right) / (n-1)^2$$
$$\text{where each } e_i \text{ is a physician's evaluation for a given hypnogram}$$
$$\text{and } c_i \text{ is the corresponding confidence score on a scale of 1 to 5.} \tag{3.2}$$

Additional feature engineering and preprocessing was done to prepare the hypnograms as input for visualization and machine learning methods. However, as these preprocessing steps are application specific, they will be introduced in the remainder of the methods chapter in context of the respective applications.

3.2 Visualization

Hypnograms are a simplified representation of an all-night sleep recording (i.e. a PSG), making them inherently suitable for visualization. Nevertheless, visualizing a large amount of hypnograms simultaneously is not easily done. In order to reveal patterns across all hypnograms that are associated with interpretation by physicians, t-SNE was used to reduce high-dimensional hypnograms to a two-dimensional space. All hypnograms were visualized simultaneously as points in this space and the markers were colored by evaluations and attributes (e.g. diagnosis) to reveal patterns across hypnograms. Important is that hypnograms that are similar in the high-dimensional space are grouped closely in the low-dimensional space. Naturally, this implies that t-SNE relies on a distance metric that quantifies similarity between hypnograms.

The original hypnogram is high-dimensional, but measuring similarity between original hypnograms is a non-trivial problem. Therefore, we experimented with several high-dimensional hypnogram representations and associated distance metrics as input for t-SNE. More specifically, we used the original hypnograms, features computed over the full hypnogram and features computed over a sliding window (for various window settings).

Our t-SNE methods are described in section 3.2.1 and the high-dimensional hypnogram representations and associated similarity metrics are described in section 3.2.2.

3.2.1 T-SNE

T-SNE was implemented using `scikit-learn` for `Python`[7]. The perplexity parameter of t-SNE can be thought of as the effective number of nearest neighbors [27]. Results can vary strongly for different perplexity values. Therefore, we compared perplexity values of 5, 25, 50 and 100. Using visual inspection, the most suitable perplexity value was chosen and used for the remainder of the analysis. A learning rate of 100 was used and the maximum number of iterations was set to 5000. On multiple runs of t-SNE, the results are expected to be slightly different since t-SNE tries to optimize a non-convex function. Therefore, it was decided to do 5 runs per input using fixed, but different, random states and preserve only the result with the minimal Kullback-Leiber (KL) divergence. This is a valid strategy according to Laurens van der Maaten and ensures that our results are reproducible and that a suitable random state was used for this reproducibility [27].

The results of t-SNE were visualized and the markers were colored by diagnosis, AHI grouping, evaluations and disagreement between physicians (using the measure described in equation 3.2). The markers were made partially transparent to mitigate overplotting issues in case of overlapping markers. If the visualization shows a clear separation of the data with respect to an attribute, the high-dimensional representation is more suitable for separating the groups

[7] `https://scikit-learn.org/`

for this attribute. Therefore, the visualizations were inspected for clusters and separability of attributes.

Additionally, the hypnogram pictures were used as markers in the t-SNE space to interpret associated patterns in the hypnogram. The resulting plot was quite cluttered, therefore we randomly discarded half of the hypnograms for this plot and the remaining hypnograms were cropped and down-scaled to further reduce cluttering. The same technique was used to visualize only the high-disagreement hypnograms. The resulting plot was used to reveal patterns associated with disagreement.

3.2.2 Representations

As mentioned before, we considered three different representations for hypnograms as input for t-SNE. Namely, the original representation (sequence of sleep-stages), features computed over the full hypnogram and features computed over a sliding window. Many applications, in particular unsupervised learning, rely on the notion of a distance metric which quantifies the (dis)similarity between two instances. This is also the case for t-SNE, therefore we define one or more distance metrics per representation.

3.2.2.1 Original

The most simple and intuitive representation that was used is simply the original form, i.e. as a sequence of sleep stages. This is formalized in 3.3.

$$h_{original} := \langle s_1, s_2, ..., s_t \rangle \text{ where each } s_i \in \{W, N1, N2, N3, R\} \tag{3.3}$$

For the original representation, distance between two hypnograms was measured using the Hamming distance which is simply the proportion of disagreeing sleep stages. The Hamming distance is only defined over equal length sequences, therefore the distance between two unequal length hypnograms h_a and h_b was computed only over the mutual length (i.e. if h_a is 6 hours and h_b is 8 hours, only the first 6 hours of h_b are considered). This is formally defined in equation 3.4 for two hypnograms in their original representation (i.e. as in 3.3). Obviously, this notion of distance is quite naive. For example, a hypnogram can be dissimilar to itself when shifted in time, even though we would intuitively call it similar.

$$d_{hamming}(h_a, h_b) := \frac{\sum_{i=1}^{t} s_i^a \neq s_i^b}{t} \quad \text{where } t \text{ is the min length of } h_a \text{ and } h_b. \tag{3.4}$$

3.2.2.2 Features

The second representation that we considered is a hypnogram represented by features computed over the entire hypnogram. A feature can be understood as a mapping from the original representation of a hypnogram to a number (e.g. percentage of N3 sleep). The features that we considered include parameters of the PSG report (e.g. Sleep Onset Latency), but also other computed features (e.g. number of sleep cycles). Each hypnogram can be represented by a feature vector, this is formalized in 3.5.

$$h_{feature} := (f_1(h), f_2(h), ..., f_n(h)) \\ \text{where } h \text{ is the original hypnogram and each } f_i \text{ is a feature.} \quad (3.5)$$

The full list of 34 included features is presented in table 3.1. Note that some of the features can either be computed as an absolute count or as an index (standardized by hypnogram length), these are marked in the table. In this case, we used only the index version of the feature, not the absolute counts.

Distance between two feature vectors can easily be computed using distance measures such as euclidean distance or cosine similarity. We chose to use euclidean distance because the difference in magnitude of features was considered relevant. Features were standardized to have zero mean and unit variance before applying euclidean distance, this ensures that features measured on larger scales are not dominant. In contrast to the original representation, features are a simplified representation of the original data that no longer contains a time-dimension. Therefore, the features capture global characteristics of the hypnogram.

3.2.2.3 Sliding Window

The third and final representation was obtained by computing features over a sliding window. A sliding window was used to extract length w subsequences of the hypnogram, each consecutive window is shifted with step s. For each subsequence, a feature vector was computed, this results in a representation of the hypnogram that is a matrix as shown in equation 3.6.

$$h_{window} := \begin{bmatrix} f_1(s_1, ..., s_w) & f_1(s_{1+s}, ..., s_{w+s}) & ... & f_1(s_{1+ns}, ..., s_t) \\ f_2(s_1, ..., s_w) & f_2(s_{1+s}, ..., s_{w+s}) & ... & f_2(s_{1+ns}, ..., s_t) \\ \vdots & \vdots & \ddots & \vdots \\ f_n(s_1, ..., s_w) & f_n(s_{1+s}, ..., s_{w+s}) & ... & f_n(s_{1+ns}, ..., s_t) \end{bmatrix}$$

$$\text{where } w \text{ is the size of the window, } s \text{ is the step-size, } s_1, ...s_t \text{ are the stages} \\ \text{of the original hypnogram, each } f_i \text{ is a feature and } n \text{ is the smallest integer} \\ \text{such that } 1 + ns + w \leq t \text{ holds.} \quad (3.6)$$

Name	**Abbreviation**	**Description**
Time in Bed	TIB	Total time of hypnogram
Total Sleep Time	TST	Total time of hypnogram spent asleep (i.e. not in W)
Sleep Efficiency	SE	Percentage of hypnograms spent asleep (TST/T *100)
Sleep Onset Latency	SOL	Number of minutes from the start till first epoch that is not W
Sleep Period Time	SPT	Number of minutes from sleep onset till final awakening
N3 Onset Latency	N3OL	Number of minutes from the start till first N3 epoch
REM Onset Latency	REMOL	Number of minutes from the start till first REM epoch
Wake After Sleep Onset	WASO	Number of minutes spent in W after sleep onset
Snooze Time	ST	Number of minutes W at the end of the hypnogram
%W[*]	pW	Percentage of hypnogram spent in Wake
%N1[*]	pN1	Percentage of hypnogram spent in N1
%N2[*]	pN2	Percentage of hypnogram spent in N2
%N3[*]	pN3	Percentage of hypnogram spent in N3
%REM[*]	pREM	Percentage of hypnograms spent in REM
Sleep Stage Transitions[†]	SST	Total number of transitions from a stage to another
N3 Awakenings[†]	N3WKN	Number of transitions from N3 to W
REM Awakenings[†]	REMWKN	Number of transitions from REM to W
Long Awakenings[†]	WKNL	Number of periods of ≥ 5 minutes consecutive W after sleep onset and before final awakening
Awakenings[†]	WKN	Number of transitions to W
N1 Sleep Stage Transitions[†]	N1SST	Number of transitions to N1
N2 Sleep Stage Transitions[†]	N2SST	Number of transitions to N2
N3 Sleep Stage Transitions[†]	N3SST	Number of transitions to N3
REM Sleep Stage Transitions[†]	REMSST	Number of transitions to REM
Max Duration W	maxW	Maximum amount of minutes consecutively spent in W
Max Duration N1	maxN1	Maximum amount of minutes consecutively spent in N1
Max Duration N2	maxN2	Maximum amount of minutes consecutively spent in N2
Max Duration N3	maxN3	Maximum amount of minutes consecutively spent in N3
Max Duration REM	maxREM	Maximum amount of minutes consecutively spent in REM
Min Duration W	minW	Minimum amount of minutes consecutively spent in W
Min Duration N1	minN1	Minimum amount of minutes consecutively spent in N1
Min Duration N2	minN2	Minimum amount of minutes consecutively spent in N2
Min Duration N3	minN3	Minimum amount of minutes consecutively spent in N3
Min Duration REM	minREM	Minimum amount of minutes consecutively spent in REM
Sleep Cycles[†]	CCL	Number of sleep cycles[‡]

Table 3.1: Hypnogram features

The window features were computed for the window sizes and step sizes as shown in table 3.2. The step size s was always chosen to be one-third of the window-size, which implies that each sample (except at the ends) is contained in exactly three windows. Recall that sleep epochs have a length of 30 seconds, thus a window of 30 minutes contains 60 samples. For each of the five stages, the probability of that stage in the window is computed. Moreover, pairwise transition probabilities are included that are the same as the parameters of a simple Markov chain (e.g. $\mathbb{P}(s_i + 1 = W|s_i = N2)$). This feature is computed for all 25 pairwise stage combinations (including combinations of a stage with itself), which brings the total number of features per window to 30.

In contrast to the feature representation, each sliding window representation still contains a time dimension. Computing distance between two hypnograms in window representation is therefore less straight-forward than in feature rep-

[*] Can also be computed as an absolute feature (e.g. minutes of W)

[†] Can also be computed as an index feature (e.g. sleep stage transitions per hour)

[‡] We defined a sleep cycle as a period $\geq t_r$ minutes with at least p_r% REM in this period, followed by a period $\geq t_n$ minutes with a minimum of p_n% NREM. The parameters were chosen to be: $t_r = 10$ minutes, $t_n = 30$ minutes and $p_r = p_n = 55\%$.

Name	**w15**	**w30**	**w60**	**w90**	**w120**
w	15m	30m	60m	90m	120m
s	5m	10m	20m	30m	40m
# samples	30	60	120	180	240

Table 3.2: Window-size w, step size s and number of samples per window of the five window representations that were computed.

resentation. The sliding window hypnograms have unequal lengths, we dealt with this in the same way as for the original representation. If two hypnograms h_a and h_b have unequal length, distance was computed only over the mutual part. For example, if h_a is described by 20 windows and h_b by 22, only the first 20 windows of h_b were used for computing the distance.

Distances between hypnograms in window representation were computed using two different distance metrics: window distance, which was inspired by euclidean distance, and dynamic time warping distance.

1. **Window distance**. Euclidean distance would be an inappropriate metric since the dimensionality varies (hypnogram pairs can have different mutual length). Thus, hypnogram pairs with a large mutual length would be associated with larger distances as the number of windows is larger. Instead, it was decided to use the average squared distance between the window features as a distance metric that controls for different length hypnogram pairs, this is formalized in 3.7.

$$d_{window}(h_a, h_b) := (\sum_{i=1}^{t}\sum_{j=1}^{n}(h^a_{i,j} - h^b_{i,j})^2)/t$$

where h_a and h_b are hypnograms in window representation (as in 3.6)
and t is the minimum length of h_a and h_b. (3.7)

2. **Dynamic Time Warping distance**. Dynamic Time Warping (DTW) can be used to compute distances between (multivariate) time-series that are not perfectly synchronous. In other words, DTW does not necessarily compare the i-th sample of a time-series with the i-th sample of another time-series. Instead, DTW can be used to align samples of a time-series with samples of another time-series that are similar and temporally close, thereby 'warping' the time-series to compute an optimal alignment. The similarity between two time-series is computed after aligning the time-series. Thus, DTW can partially overcome the limitation that we mentioned in the original representation: that a hypnogram can be dissimilar to itself when shifted in time, even though we would intuitively call it similar.

The sliding window representation can be seen as a multivariate time-series where each feature is a channel and each window is treated as a point in time. DTW similarity was computed using the `dtw` method of the `tslearn` package for `Python`[1]. For hypnograms that varied extremely in length, the `dtw` function occasionally returned infinite or empty values. These values were replaced by the maximum DTW distance over all hypnogram pairs. Moreover, the `dtw` function is asymmetric (i.e. $h(a,b) \neq h(b,a)$), we dealt with this by defining the distance as the mean of the asymmetric distances for each pair.

The window features are all in range $[0,1]$. Therefore, standardization was not strictly required before computing distances. Nevertheless, we were interested in the effect of standardization of window features and how the results of t-SNE would be different from the non-standardized window features. Therefore, both non-standardized and standardized window features were included. Standardization was applied per feature over all windows and hypnograms.

3.3 Classification

One of the goals that we defined in section 1.1 concerns identifying which features of the hypnogram drive interpretation by physicians. For this purpose four classifiers were trained that take as input hypnograms and predict for each hypnogram the evaluation that was collected as described in section 3.1.1, thereby essentially 'mimicking' a physician. The normal sleep structure evaluation was barely used, therefore it was decided to use a binary grouping of evaluations; normal sleep structure and abnormal (healthy) were grouped as *healthy* and abnormal (disordered) was called *disordered*. These labels were used as target variable for all classifiers, thereby making it a binary classification problem. The evaluations of the physician who scored the most hypnograms were used.

In total we used four different models: logistic regression, decision tree and two CNN models. The logistic model and decision tree take hypnogram features as input and were chosen for their inherent explainability. The CNN models are able to derive features automatically and therefore take the raw hypnograms as input making it a hypothesis-free approach (i.e. we do not make explicit prior assumptions on which features are relevant). The methods for training and evaluating these models are described in the remainder of this section.

3.3.1 Decision Tree

A decision tree was trained that takes as input the features, as previously described in table 3.1. The model predicts healthy/disordered labels. The decision tree was implemented using `scikit-learn`. Optimal splits were found using

[1] `https://tslearn.readthedocs.io/en/stable/user_guide/dtw.html#dtw`

Gini-impurity and balanced class weights ensured that these splits were not biased towards the over-represented disordered group. Limitations to the maximum depth and number of samples per leaf/split were set to prevent learning a perfectly accurate, overfitting model. The best value for these parameters was chosen using a grid search in combination with 10-fold stratified cross validation, by using the stratified option it was ensured that each fold had a balanced class diversity.

Evaluation Evaluation of the tree was done by inspecting the resulting model, accuracy was computed over the whole dataset and using 10-fold stratified cross validation. Moreover, a confusion matrix was created to inspect the performance of the model across the classes. In addition, receiver operating characteristic (ROC) curve was used to evaluate the performance of the tree at all classification thresholds and area under the curve (AUC) was used for comparison with the other classification models. ROC curve depicts the true positive rate (TPR) and false positive rate (FPR) at different decision thresholds. In our case, healthy was treated as the positive class. This implies that TPR is the proportion of actual healthy that is correctly predicted and FPR is the proportion of actual disordered that is incorrectly predicted.

Due to the subjective nature of the labels we do not expect the classes to be perfectly separable. Therefore, the accuracy of the model was compared across confidence levels assigned by the physician. Here we assume that errors on low-confidence evaluations are less severe than on high-confidence evaluations. Similarly, for the hypnograms that were evaluated by multiple physicians, we inspected the model performance with respect to agreement between physicians. Here we used the simple binary agreement metric that was described in 3.1.2.

3.3.2 Logistic Regression

A logistic regression model was implemented using `scikit-learn`. The input features were standardized (zero mean, unit variance) before training to ease interpretation of model coefficients. Again, balanced class weights were used to deal with the imbalance between the groups. The model was trained using L2 regularization which helps in preventing overfitting and dealing with multicollinearity. The predicted probabilities were binarized at $p = 0.5$ to obtain class predictions. Some of the features were excluded for having extremely high correlations (e.g. sleep efficiency and W% which are perfectly negatively correlated). The model was evaluated using the same evaluation methods as for the decision tree. In addition, the model coefficients were inspected.

3.3.3 Baseline CNN

The logistic and decision tree model take features as an input, which are a simplified representation of a hypnogram and describe mainly global characteristics. It might be that there are other aspects of a hypnogram, not captured in the feature representation, that can explain interpretation by physicians. The advantage of CNN over these traditional machine learning models is that it can automatically learn such discriminative features of the input.

Since we are unaware of any existing applications of CNN on hypnograms, we started by fitting a (relatively) simple model to demonstrate feasibility of our approach. For this we used a FCN model with GAP. This was motivated by their proven effectiveness for time-series and their inherent explainability through CAM [50, 57]. It should be noted that, by definition of GAP, such a model can only detect global features of the hypnogram since any detected feature is averaged over the full time dimension before class prediction. Therefore, this model is not able to assign different importance to features based on their temporal location.

Input The FCN model takes as input the original hypnogram in a one-hot encoding to enable the model to correctly interpret the discrete stages. In a one-hot encoding, each hypnogram is represented as a binary matrix of 5 by t (number of epochs), where each row corresponds to a sleep stage. Each column of the matrix has the value one exactly once, the remaining values are all zero (each timestamp describes a single sleep stage). This is shown in 3.8.

$$h_{one-hot} := \begin{bmatrix} s_1^W & s_2^W & ... & s_t^W \\ s_1^{N1} & s_2^{N1} & ... & s_t^{N1} \\ s_1^{N2} & s_2^{N2} & ... & s_t^{N2} \\ s_1^{N3} & s_2^{N3} & ... & s_t^{N3} \\ s_1^R & s_2^R & ... & s_t^R \end{bmatrix}$$

$$\text{where each } s_i^v \in \{0,1\} \text{ and each column } i \text{ has exactly one } s_i^v \text{ that is 1.} \tag{3.8}$$

Architecture The FCN model consists of a small number of convolutional layers, followed by a GAP layer that takes the average value for each channel across the temporal dimension. A suitable number of layers and nodes was selected by experimenting with several combinations. The output of the model was obtained using a fully-connected final layer with sigmoid activation function to map the outcome as probabilities. Outcome probabilities larger than 0.5 are predicted as healthy, otherwise the model predicts disordered. The model was trained using balanced class weights and binary-cross entropy loss. Implementation was done using `Keras`[2].

Evaluation For training and evaluation, a fixed train/test split of 70/30 was used over cross-validation since training the model for multiple folds was considered too expensive. The FCN model was evaluated in a similar manner as the previous models using accuracy, confusion matrix, ROC curve and accuracy by

[2] `https://keras.io/api/`

confidence and disagreement. In addition, CAM was implemented as described in [57]. The resulting heatmap was plotted as background for a group of randomly sampled hypnograms from both classes to evaluate which features of the hypnogram are used by the model.

3.3.4 Advanced CNN

Recall that one of our goals is to identify (previously) unknown features of the hypnogram that are associated with interpretation. The decision tree and logistic model have limited ability in uncovering such features since they take only global features as input. Similarly, the FCN model cannot, by design, assign different weight to features based on their the temporal location. Therefore, we experimented with more advanced architectures to identify novel discriminatory features that lead to an overall increase in classification performance.

For this purpose, we adapted the FCN model by adding pooling layers after convolutions, replacing the GAP layer with one or more fully-connected layers, and adding batch normalization and/or dropout layers to prevent overfitting. Moreover, we experimented with splitting the input hypnogram into equal parts and training a different model for each part (similar to a sliding window approach), the results of the individual parts were recombined into an overall prediction.

The input, labels and evaluation methods for the FCN model were also used for the advanced model. By inspecting the results of the evaluation methods, it was assessed whether the increased complexity of the model yielded a significant increase in performance, which might indicate that the model detects and uses a novel discriminatory feature of the hypnogram.

Chapter 4

Results

In this chapter we describe the results that were obtained using the previously described methods, the current chapter follows the same structure as the methods chapter. In section 4.1, the results of the data collection are described, followed by the visualization and classification results in section 4.2 and 4.3 respectively.

4.1 Data

A total of 1067 SOMNIA and 100 HealthBed subjects was included. The age distribution for these subjects is shown in figure 4.1. For most of the subjects a diagnosis was available, 577 subjects were diagnosed with exactly one disorder, 107 subjects were healthy or not diagnosed and the remainder had two to four diagnoses. The distribution of diagnoses can be seen in figure 4.2. The three most frequent diagnoses are sleep disordered breathing which occured 653 times (mostly obstructive sleep apnea), insomnia which occured 378 times and movement disorder which occured 172 times.

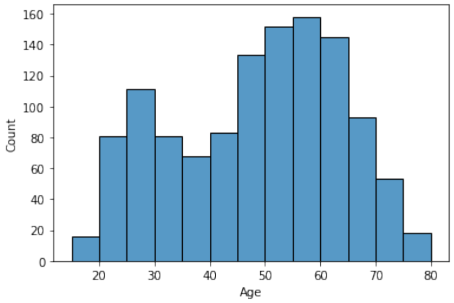


Figure 4.1: Subjects older than 18 and younger than 80 were included in the research.

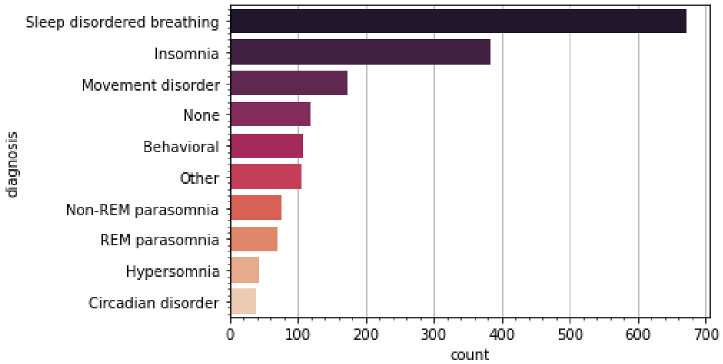


Figure 4.2: All diagnoses ordered by frequency. The majority of subjects has been diagnosed with sleep-disordered breathing.

The distribution of hypnogram length (in hours) is shown in figure 4.3. The majority of hypnograms are between 8 and 9 hours, another large group is roughly between 6.5 hours and 10 hours. There are a handful of outliers were the hypnogram is less than 3.5 hours or more than 11 hours.

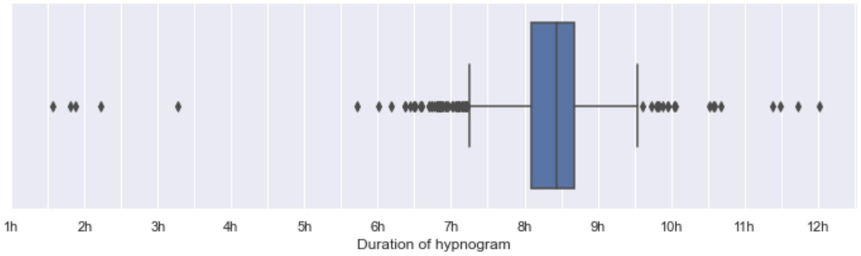


Figure 4.3: Boxplot showing durations of hypnograms. The majority of hypnograms are between 8 and 9 hours. A handful of hypnograms is shorter than 3.5 hours or longer than 11 hours.

4.1.1 Annotations

We received responses of two physicians who scored respectively 612 and 405 hypnograms. For 242 hypnograms, two annotations were collected, 533 hypnograms were annotated by one physician and 445 hypnograms were not annotated at all. The option normal sleep structure was barely used, only 9 (1%) and 17 (4%), by the two physicians respectively. Therefore, it was decided to group the evaluations as *healthy* (normal and abnormal-healthy) and *disordered* (abnormal-disordered). Similarly, the lowest-confidence score was only used 13 times in total, therefore confidence was grouped as low (1-2), medium (3) and high (4-5).

Most of the hypnograms were evaluated as disordered, this was true for 82% and 72% for physician 2 and 4 respectively. Agreement between the two physicians was 80%. The frequency for healthy and disordered across confidence levels and per physician are shown in figure 4.4. From this figure it can be observed that the disordered evaluations were assigned high confidence by both physicians (i.e. the left two purple bars are high). In contrast, both physicians were in general less confident on the healthy evaluations (purple bars on the right are small and orange bars are relatively large).

For the hypnograms that were associated with exactly one disorder, we computed the evaluations of each physician for the five most frequent diagnoses. The results are shown in figure 4.5, the orange and blue bars represent physician 2 and 4 respectively. The hypnograms that were not associated with any diagnosis (i.e. healthy subjects) were evaluated disordered in 63% and 45% of the cases for physician 2 and 4 respectively. In the 'other' group, it can be observed that the evaluations are approximately equally distributed. For the other diagnoses, the majority of hypnograms was evaluated as disordered, differences between physicians are small.

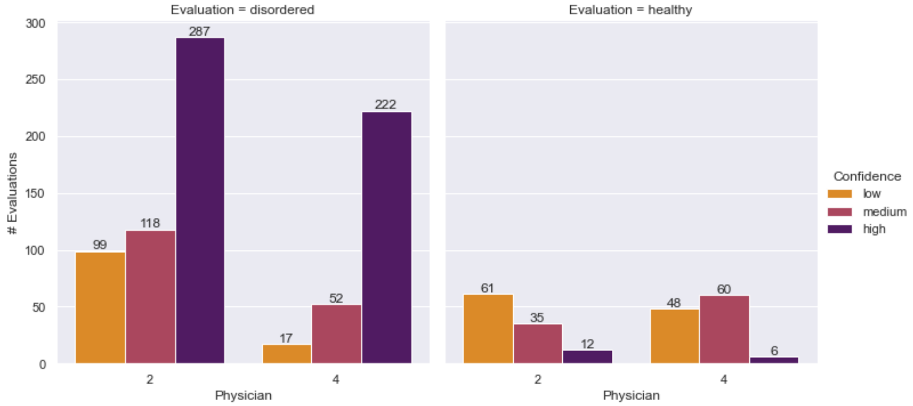


Figure 4.4: Overall the majority of hypnograms was evaluated as disordered, confidence was in general high for those evaluations. In contrast, both physicians were less confident on the, more rare, healthy evaluations.

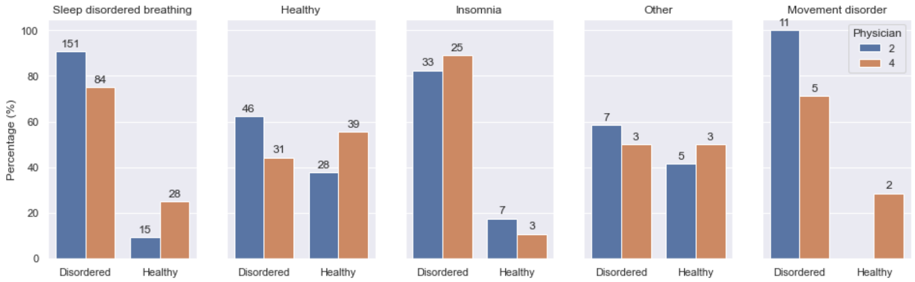


Figure 4.5: Distribution of healthy/disordered evaluations per physician for the five most frequent diagnoses.

4.2 Visualization

As described in 3.2, we used t-SNE to visualize patterns across all hypnograms. The high-dimensional hypnogram representations and associated distance metrics on which we applied t-SNE are summarized in table 4.1. We compared a total of 22 feature sets (each window feature set was used in 2x2 combination with DTW/window distance and standardization). Perplexity values of 5, 25, 50 and 100 were used, perplexity of 5 was clearly too low, observed from small artifacts in the plots. Differences between 25, 50 and 100 were small, therefore we only consider perplexity of 50 for the remainder of the results. Furthermore, it was observed that standardization of window features did not improve separability of the data, therefore we excluded standardized window feature sets.

Name	**Representation**	**w**	**s**	**Distance metric**	**Standardization**	**Dimensions**
original	original/one-hot			Hamming	No	1442
features	features			Euclidean	Yes	34
w15	window	15m	5m	window/DTW	Yes/No	4290
w30	window	30m	10m	window/DTW	Yes/No	2130
w60	window	60m	20m	window/DTW	Yes/No	1050
w90	window	90m	30m	window/DTW	Yes/No	690
w120	window	120m	40m	window/DTW	Yes/No	510

Table 4.1: Summary of the feature sets that were used and the resulting number of dimensions per hypnogram.

In figure 4.6, the resulting visualizations for the feature representation are shown. Each marker represents a hypnogram, markers are colored by diagnosis, AHI, evaluations and disagreement (as described in 3.1.2) in the four plots in this figure. It can be observed that the points align like a ball, there are no disconnected clusters of hypnograms. In the plot for diagnoses (top-left), it is seen that most of the healthy subjects (the green markers) are quite strongly clustered together in the upper-left. Similarly, most of the non-REM parasomnia cases, marked in red, are in the top-center of the plot. For the other diagnoses, the markers are quite interspersed across the space. The plot for

AHI (top-right) shows that the hypnograms with normal AHI (i.e. less than 5) are grouped in the upper-left, in contrast the extreme cases (more than 30) are mostly at the bottom and right edges of the space. Similarly, the plot for evaluations (bottom-left) shows that most of the healthy evaluations are grouped in the upper-left, even though cases exist that are more at the center or right of the space. The hypnograms at the bottom and right edges were always evaluated as disordered, most often with high confidence. The final plot (bottom-right), shows the disagreement between physicians (i.e. difference between confidences on a continuous scale), it can be observed that the cases with medium disagreement (5 ± 1, white/light markers) are mostly towards the centre of the space. A disagreement of 5 indicates that the two physicians assigned different labels, but at least one of the two evaluated the hypnogram with low confidence. The high disagreement cases, indicated by (dark)orange or red markers are more at the edges of the space.

The same visualizations for the original representation (Hamming distance) and window feature representations (DTW and window distance) are shown in appendices B.2 and B.3 respectively. Overall, the t-SNE projections for these representations show similar patterns as the feature projection, but the markers are more interspersed with respect to diagnosis, AHI and evaluation. In particular, the original representation, seen in figure B.2, shows a more mixed pattern. For the window representations, there is no clear difference between DTW and window distance. In general, larger window sizes show less interspersed patterns than smaller window sizes.

There are in total 25 hypnograms for which the disagreement was larger or equal to 6, which implies that the two physicians assigned a different evaluation and both with at least medium confidence or one with high confidence. This corresponds to the orange and red markers in the bottom-right of figure 4.6). We took the subset of these high-disagreement hypnograms and plotted them in t-SNE projection of hypnogram features (same space as figure 4.6) while using the original hypnograms as markers, the result is shown in figure 4.7. There are some small clusters of hypnograms with high disagreement, in the top-left we see some hypnograms that show little fragmentation, awakenings occur mostly at the begin, one of the hypnograms contains some REM right at the start. The hypnograms on the top-right show many back and forth transitions to N2 while in REM. All of the hypnograms contain several periods of N3 and REM (although in some cases only short) and show, to some extent, a cyclical pattern.

The same method of plotting hypnograms in the t-SNE feature projection space was applied on all hypnograms. The resulting plot is quite cluttered and requires careful observation, nevertheless it provides some more context to the previous results, therefore this plot and the description of the results are included in appendix B.4.

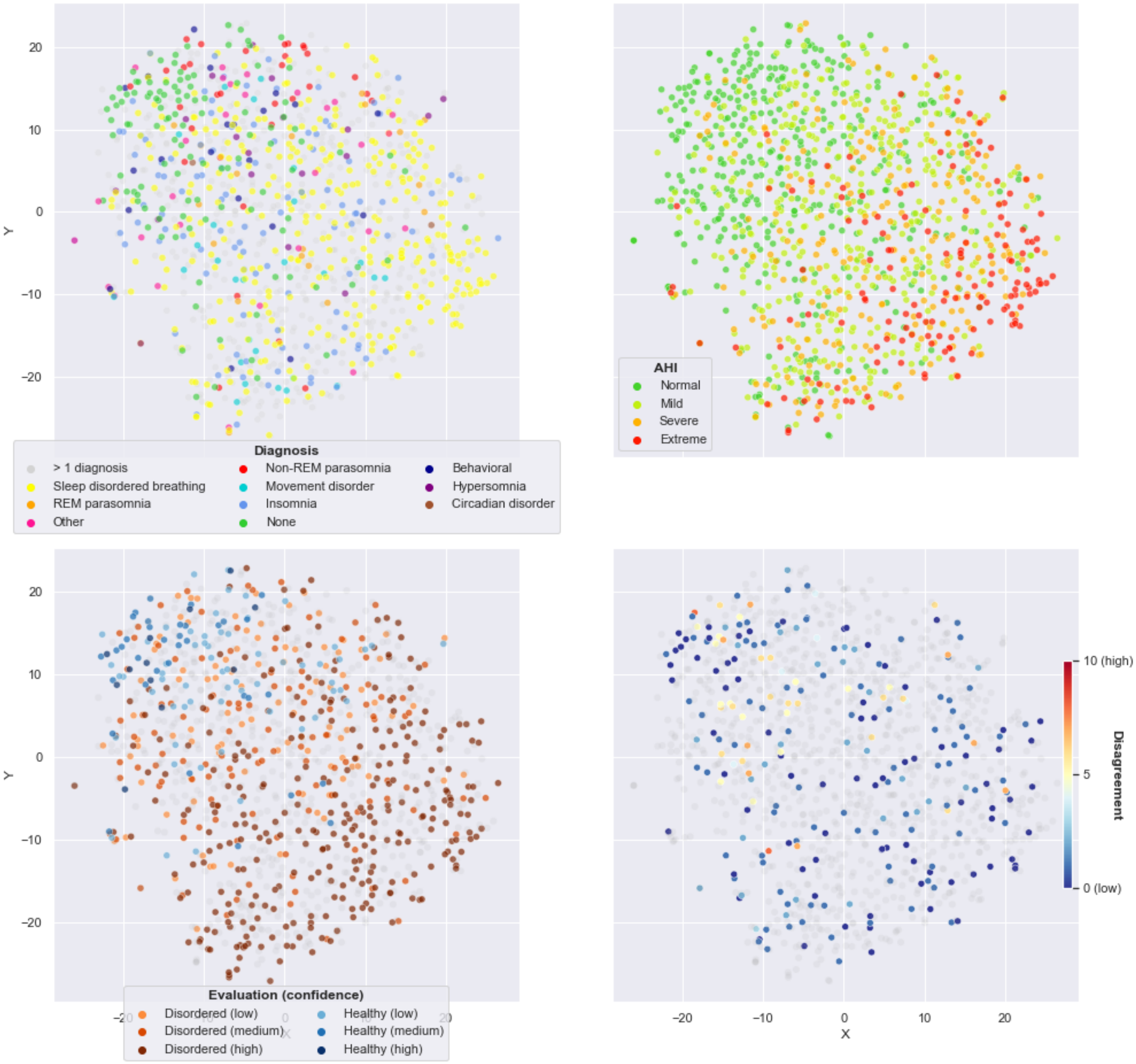


Figure 4.6: Hypnograms as features projected to two-dimensions using T-SNE. Each point represents a hypnogram, hypnograms with similar features are close to each other. The markers are colored by diagnosis **(top-left)**, AHI **(top-right)**, evaluation and confidence **(bottom-left)** and disagreement **(bottom-right)**.

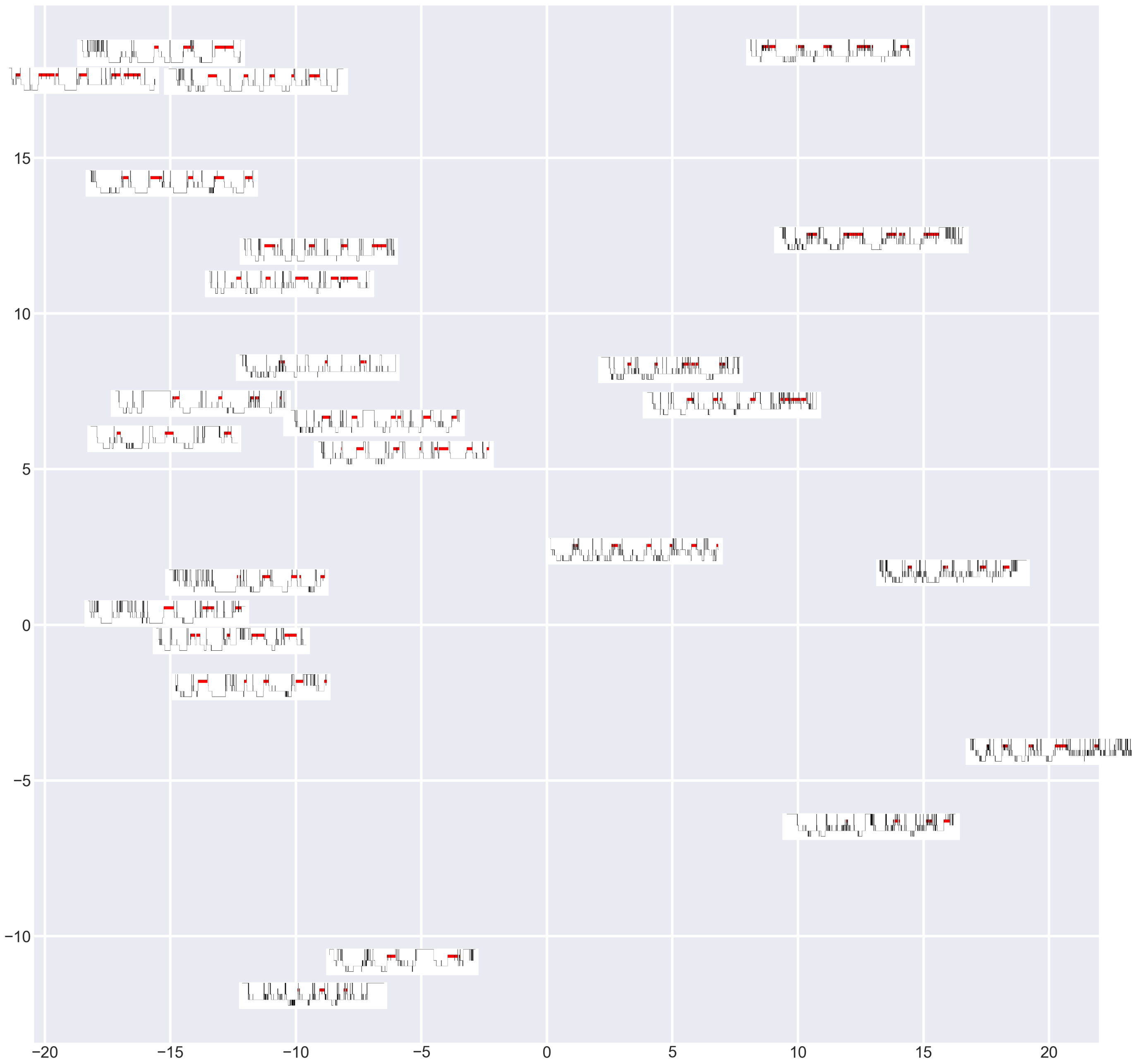


Figure 4.7: Hypnograms where disagreement was equal or higher than six, shown in the t-SNE projection of hypnogram features. These are the same hypnograms as the orange and red markers in the bottom-right of figure 4.6

4.3 Classification

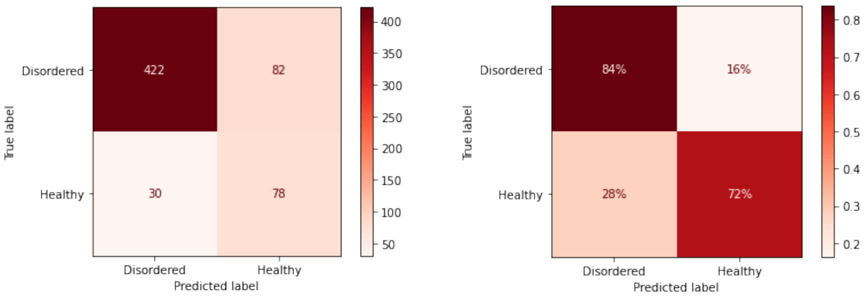
In order to identify which features of the hypnogram drive interpretation by physicians we trained four classification models predicting healthy/disordered, as described in section 3.3. We used decision tree and logistic regression models taking hypnogram features as input for their inherent explainability. As the ability of these models is always limited by the choice of features, CNN models were used for their potential to detect novel features that are associated with interpretation. A baseline FCN with GAP was created to demonstrate the effectiveness of CNN on hypnograms and a more advanced CNN model was used that can assign importance to features based on their temporal location.

4.3.1 Decision Tree

Using 10-fold stratified cross validation and grid search the best tree was found to have depth two. The resulting tree is shown in figure 4.9. Note that the two right-most leaves of the tree are both labeled disordered, therefore the complete model translates to the following simple rule:

> *A hypnogram is 'healthy' if it has less than 25 awakenings and on average less than 0.36 long awakenings (five minutes or longer) per hour, otherwise the hypnogram is 'disordered'.*

The decision tree is 82% accurate on all evaluations, the average accuracy of a tree with depth two over the 10 folds was 77%. The absolute and normalized confusion matrix are shown in figure 4.8. Even though balanced class weights were used during training, the model performs better for disordered hypnograms which it detects with 84% accuracy in comparison to the 72% accuracy on the healthy hypnograms. In absolute terms, a large amount of the predicted healthy hypnograms actually belongs to the disordered class, thus the precision for healthy predictions is quite low.



(a) Absolute counts.

(b) Normalized over true label.

Figure 4.8: Confusion matrix for the decision tree.

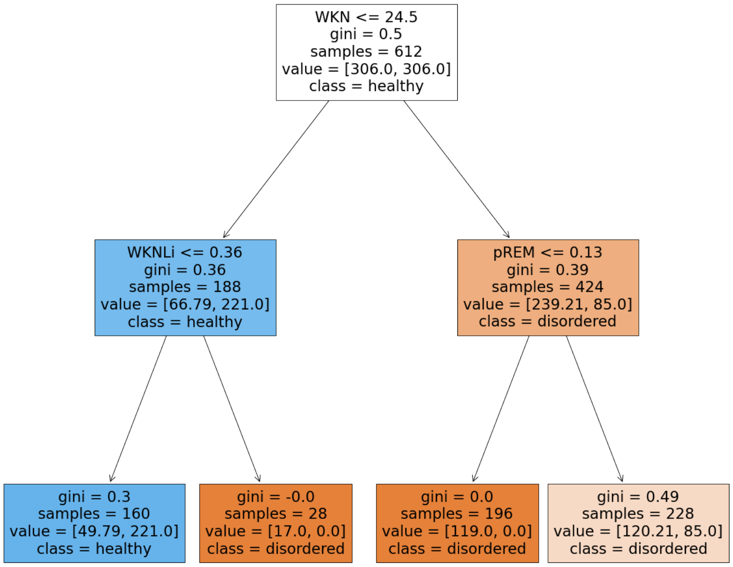


Figure 4.9: Decision tree model that predicts a hypnogram as ’healthy’ if it has less than 25 awakenings and on average less than 0.36 long awakenings (five minutes or longer) per hour, otherwise the hypnogram is ’disordered’.

The performance of the model across evaluations by confidence level is shown in figure 4.10. The model performs more accurate on the high-confidence hypnograms, on which it is 92% accurate. In contrast, the medium-confidence hypnograms were detected 74-75% accurate and the low-confidence hypnograms were detected with 71% and 67% for disordered and healthy respectively. Similarly, for the subset of 242 hypnograms that were assessed by both physicians, the model performs better on the hypnograms where the physicians agree. In case of agreement between physicians the decision tree is 89% accurate. In contrast, the accuracy was only 59% when the physicians did not agree.

Finally, each leaf of the tree is associated with a probability (leaves are not pure). These predicted probabilities were used to create the ROC curve shown in figure 4.11. The ROC curve shows the TPR and FPR at various binary decision thresholds, there are only two points on the curve since the decision tree has two non-pure leaves. The AUC of the curve is 0.84.

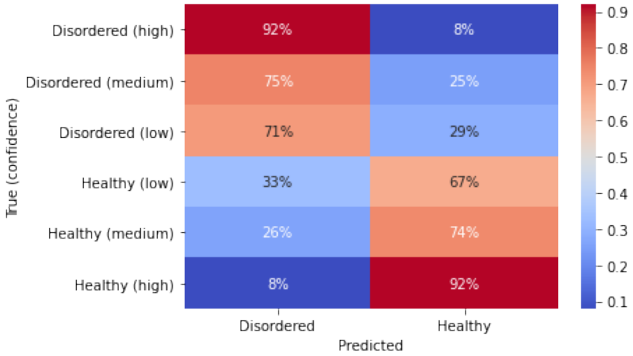


Figure 4.10: Performance of decision tree with respect to physician confidence.

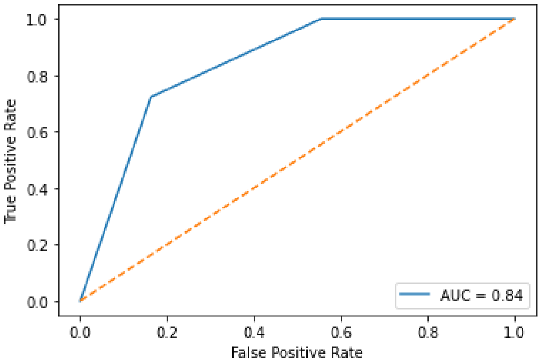


Figure 4.11: ROC curve displaying the trade-off between TPR and FPR across decision thresholds for the decision tree.

4.3.2 Logistic Regression

The logistic regression model was trained with the same data as the decision tree model, there were no hyperparameters that we needed to tune. The obtained coefficients of the model are shown in table 4.2. Recall that the features were standardized, therefore the coefficients can be compared. In the table, the coefficients are presented in sorted order, positive model outcome means that a hypnogram has a higher probability for healthy. The largest positively contributing features are percentage REM and N2, the largest negative contributors are percentage W, REM onset latency (minutes till first REM) and REMSSTi (number of REM transitions per hour).

Accuracy for the logistic model is 82% over all evaluations, the average accuracy over 10 stratified folds was 77%, which is the same as the decision tree. The absolute and normalized confusion matrix are shown in figure 4.12. In contrast to the decision tree, the logistic model performs better on the healthy

Feature	**pREM**	**pN2**	**CCL**	**pN3**	**maxN3**	**maxN2**	**REMWKNi**	**ST**
Coefficient	0.76	0.56	0.38	0.34	0.29	0.28	0.26	0.17

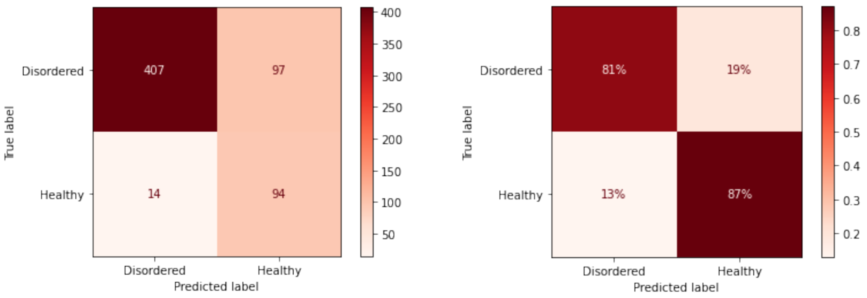
Feature	**SOL**	**maxN1**	**maxW**	**WASO**	**WKNL**	**TIB**	**N3SSTi**	**N3OL**
Coefficient	0.08	0.07	0.06	-0.05	-0.14	-0.15	-0.19	-0.22

Feature	**N3WKN**	**pN1**	**maxREM**	**SSTi**	**WKNi**	**REMSSTi**	**REMOL**	**pW**
Coefficient	-0.26	-0.32	-0.47	-0.64	-0.67	-0.73	-0.78	-0.79

Table 4.2: Feature coefficients ordered from greatest positive (top-left) to greatest negative (bottom-right).

hypnograms with 87% accuracy and 81% on the disordered. The model performance by physician confidence is shown in figure 4.13. Again the model is more accurate on the high confidence hypnograms where it is 94 and 100% accurate for disordered and healthy respectively, in case of low confidence this was 54% and 82%. The logistic model was 94-100% accurate on the high confidence hypnograms. and 90% accurate on the hypnograms with positive agreement.

The logistic regression model predicts a probability of being healthy for each hypnogram, by default a binary decision threshold of 0.5 is used. The ROC curve, shown in figure 4.14, shows the trade-off between TPR and FPR at all distinct binary decision thresholds, the curve has an AUC of 0.92.



(a) Absolute counts.

(b) Normalized over true label.

Figure 4.12: Confusion matrix for the logistic model.

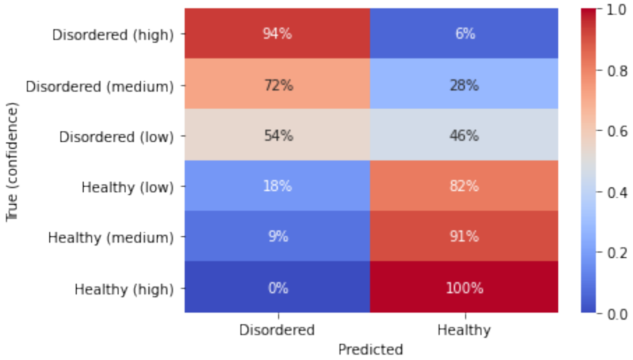


Figure 4.13: Performance of logistic model with respect to physician confidence.

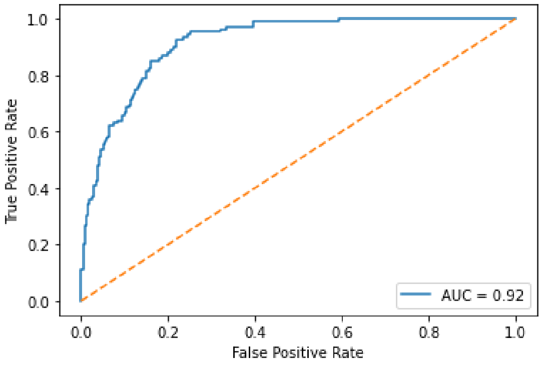


Figure 4.14: ROC curve displaying the trade-off between TPR and FPR across decision thresholds for the logistic model.

4.3.3 Baseline CNN

The architecture of the FCN model with GAP that was used as baseline CNN model is shown in table 4.3. Three convolutional layers were used as this gave a slight performance improvement over two layers, adding more layers or nodes to existing layers did not yield performance improvements. The overall accuracy of the model on the training data was 83 %, on the test data it was 80% accurate. As seen in figure 4.15, the loss decreased quickly during the first 20 epochs of training, in the remaining 80 the loss gradually decreased. Meanwhile, accuracy was mostly decreasing in the beginning and going up and down in the end.

The confusion matrix that describes the performance of the FCN model on the test data is shown in figure 4.16. The model was 74% accurate on the healthy hypnograms and 81% on the disordered. Similar to the previous models, approximately half of the predicted healthy hypnograms was actually evaluated as disordered, indicating that the model has low precision. The performance of

Layer	**# Nodes**	**Activation**	**Size**	**Stride**	**Output Shape**	**# Param**
Convolution 1D	15	ReLU	3	1	(1407, 15)	240
Convolution 1D	30	ReLU	3	1	(1407, 30)	1380
Convolution 1D	15	ReLU	3	1	(1407, 15)	1365
Global Average Pooling					15	0
Dense	1	Sigmoid			1	16
Total Param						3001

Table 4.3: Architecture of FCN with GAP

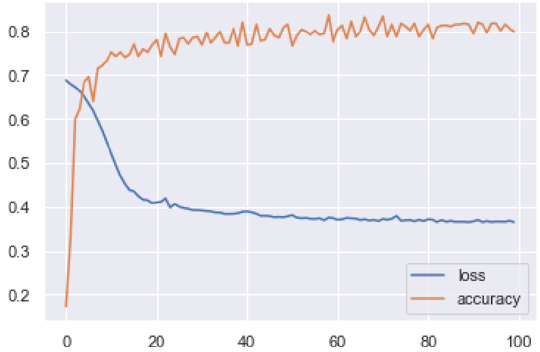
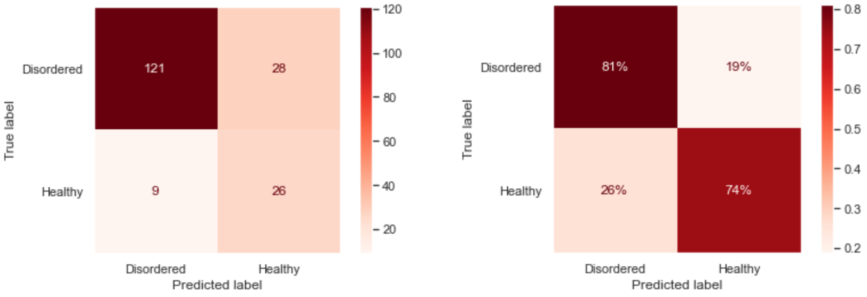


Figure 4.15: Loss and accuracy of the model on the training set during 100 epochs of training.

the model with respect to confidence assigned by physicians is shown in figure 4.17, performance on high-confidence hypnograms is accurate with 94% and 100% on the disordered and healthy respectively. The model performs relatively well on the low-confidence healthy ones with 78% accuracy, which is higher than on the medium-confidence hypnograms. In case the physicians agreed on their evaluation, the model correctly predicts the evaluation in 90% of the cases, in case of disagreement this is 52%. The ROC curve of the FCN model, which has



(a) Absolute counts.

(b) Normalized over true label.

Figure 4.16: Confusion matrix for the FCN model on the test data.

an AUC of 0.84, is shown in figure 4.18.

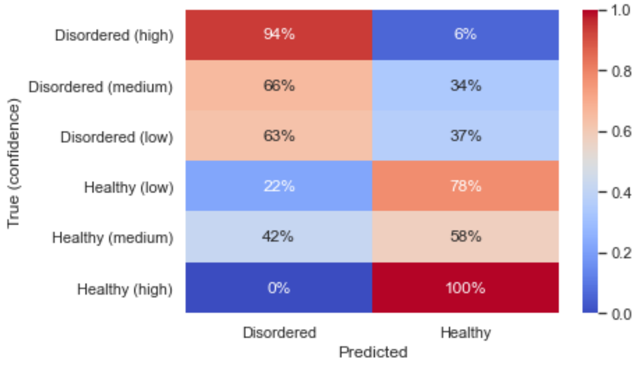


Figure 4.17: Performance of the FCN model on the test data with respect to physician confidence.

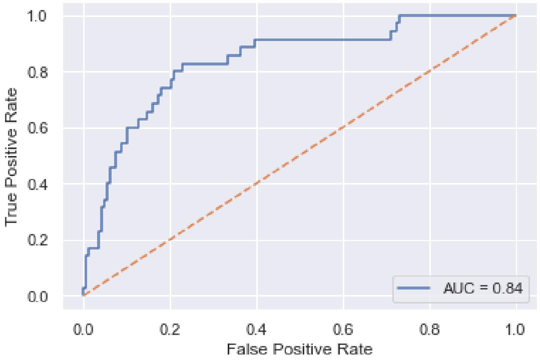
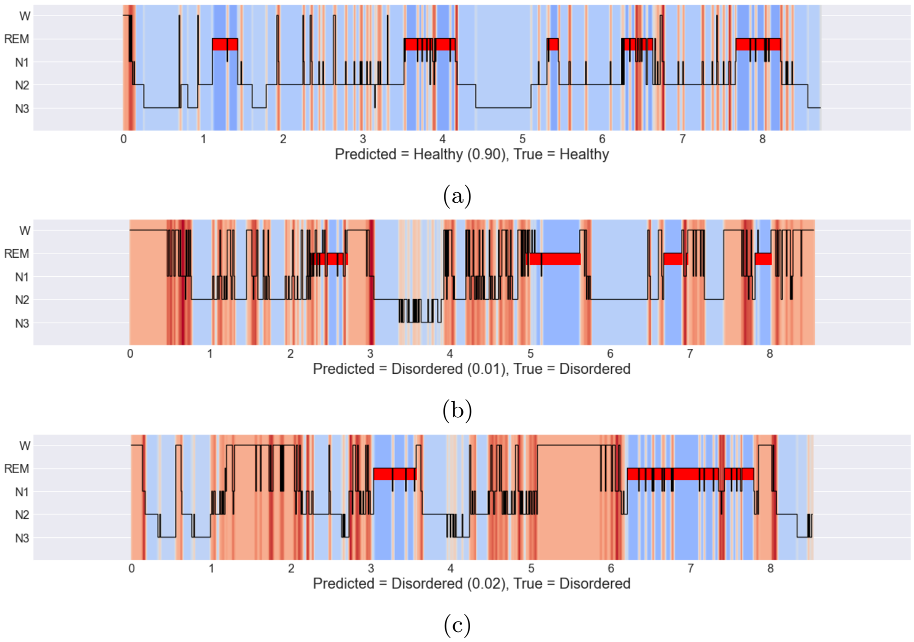


Figure 4.18: ROC curve displaying the trade-off between true positives and false positives across decision thresholds of the binary classification problem for the FCN model.

Class activation maps of three randomly sampled hypnograms are shown in figure 4.19. The blue and red regions in these plots contribute to healthy and disordered predictions respectively. It can be observed that the model distinguishes between different transitions and continuous presence of specific stages, assigning each a different weight. The majority of transitions contributes to disordered prediction, in particular N1 transitions are strongly associated with disordered. Exception to this rule are transitions from N2 to N3, which contribute positively. When a stage is continuously present, N1 and W are negative contributors, the other stages contribute positively.



(a)

(b)

(c)

Figure 4.19: Three hypnograms and their class activation maps, predictions and true label. Blue regions contribute to healthy and red regions to disordered.

4.3.4 Advanced CNN

We experimented with several different architectures, however there was little improvement in terms of performance over the baseline FCN. Changing filter size and strides and/or adding pooling layers, dropout and batch normalization did not yield an improved performance. Furthermore, changing the GAP layer to one or more fully-connected layers improves the theoretical ability of the model to detect complex patterns, but this did not lead to performance improvement. Also, removing the GAP layer would hurt the explainability of the resulting network. As an alternative, we developed a simple, explainable, variant of the baseline FCN that can take into account temporal location.

This model, which we will refer to as advanced CNN, takes as input a hypnogram and splits it into three equal parts (referred to as *early*, *middle* and *late*). Each part is fed into a separate FCN with GAP as previously described in table 4.3. After the GAP layer the results of the separate models are concatenated and a dense layer with sigmoid activation was used to obtain output probabilities for the healthy and disordered classes. CAM was implemented as a simple combination of the CAM of the individual parts.

The same training and test data was used as for the FCN model, resulting in an accuracy of 85% on the training set and 84% on the test set. Interestingly, the loss and accuracy reached a plateau during the first few epochs of training, which is seen in figure 4.20.

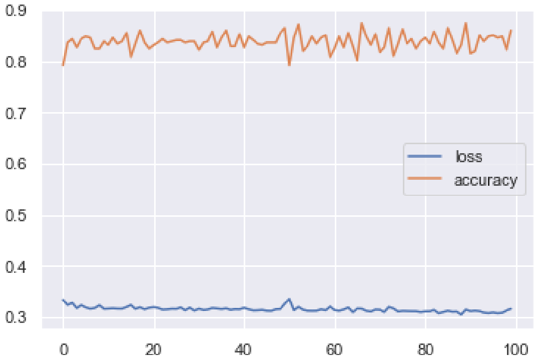
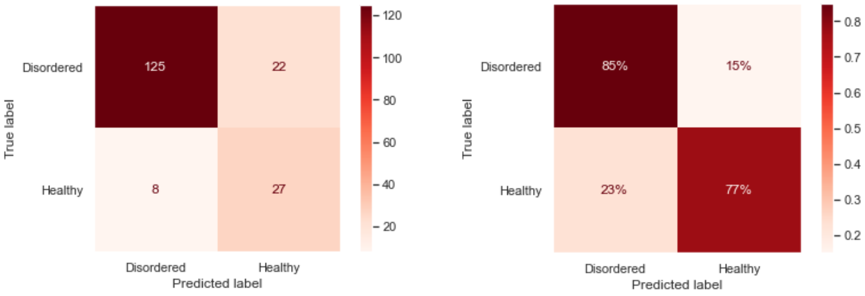


Figure 4.20: Loss and accuracy of the model on the training set during 100 epochs of training.

The confusion matrix that describes the performance of the model on the test data is shown in figure 4.21. The model is 85% and 77% accurate on the disordered and healthy evaluations respectively. The performance of the model with respect to confidence assigned by physicians is shown in figure 4.22, performance on high-confidence hypnograms is accurate with 97% and 100% on the disordered and healthy respectively. Accuracy on low- and medium-confidence hypnograms is lower, between 65 and 75%. For the hypnograms that were evaluated by both physicians, it was found that the model was 91% accurate on the cases where the physicians agree and 55% otherwise. AUC for the the ROC curve is 0.88 as shown in figure 4.23.

Class activation maps for two randomly sampled hypnograms are shown in figure 4.19. Again, the class activation maps show that the model detects transitions and continuous presence of stages, however now the impact is different across the three parts. For example, in the CAM of the hypnogram shown in figure 4.24a it can be seen that REM has a stronger positive contribution at



(a) Absolute counts.

(b) Normalized over true label.

Figure 4.21: Confusion matrix for the model on the test data.

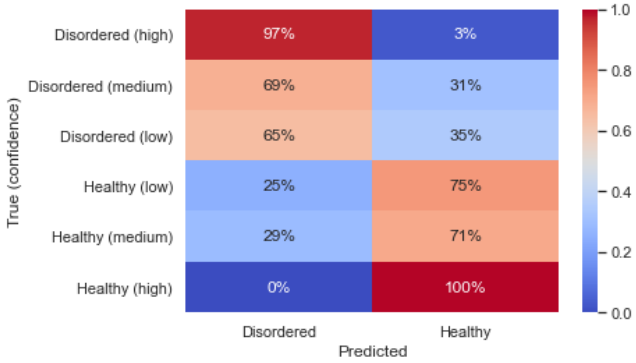


Figure 4.22: Performance of the model on the test data with respect to physician confidence.

the start of the night than at the end of the night. In order to systematically identify these patterns we passed an artificial hypnogram through the network where each epoch describes the same stage (e.g. a hypnogram that is W at each epoch) and evaluate the outcome at the three parts. This was done for each of the five stages, the resulting impact scores for each of the five stages across the early, middle and late parts are shown in figure 4.25. The model learned that W most strongly contributes to disordered evaluation when occurring in the middle of the hypnogram. REM, N2 and N3 contribute positively in the first part of the hypnogram, neutral in the middle and N3 even contributes negatively when occurring late in the hypnogram. Presence of N1 contributes negatively during the middle and late parts of the hypnogram.

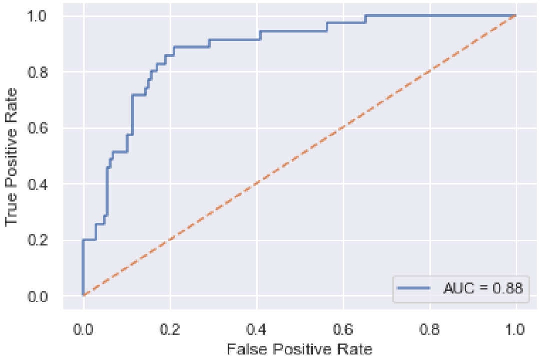


Figure 4.23: ROC curve displaying the trade-off between true positives and false positives across decision thresholds of the binary classification problem.

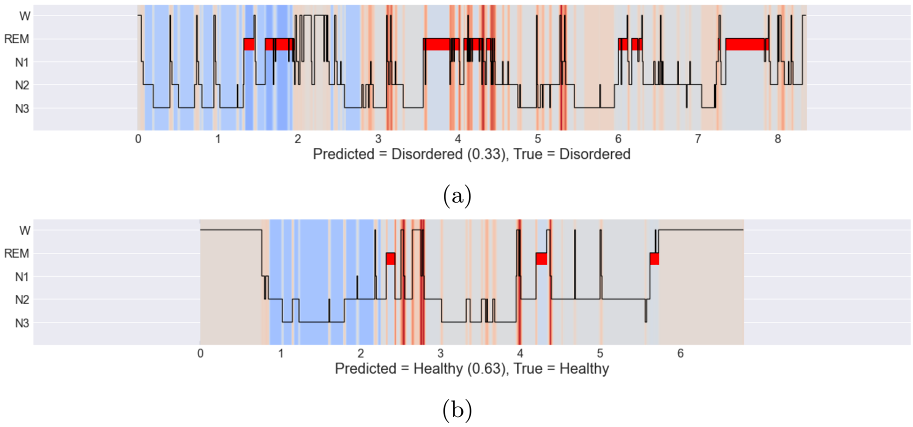


Figure 4.24: Two hypnograms and their class activation maps, predictions and true label. Blue regions contribute to healthy and red to disordered.

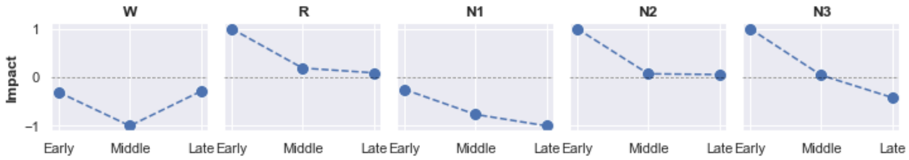


Figure 4.25: Impact of continuous presence of each of the stages during the three parts of the night. Impacts were obtained by passing an artificial hypnogram, describing the same stage at each epoch, through the network and evaluating the contribution at each part. Points above the dotted zero line contribute to healthy predictions and below the dotted line to disordered. For each stage, the impacts were scaled by the maximum absolute value across the three parts.

Chapter 5

Discussion

Interpretation of the hypnogram is done using "clinical intuition" and relies on visual pattern recognition. Therefore, our goal was to identify and interpret aspects within and across hypnograms that contribute to interpretation by physicians. For this purpose, a data collection was conducted to obtain subjective evaluations for hypnograms. Machine learning and visualization methods were used to gain insights into the collected evaluations. In this chapter, we discuss the implications of our results and reflect on the research questions that we defined in the first chapter.

The physicians rated most of the hypnograms as disordered and systematically indicated high confidence for disordered evaluations and low confidence for healthy evaluations. Even when the subject was healthy, the sleep structure was evaluated as abnormal with a suspected disorder in approximately half of the cases. This can be explained by first-night effects that impact the sleep during first night of PSG. Alternatively, a partial explanation might be that the notion of normal sleep structure is based on textbook examples that are rarely seen in practice. In any case, the ability to identify healthy subjects from the hypnogram itself is limited.

The visualization results illustrate that characteristics of the hypnogram which are relevant for physician interpretation are best captured by features computed over the full hypnogram. The similarity measure for the original representation is arguably naive; but the windowed representation, even when used with dynamic time warping, could also not compete with the strength of the features. In the t-SNE plots, interspersed patterns are seen with respect to evaluations and diagnosis, which implies that hypnograms can be similar in terms of features even though they are associated with different diagnoses and/or evaluations. In general, confidence is lowest and disagreement is highest towards the center of the t-SNE space that was shown in figure 4.6. This is also the area where the healthy and disordered hypnograms are the most interspersed. These hypnograms can therefore best be understood as edge cases that contain characteristics of both disordered and healthy sleep structure. There were no additional strong clusters of hypnograms that were associated with high-disagreement.

Overall, presence of REM, N2 and N3 contribute to healthy evaluation and presence of W, N1 and fragmentation contribute to disordered evaluation. The four classification models align on these observations and are able to describe the classes with 77-84% accuracy on hypnograms that were unseen during training. Note that the evaluations are subjective and the classes are therefore not expected to be perfectly separable. Accuracy on hypnograms evaluated with high confidence and where physicians agreed is between 90 and 100%, which emphasizes the effectiveness of the models and the subjective nature of the labels. These results are in accordance with the model for normal/abnormal hypnograms that was described by Amouh [4]. The decision tree, logistic and FCN model use only simple features of the hypnogram, nevertheless they accurately captured the two classes. The advanced CNN model, distinguishes between features that occur in the early, middle and late parts of the hypnogram. Including those temporal patterns leads to a slight improvement in all evaluation metrics over the baseline CNN. Stages contribute positively when occurring continuously in the early part of the hypnogram (N2, N3 and REM), negatively when occurring in the middle (W and N1) or in the end (N1 and N3). It should be noted that this model captures quite specific patterns. On the other hand, physicians gave a quick evaluation and might have captured less detail. Therefore this model might overemphasize these local patterns that might be more descriptive of the structure in the hypnograms rather than the evaluations.

The hypnogram can essentially be seen as a simplified representation of a PSG recording. This simplification is required as PSG in itself is too complex and large to interpret. However, forcing a rater (either a sleep technician or an algorithm) to take a stance on a sleep stage while the underlying PSG signal is continuous, might lead to missing or over-scored fragmentation. Our results, show that fragmentation is an important driver for physician interpretation, therefore this can be problematic. This limitation of the hypnogram and the ability of computer models to process and analyze high-dimensional data at low-cost was a motivation for other researchers to propose alternative representations for the hypnogram with a higher temporal resolution and describing a distribution of stages at each timestamp (i.e. a hypnodensity plot) [32, 43]. These representations contain more information than the hypnogram which makes them more suitable for computer analysis, this is particularly relevant when the amount of available sleep recordings increases and automatic analysis becomes more relevant.

To finalize the discussion, we reflect on the research questions that were defined in section 1.1.

1. *How can visualization and machine learning be used to gain insight into a large number of hypnograms?*

   In our work, we demonstrated how t-SNE can be applied to visualize a large number of hypnograms simultaneously, features computed over the full hypnogram were found to be most effective for our application. Similarly, features were used in other works that tried to automatically label hypnograms [4, 9, 44]. Nevertheless, a sliding window representation can be more effective for applications where local patterns in the hypnogram are relevant. Additionally, we showed that by one-hot encoding a hypnogram it can be used as input for neural networks. Other methods exist that can be used to gain insight into hypnograms, such as small multiple visualization or visual analytics frameworks for (discrete) time-series [45, 3]. We believe that such methods will become increasingly relevant for hypnograms when wearable sleep trackers will enable collecting hypnograms at a larger scale an over multiple nights.

2. *Which features of the hypnogram drive interpretation by physicians?*

   Transition dynamics and distribution of stages was found to accurately model the evaluations of the physicians. Presence of REM, N2 and N3 contribute to a positive interpretation whereas W, N1 and fragmentation are negatively contributing factors.

3. *Are there previously unknown features of the hypnogram that are associated with interpretation? Can these features be used for analysis and/or assessment of hypnograms?*

   From our advanced CNN results we identified that temporal location of features can further explain the interpretations of the physicians. More specifically, at the beginning of the hypnogram N2, N3 and REM are particularly strong contributors. In the middle of the hypnogram W and N1 are negative contributors and at the end of the hypnogram N1 and N3 are negative contributors.

   However, these observations are what was learned by the model and might capture too specific details that were not necessarily motivations for the physician. Moreover, the other models showed that fragmentation and stage distribution over the full hypnogram are already strong predictors. Therefore, the gain of incorporating novel features to account for interpretation will be relatively small with respect to the known features.

4. *Which factors determine and influence disagreement between physicians and certainty within physicians?*

   In general, physicians were more uncertain on healthy evaluations. Agreement and confidence were high for the strongly fragmented hypnograms with high AHI, which were all evaluated as disordered. Agreement between physicians was 80%, but it should be noted that there was 10 percentage point difference between the physicians on how many hypnograms were evaluated as healthy. Not surprisingly, it was observed that many of the cases where the physicians disagreed are edge cases, i.e. they contain both characteristics of disordered and healthy. Visualization of high-disagreement hypnograms did not reveal strong clusters.

5.1 Limitations

The current research is subject to some limitations. First of all, only hypnograms that were collected and scored at Kempenhaeghe were considered. Since Kempenhaeghe is a specialized centre, the data might include more extreme cases than seen at a regular hospital. Moreover, sleep technicians at Kempenhaeghe might follow the AASM rules more strictly compared to less specialized sleep centres and therefore obtain more fragmented hypnograms.

Second, the hypnograms were considered under a restricted research setting that is in many ways different from a clinical setting. Other information such as the background of a patient, AHI, snoring, body position and subjective sleep quality was not included, which would likely change the way the physician looks at the hypnogram. For example, interpretation of fragmented sleep might be different when a physician knows whether the fragmentation is associated with apnea events. In addition, the hypnograms were evaluated at a high pace, therefore features that stand out more at first glance might be overemphasized in the current research. Therefore, we suspect that in practice, more specific local features of the hypnogram might be more relevant.

5.2 Future work

- Increasing amounts of (wearable) sleep trackers are available to consumers, presenting them with hypnograms. However, a hypnogram is difficult to interpret and requires experience and domain knowledge. Consequently the hypnograms presented by these sleep trackers might drive wrong interpretation. Therefore, future research might use our results and aim to identify how people that are not specialists in the area of sleep can be supported in correctly interpreting results of surrogate sleep trackers.

- Similarly, the current results can be used to identify a measure that quantifies whether two hypnograms are similar with respect to features that drive physician interpretation. This can be used to ensure that automatic sleep staging algorithms are optimized for reconstructing a hypnogram that contains clinically relevant features, rather than just optimizing for overall similarity. In addition, such a metric could be used for applications of unsupervised learning on hypnograms ranging from t-SNE to variational autoencoders. For example, to identify outlier hypnograms in multi-night recordings.

- The discrete nature of the hypnogram introduces error (e.g. missing or over-scored fragmentation). This limitation of the hypnogram and the ability of computer models to process and analyze high-dimensional data at low-cost was a motivation for other researchers to propose alternative representations for the hypnogram with a higher temporal resolution and describing a distribution of stages at each timestamp (i.e. a hypnodensity plot) [32, 43]. Based on our results we believe that such representations can be potentially beneficial. Future work could be done to associate these alternative representations with interpretation by physicians, similar to the current study. This future work could aim to identify whether using these representations alongside/instead of hypnograms can support/improve clinical decision making. After all, the hypnogram or any alternative representation should support physicians in assessing sleep structure and/or diagnosing patients.

References

[1] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah. Time-series clustering–a decade review. *Information Systems*, 53:16–38, 2015.

[2] M. Ali, A. Alqahtani, M. W. Jones, and X. Xie. Clustering and classification for time series data in visual analytics: A survey. *IEEE Access*, 7:181314–181338, 2019.

[3] M. Ali, M. W. Jones, X. Xie, and M. Williams. Timecluster: dimension reduction applied to temporal data for visual analytics. *The Visual Computer*, 35(6):1013–1026, 2019.

[4] T. Amouh. *Analysis of tabular non-standard data with decision trees, and application to hypnogram-based detection of sleep profile.* PhD thesis, Catholic University of Louvain, Louvain-la-Neuve, Belgium, 2011.

[5] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh. The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

[6] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, USA:, 1994.

[7] R. B. Berry, R. Brooks, C. E. Gamaldo, S. M. Harding, C. Marcus, B. V. Vaughn, 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*, 176:2012, 2012.

[8] M. A. Carskadon, W. C. Dement, et al. Normal human sleep: an overview. *Principles and practice of sleep medicine*, 4:13–23, 2005.

[9] R. Chaparro-Vargas, B. Ahmed, N. Wessel, T. Penzel, and D. Cvetkovic. Insomnia characterization: From hypnogram to graph spectral theory. *IEEE Transactions on Biomedical Engineering*, 63(10):2211–2219, 2016.

[10] S. Chokroverty. Overview of normal sleep. In *Sleep disorders medicine*, pages 5–27. Springer, 2017.

[11] D. Cian, J. van Gemert, and A. Lengyel. Evaluating the performance of the LIME and Grad-CAM explanation methods on a LEGO multi-label image classification task. *arXiv preprint arXiv:2008.01584*, 2020.

[12] H. Danker-Hopfe, P. Anderer, J. Zeitlhofer, M. Boeck, H. Dorn, G. Gruber, E. Heller, E. Loretz, D. Moser, S. Parapatics, et al. Interrater reliability for sleep scoring according to the rechtschaffen & kales and the new AASM standard. *Journal of sleep research*, 18(1):74–84, 2009.

[13] H. Danker-Hopfe, D. Kunz, G. Gruber, G. Klösch, J. L. Lorenzo, S.-L. Himanen, B. Kemp, T. Penzel, J. Röschke, H. Dorn, et al. Interrater reliability between scorers from eight european sleep laboratories in subjects with different sleep disorders. *Journal of sleep research*, 13(1):63–69, 2004.

[14] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

[15] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019.

[16] T.-c. Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, 2011.

[17] M. Guillemé, V. Masson, L. Rozé, and A. Termier. Agnostic local explanation for time series classification. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 432–439. IEEE, 2019.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman. 1D convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, 151:107398, 2021.

[20] S. Kriglstein, M. Pohl, and M. Smuc. Pep up your time machine: recommendations for the design of information visualizations of time-dependent data. In *Handbook of human centric visualization*, pages 203–225. Springer, 2014.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[22] M. Längkvist, L. Karlsson, and A. Loutfi. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 42:11–24, 2014.

[23] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[24] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing sax: a novel symbolic representation of time series. *Data Mining and knowledge discovery*, 15(2):107–144, 2007.

[25] Z. C. Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[26] S. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

[27] L. v. d. Maaten. T-SNE. https://lvdmaaten.github.io/tsne/.

[28] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[29] D. Moser, P. Anderer, G. Gruber, S. Parapatics, E. Loretz, M. Boeck, G. Kloesch, E. Heller, A. Schmidt, H. Danker-Hopfe, et al. Sleep classification according to AASM and Rechtschaffen & Kales: effects on sleep scoring parameters. *Sleep*, 32(2):139–149, 2009.

[30] V. Niennattrakul and C. A. Ratanamahatana. Clustering multimedia data using time series. In *2006 International Conference on Hybrid Information Technology*, volume 1, pages 372–379. IEEE, 2006.

[31] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR, 2013.

[32] M. Perslev, S. Darkner, L. Kempfner, M. Nikolic, P. J. Jennum, and C. Igel. U-sleep: resilient high-frequency sleep staging. *NPJ digital medicine*, 4(1):1–12, 2021.

[33] P. E. Rauber, A. X. Falcão, A. C. Telea, et al. Visualizing time-dependent data using dynamic t-SNE. 2016.

[34] M. T. Ribeiro, S. Singh, and C. Guestrin. "Why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[35] N. Ruta, N. Sawada, K. McKeough, M. Behrisch, and J. Beyer. SAX Navigator: Time series exploration through hierarchical clustering. In *2019 IEEE Visualization Conference (VIS)*, pages 236–240. IEEE, 2019.

[36] S. R. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674, 1991.

[37] S. Saha. A comprehensive guide to convolutional neural networks-the eli5 way, Dec 2018. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[38] U. Schlegel, H. Arnout, M. El-Assady, D. Oelke, and D. A. Keim. Towards a rigorous evaluation of XAI methods on time series. *arXiv preprint arXiv:1909.07082*, 2019.

[39] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[40] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. 2014.

[41] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[42] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[43] J. B. Stephansen, A. N. Olesen, M. Olsen, A. Ambati, E. B. Leary, H. E. Moore, O. Carrillo, L. Lin, F. Han, H. Yan, et al. Neural network analysis of sleep stages enables efficient diagnosis of narcolepsy. *Nature communications*, 9(1):1–15, 2018.

[44] B. J. Swihart, B. Caffo, K. Bandeen-Roche, and N. M. Punjabi. Characterizing sleep structure using the hypnogram. *Journal of Clinical Sleep Medicine*, 4(4):349–355, 2008.

[45] B. J. Swihart, B. Caffo, B. D. James, M. Strand, B. S. Schwartz, and N. M. Punjabi. Lasagna plots: a saucy alternative to spaghetti plots. *Epidemiology (Cambridge, Mass.)*, 21(5):621, 2010.

[46] E. Tufte. The visual display of quantitative information, 2001.

[47] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.

[48] M. M. van Gilst, J. P. van Dijk, R. Krijn, B. Hoondert, P. Fonseca, R. J. van Sloun, B. Arsenali, N. Vandenbussche, S. Pillen, H. Maass, et al. Protocol of the SOMNIA project: an observational study to create a neurophysiological database for advanced clinical sleep monitoring. *BMJ open*, 9(11):e030996, 2019.

[49] M. Walker. *Why we sleep: Unlocking the power of sleep and dreams.* Simon and Schuster, 2017.

[50] Z. Wang, W. Yan, and T. Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE, 2017.

[51] L. Wei and E. Keogh. Semi-supervised time series classification. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 748–753, 2006.

[52] K. Y. Wong and F.-l. Chung. Visualizing time series data with temporal matching based t-SNE. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[53] N. Xie, G. Ras, M. van Gerven, and D. Doran. Explainable deep learning: A field guide for the uninitiated. *arXiv preprint arXiv:2004.14545*, 2020.

[54] Y. Yu, Y. Zhu, S. Li, and D. Wan. Time series outlier detection based on sliding window prediction. *Mathematical problems in Engineering*, 2014, 2014.

[55] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[56] Q. Zhang and S.-C. Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.

[57] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

Appendix A

Original Diagnosis

A.1 Distribution

Distribution of diagnoses before grouping similar diagnoses using the map in appendix A.2. The majority of subjects was diagnosed with obstructive sleep apnea. Many of the original diagnoses occurred only a handful of times in the data.

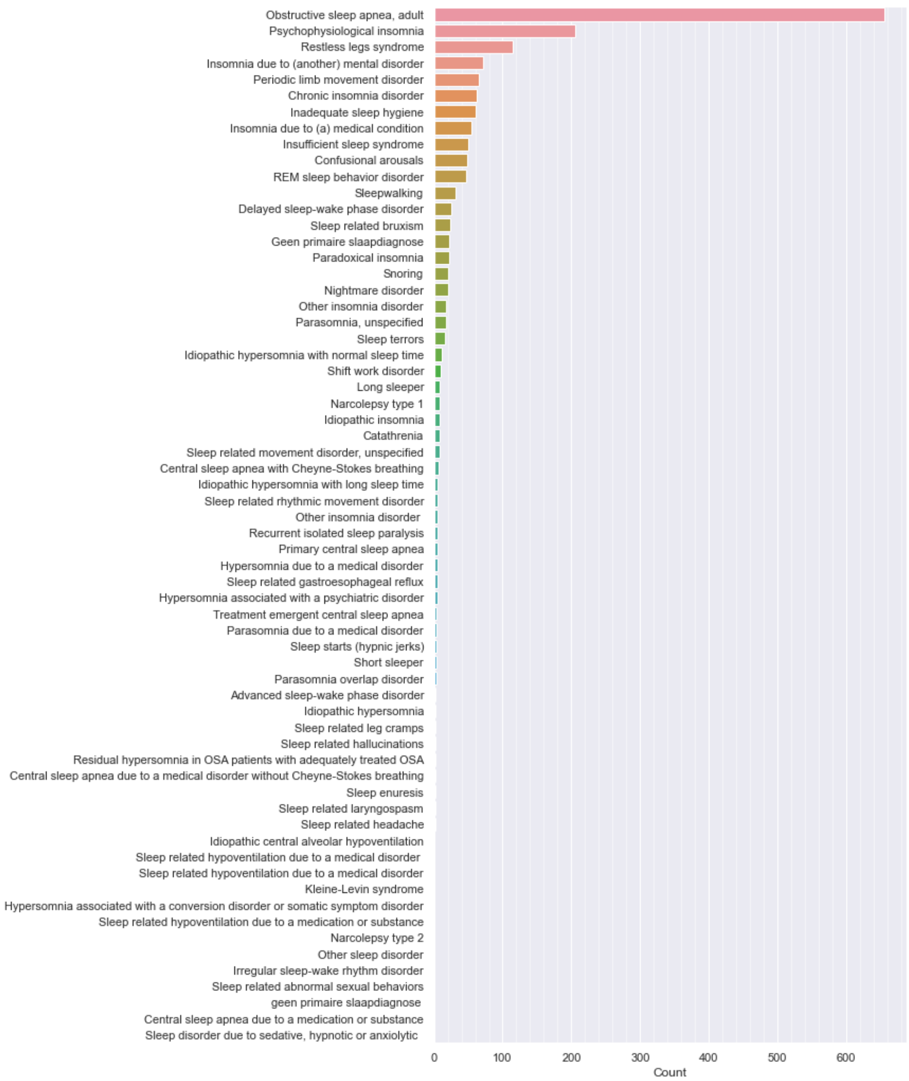


Figure A.1: All diagnoses ordered by frequency. The majority of subjects has been diagnosed with Obstructive sleep apnea. The vast majority of diagnoses occur only a handful of times.

A.2 Mapping

The map below was used to group the large amount of unique diagnoses to groups describing similar diagnoses. The distribution of the original diagnosis can be found in appendix A.1.

Old	**New**
Insufficient sleep syndrome	Behavioral
Inadequate sleep hygiene	Behavioral
Advanced sleep-wake phase disorder	Circadian disorder
Shift work disorder	Circadian disorder
Irregular sleep-wake rhythm disorder	Circadian disorder
Delayed sleep-wake phase disorder	Circadian disorder
Idiopathic hypersomnia with normal sleep time	Hypersomnia
Idiopathic hypersomnia with long sleep time	Hypersomnia
Idiopathic hypersomnia	Hypersomnia
Hypersomnia due to a medical disorder	Hypersomnia
Hypersomnia associated with a psychiatric disorder	Hypersomnia
Kleine-Levin syndrome	Hypersomnia
Narcolepsy type 1	Hypersomnia
Narcolepsy type 2	Hypersomnia
Residual hypersomnia in OSA patients with adequately treated OSA	Hypersomnia
Hypersomnia associated with a conversion disorder or somatic symptom disorder	Hypersomnia
Other insomnia disorder	Insomnia
Other insomnia disorder	Insomnia
Psychophysiological insomnia	Insomnia
Insomnia due to (another) mental disorder	Insomnia
Paradoxical insomnia	Insomnia
Idiopathic insomnia	Insomnia
Insomnia due to (a) medical condition	Insomnia
Chronic insomnia disorder	Insomnia
Sleep related rhythmic movement disorder	Movement disorder
Sleep related movement disorder, unspecified	Movement disorder
Periodic limb movement disorder	Movement disorder
Restless legs syndrome	Movement disorder
Confusional arousals	Non-REM parasomnia
Sleepwalking	Non-REM parasomnia
Sleep terrors	Non-REM parasomnia
Sleep related abnormal sexual behaviors	Non-REM parasomnia
Healthy	None
Geen primaire slaapdiagnose	None
Snoring	Other
Sleep starts (hypnic jerks)	Other
Sleep related leg cramps	Other
Sleep related laryngospasm	Other
Sleep related headache	Other
Sleep related hallucinations	Other
Sleep related gastroesophageal reflux	Other
Sleep related bruxism	Other
Sleep disorder due to sedative, hypnotic or anxiolytic	Other
Sleep enuresis	Other
Catathrenia	Other
Long sleeper	Other
Recurrent isolated sleep paralysis	Other
Other sleep disorder	Other
Short sleeper	Other
Parasomnia, unspecified	Other
Parasomnia overlap disorder	Other
Nightmare disorder	REM parasomnia
REM sleep behavior disorder	REM parasomnia
Parasomnia due to a medical disorder	REM parasomnia
Central sleep apnea with Cheyne-Stokes breathing	Sleep disordered breathing
Treatment emergent central sleep apnea	Sleep disordered breathing
Idiopathic central alveolar hypoventilation	Sleep disordered breathing
Sleep related hypoventilation due to a medication or substance	Sleep disordered breathing
Sleep related hypoventilation due to a medical disorder	Sleep disordered breathing
Sleep related hypoventilation due to a medical disorder	Sleep disordered breathing
Central sleep apnea due to a medication or substance	Sleep disordered breathing
Central sleep apnea due to a medical disorder without Cheyne-Stokes breathing	Sleep disordered breathing
Primary central sleep apnea	Sleep disordered breathing
Obstructive sleep apnea, adult	Sleep disordered breathing

Appendix B

T-SNE Results

The current appendix lists the plots for the t-SNE results as described in section 4.2. Each plot was obtained with a perplexity of 50, learning-rate of 100 and maximum of 5000 iterations. A fixed random state was used to ensure reproducibility of the t-SNE result, the fixed random state was chosen as the t-SNE result with the minimum KL-divergence over 5 runs of the t-SNE algorithm. The results are provided for the feature, original and window representations. The results for the window representation include all five window sizes and the two distance metrics (DTW and window distance) that were described in section 3.2.2.3. The results for standardized window features are not included as no improvement was observed over the non-standardized windows and this would require including an additional 10 figures.

B.1 Feature Representation

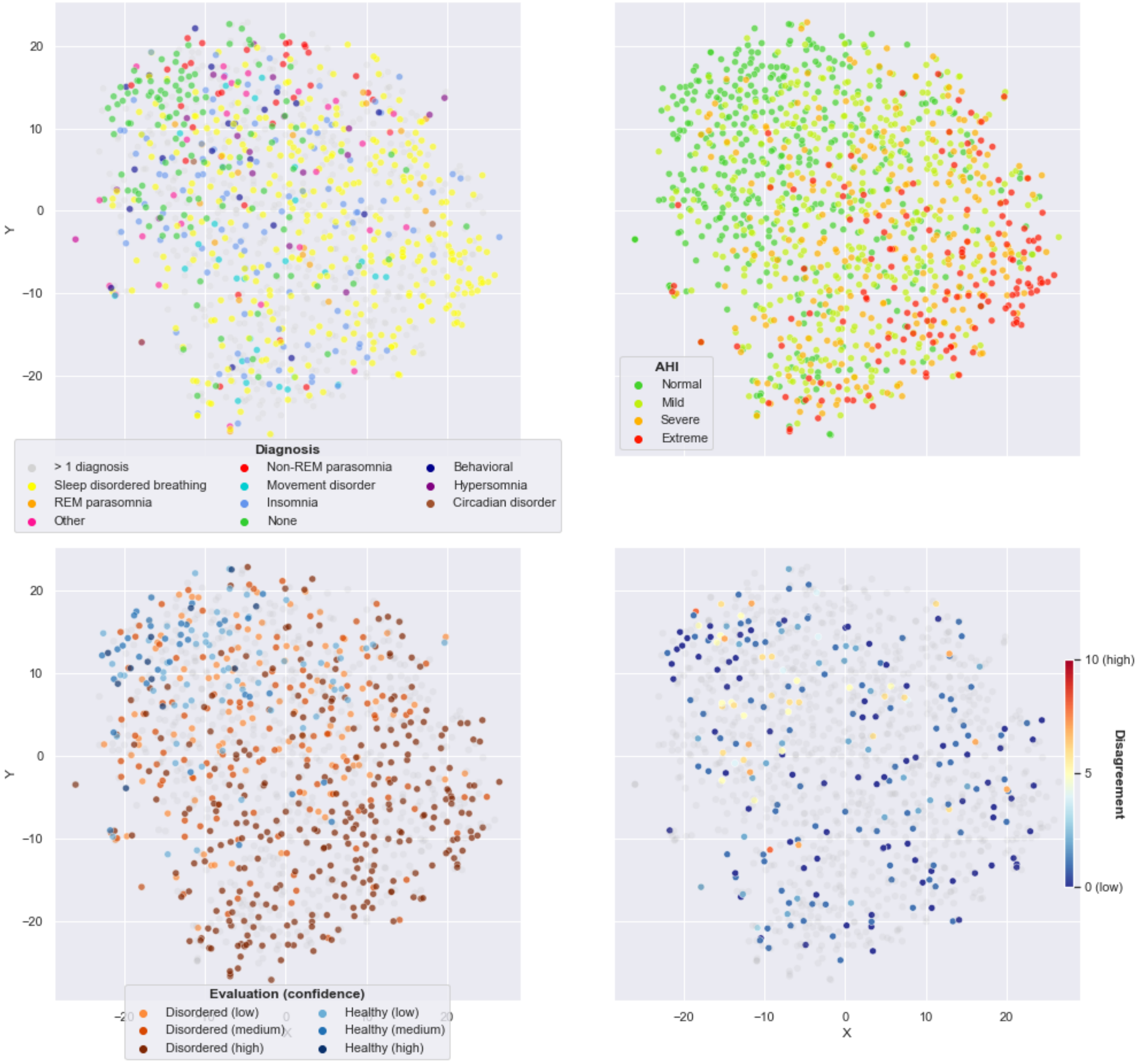


Figure B.1: T-SNE plots for feature representation.

B.2 Original Representation

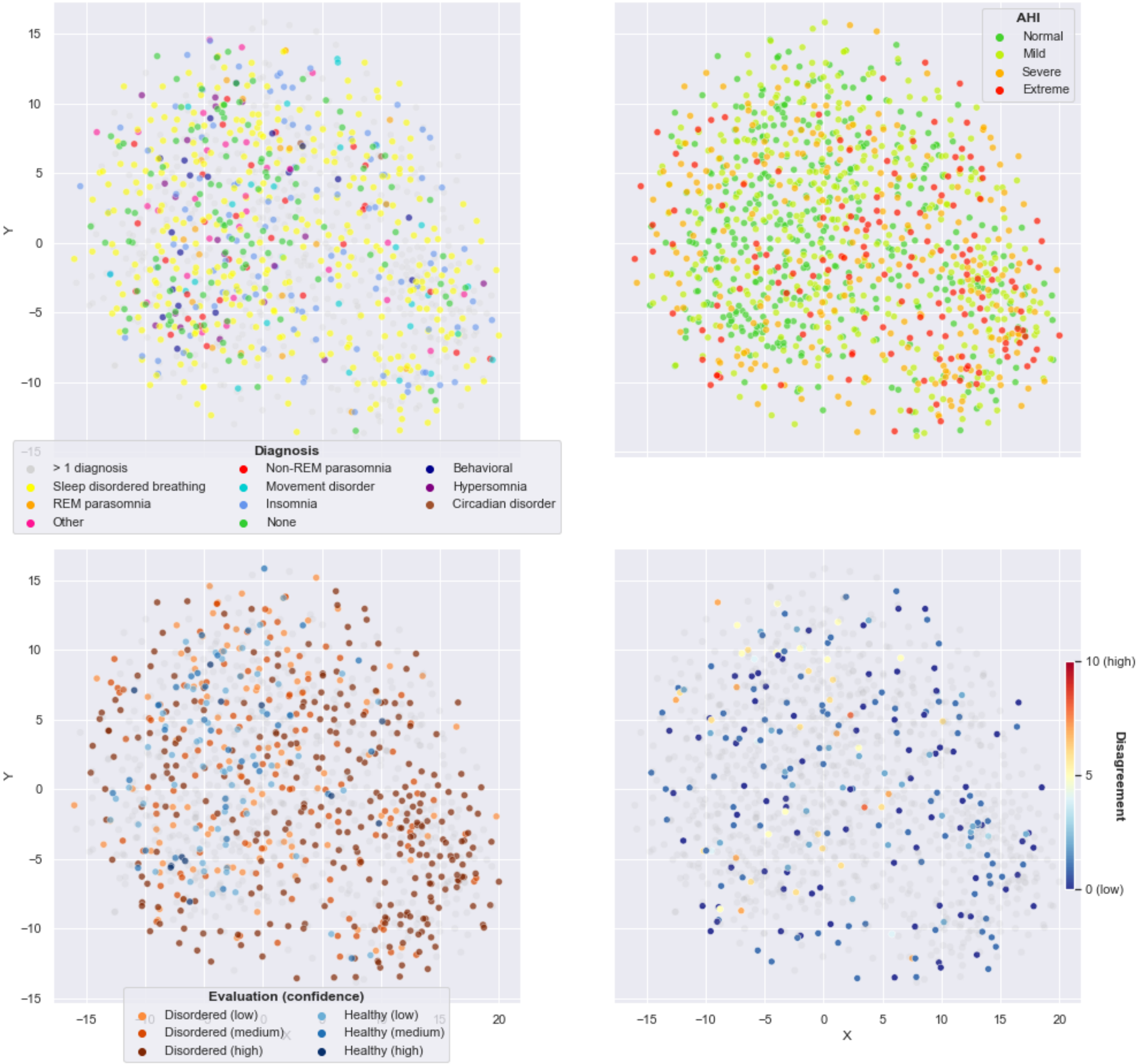


Figure B.2: T-SNE plots for original representation.

B.3 Sliding Window Representation

B.3.1 W15 Window Distance

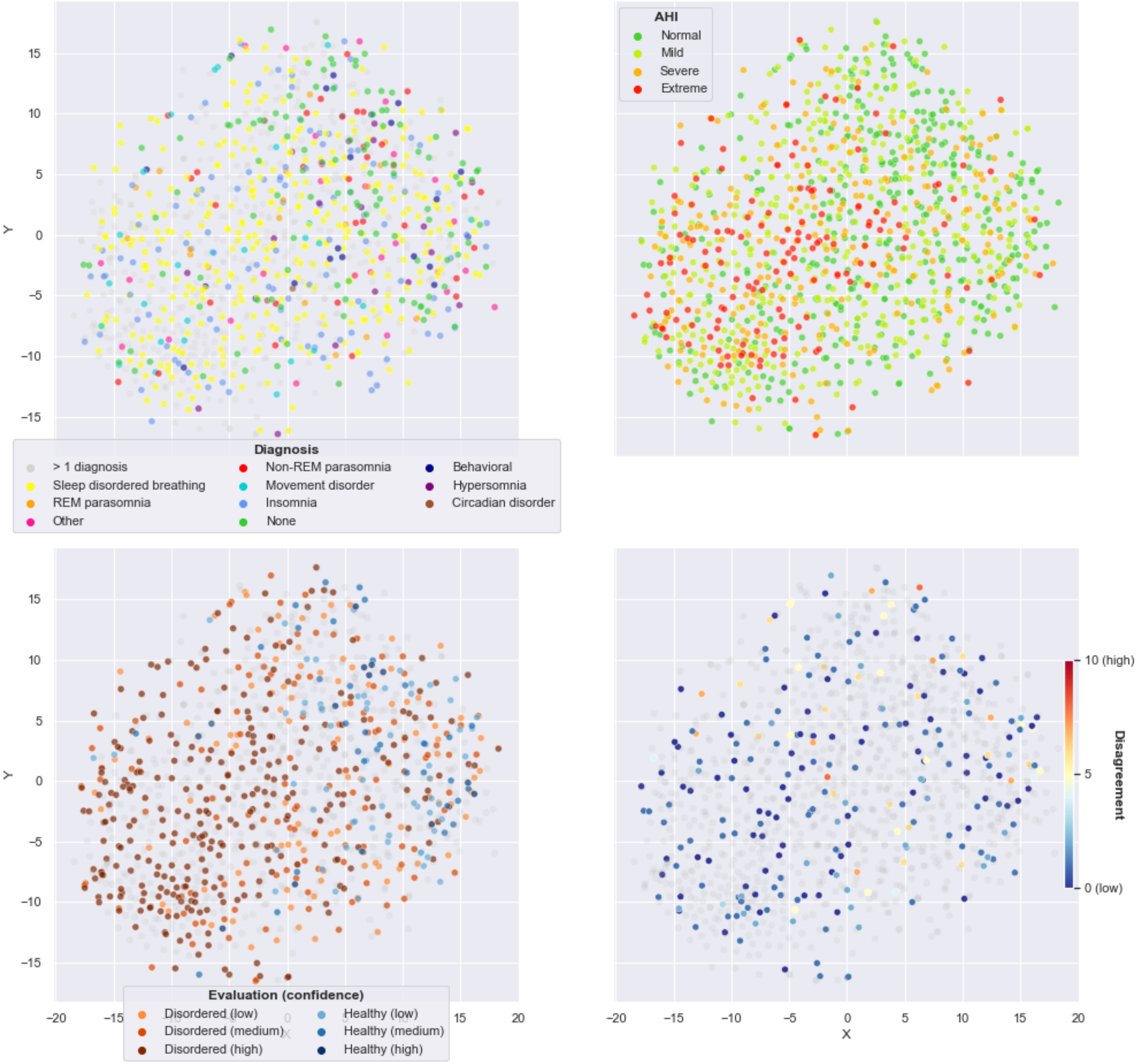


Figure B.3: T-SNE plots for 15 minutes sliding window with window distance.

B.3.2 W15 DTW Distance

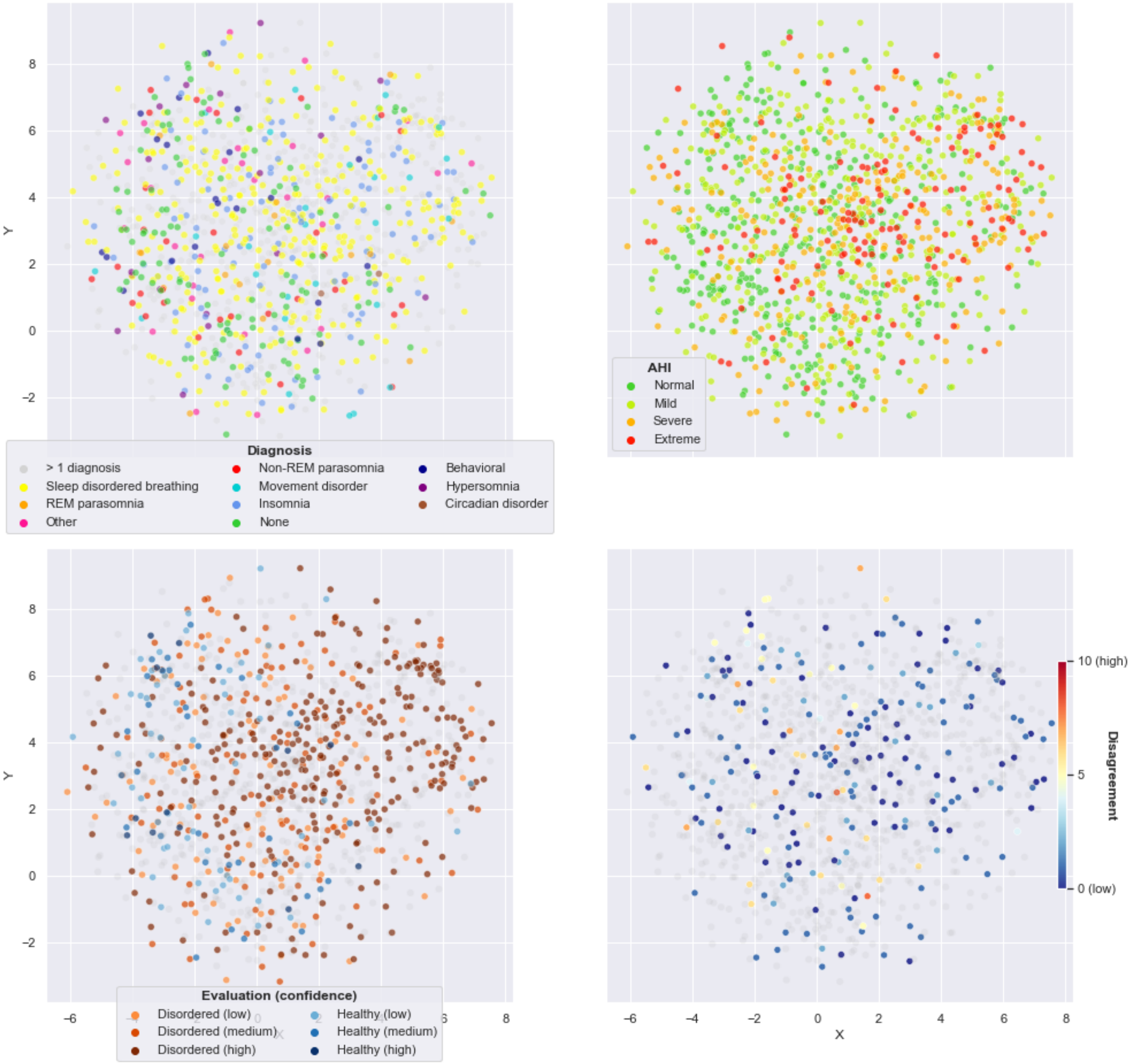


Figure B.4: T-SNE plots for 15 minutes sliding window with DTW distance.

B.3.3 W30 Window Distance

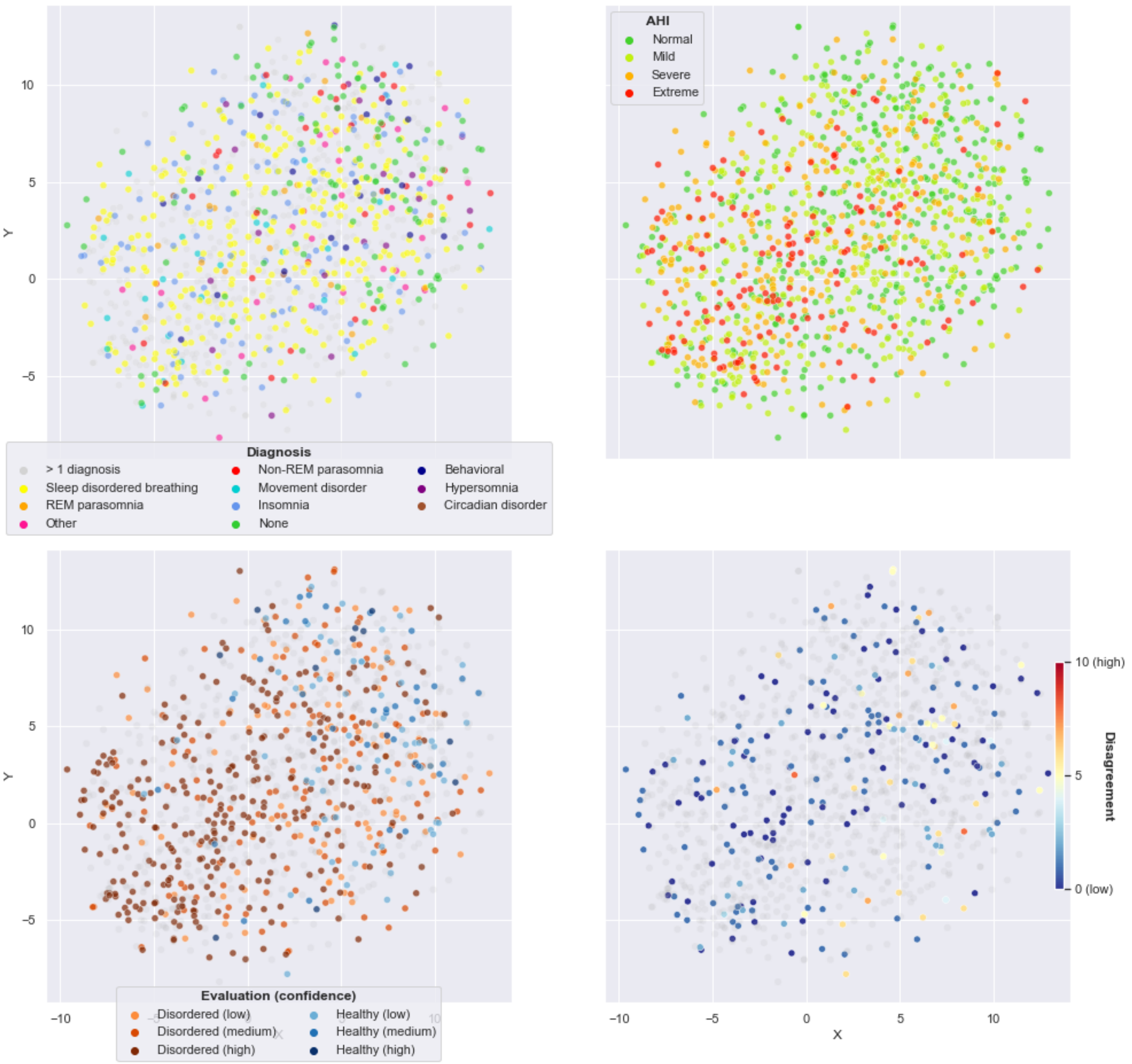


Figure B.5: T-SNE plots for 30 minutes sliding window with window distance.

B.3.4 W30 DTW Distance

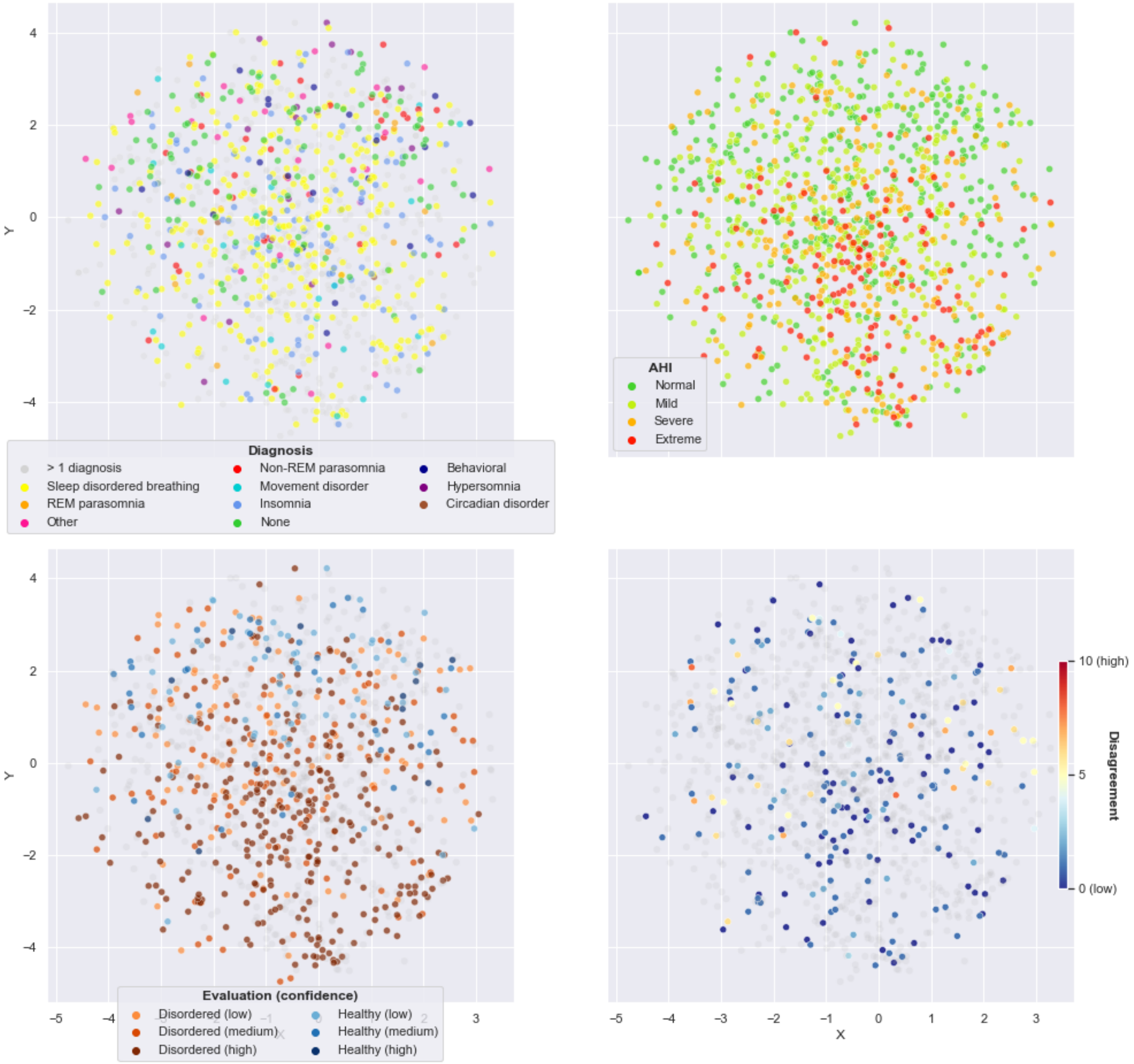


Figure B.6: T-SNE plots for 30 minutes sliding window with DTW distance.

B.3.5 W60 Window Distance

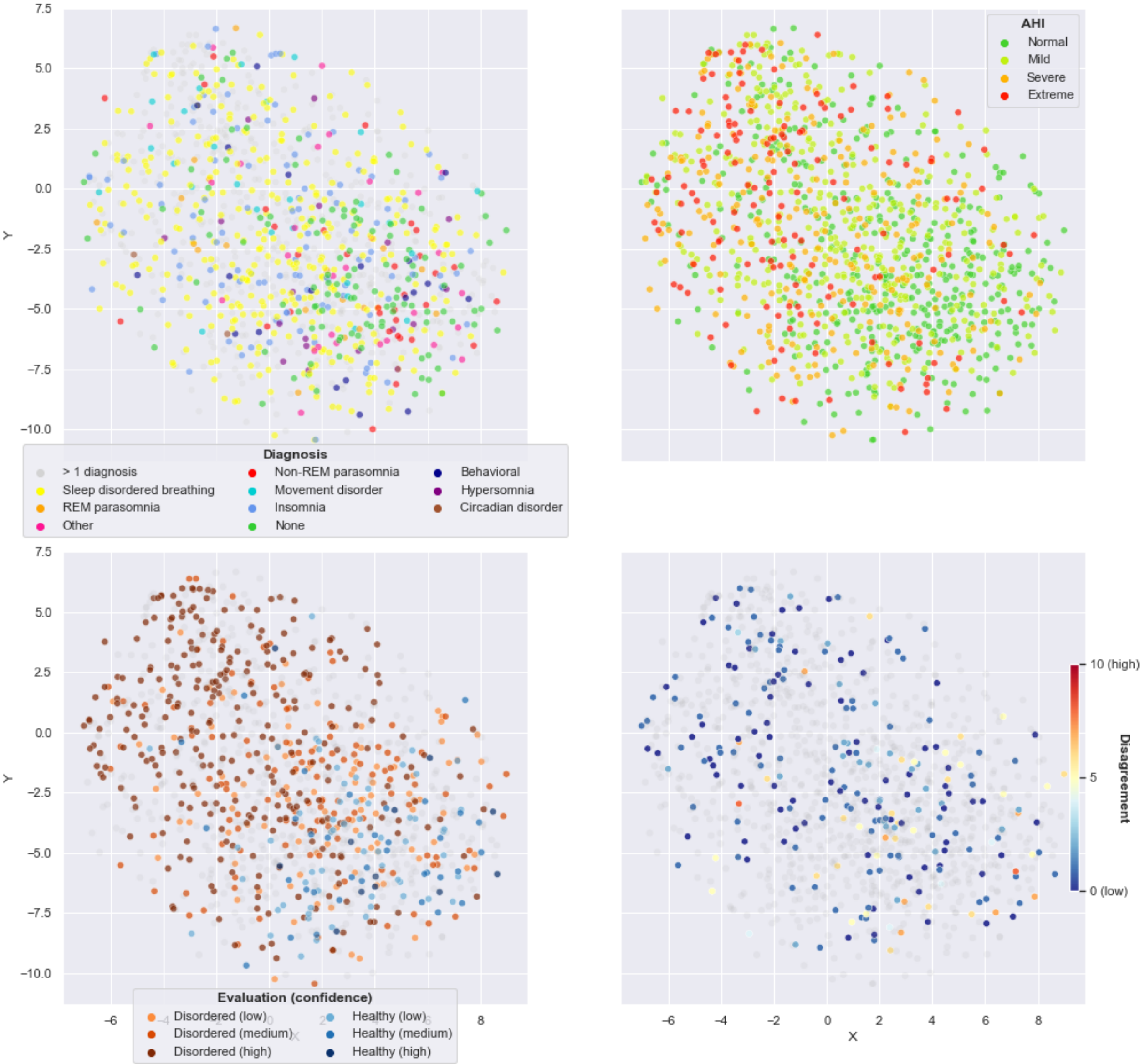


Figure B.7: T-SNE plots for 60 minutes sliding window with window distance.

B.3.6 W60 DTW Distance

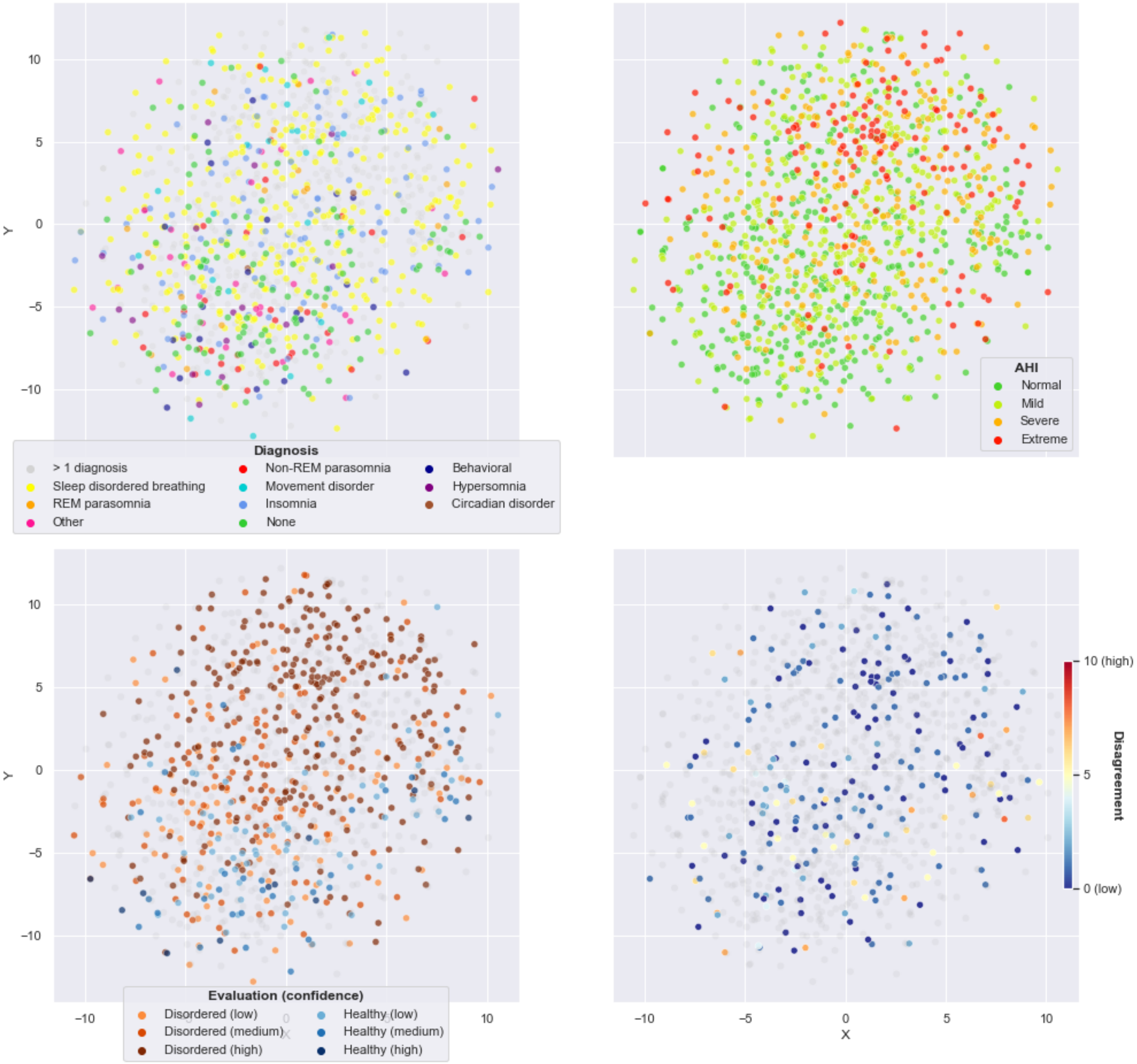


Figure B.8: T-SNE plots for 60 minutes sliding window with DTW distance.

B.3.7 W90 Window Distance

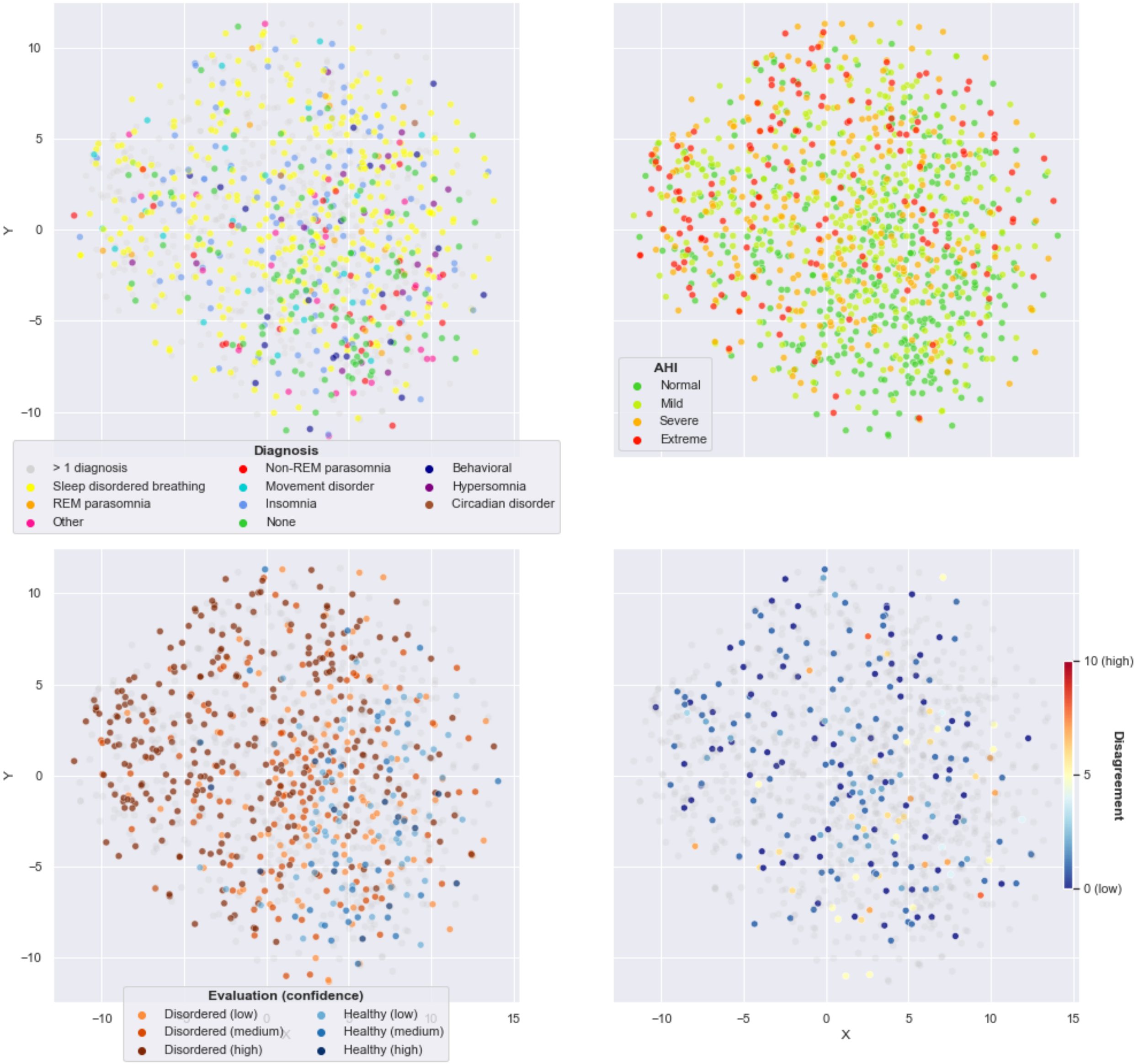


Figure B.9: T-SNE plots for 90 minutes sliding window with window distance.

B.3.8 W90 DTW Distance

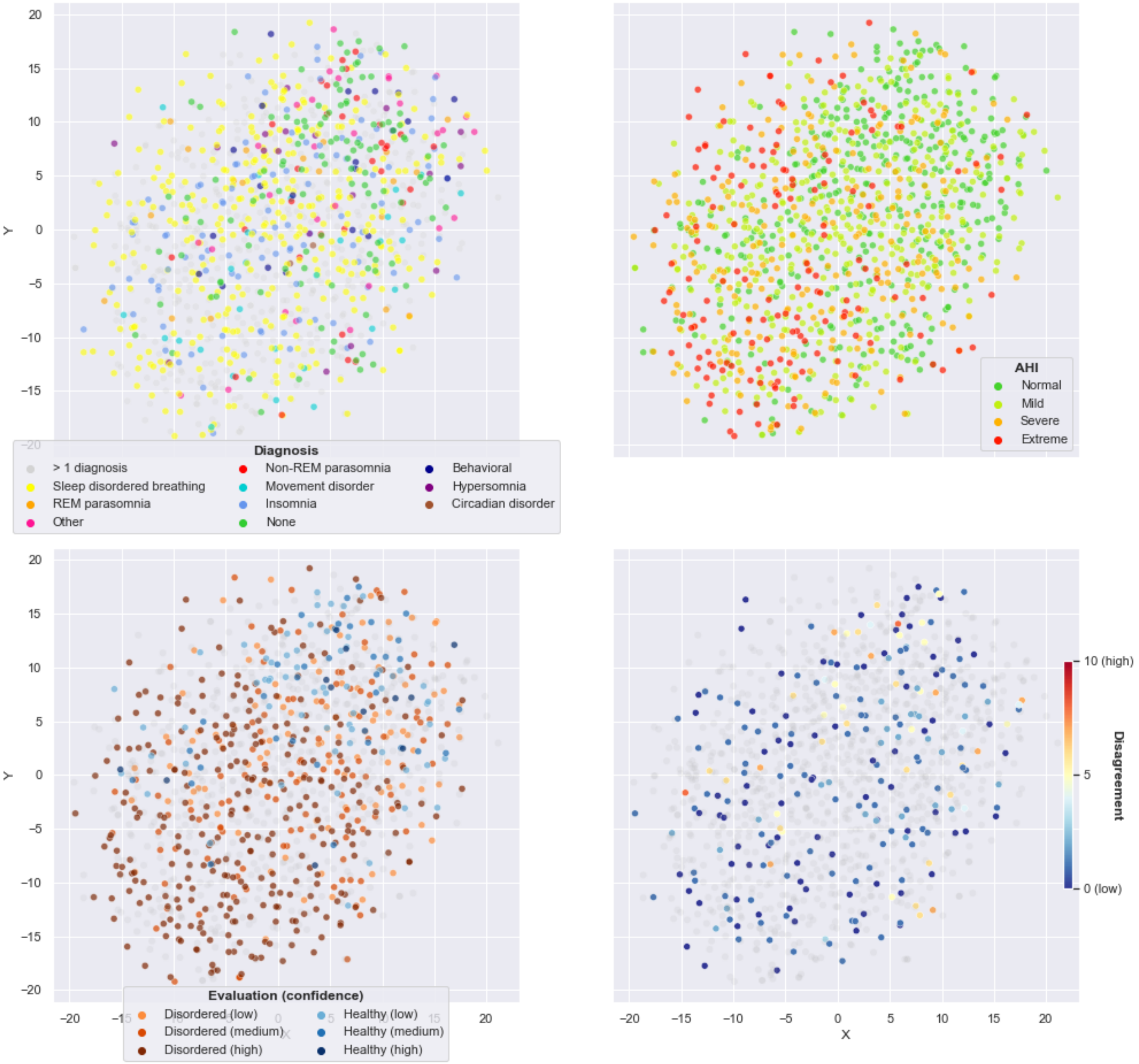


Figure B.10: T-SNE plots for 90 minutes sliding window with DTW distance.

B.3.9 W120 Window Distance

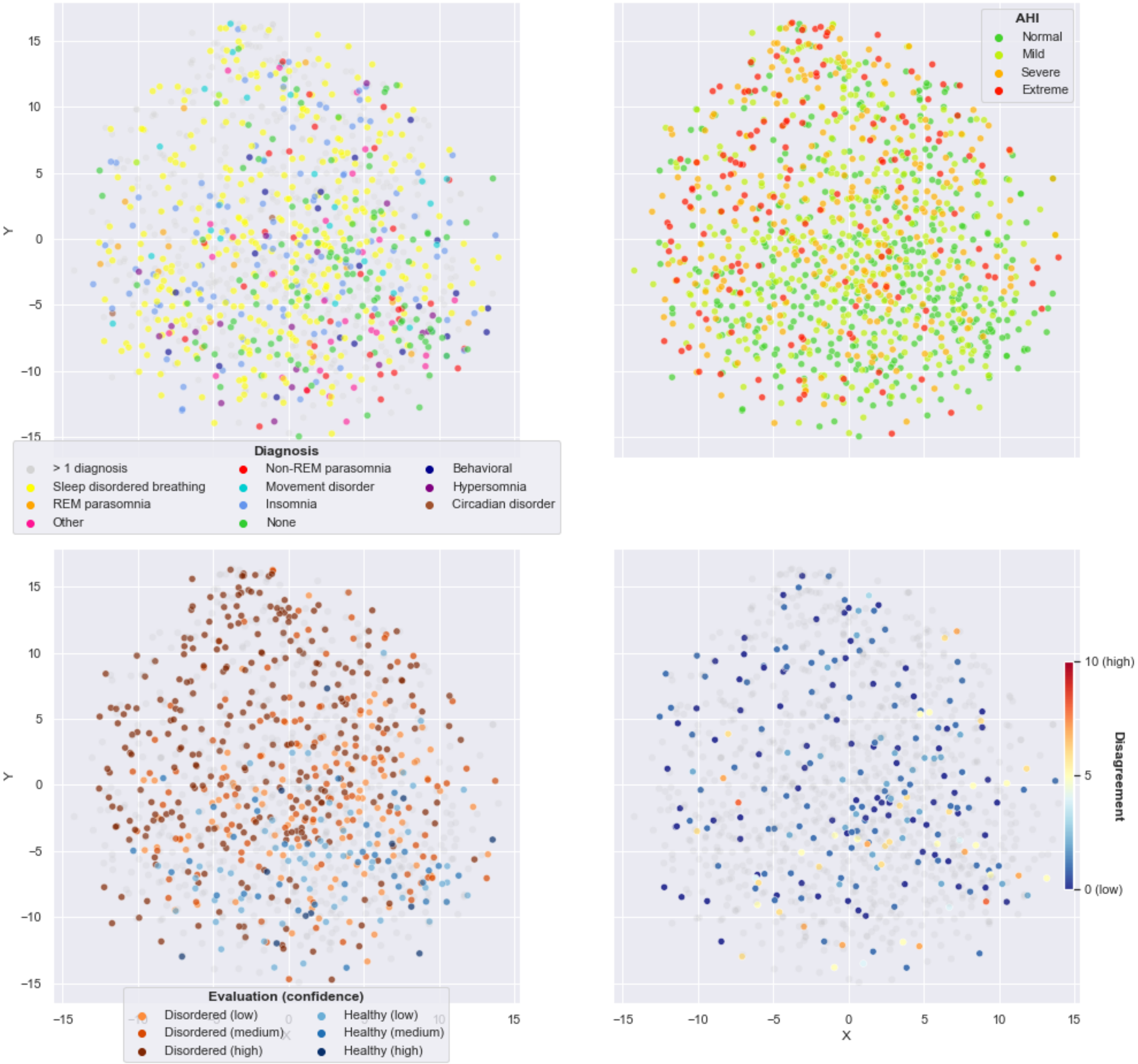


Figure B.11: T-SNE plots for 120 minutes sliding window with window distance.

B.3.10 W120 DTW Distance

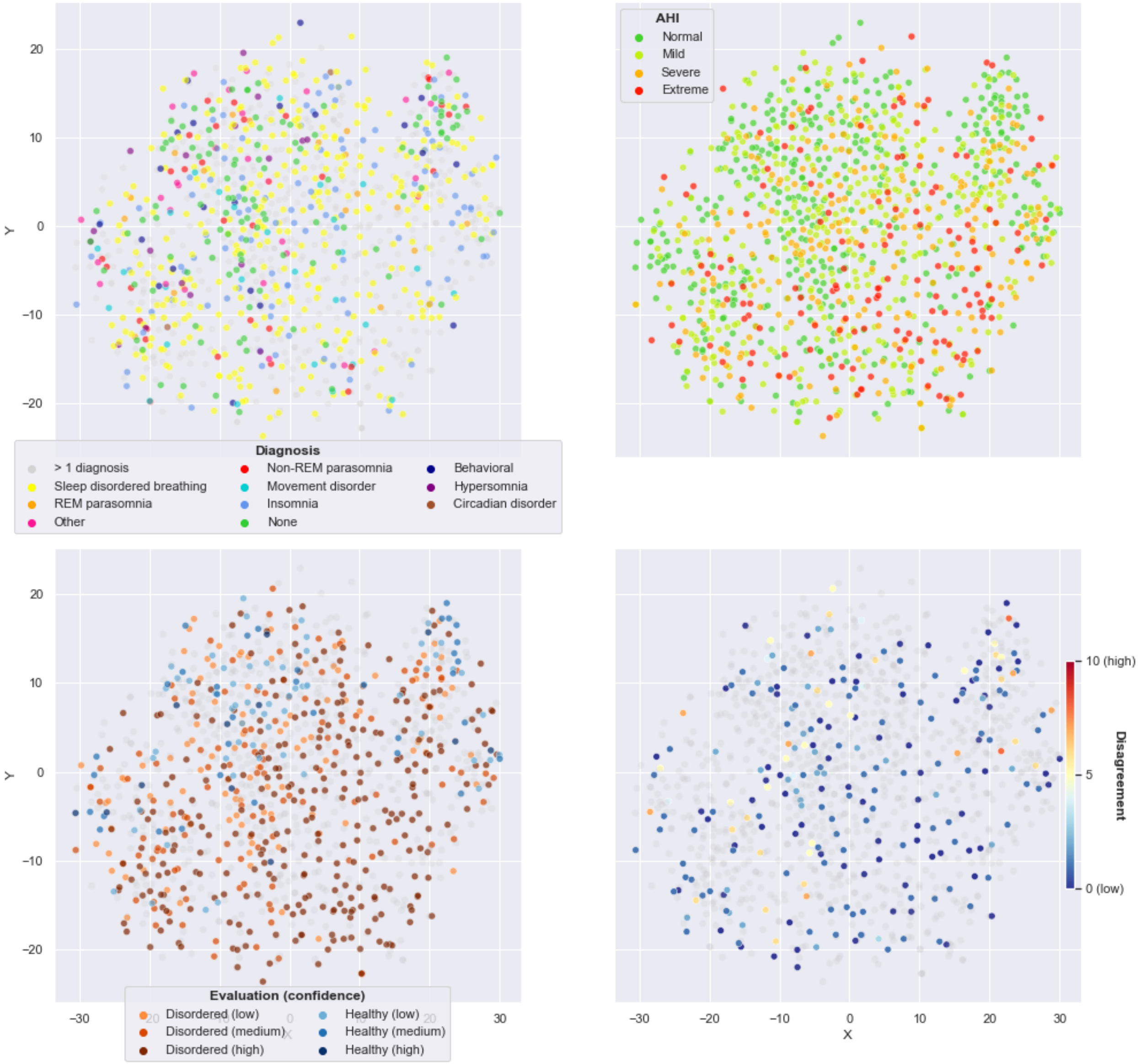


Figure B.12: T-SNE plots for 120 minutes sliding window with DTW distance.

B.4 Feature Representation (Hypnogram markers)

In addition to the t-SNE plots for feature representation shown in appendix B.1 and in the main body in figure 4.6, we used the same t-SNE projection but replaced the circle markers by the original hypnograms. A random sample of 500 hypnograms was taken and plotted in the t-SNE space, this can be seen on the next page in figure B.13. The figure is cluttered, which is to be expected when plotting 500 hypnograms in a single figure. Nevertheless, when zooming in on the areas of the plot, clear patterns emerge.[1]

On the left-upper side profound uninterrupted REM sleep can be observed in many of the hypnograms, little fragmentation is seen. Somewhat more to the top, there appears to be a large amount of N3 awakenings, which could be associated with the clusters of NREM parasomnia in this area, as previously seen in figure 4.6. On the right-side of the figure, there is a clear increase in fragmentation. Finally, on the bottom of the figure it stands out that hypnograms show large amounts of wake. Long sleep-onset times are clustered at the bottom-left.

[1] The figure is included as vector graphics in the digital version of this document, therefore quality is preserved when zooming.

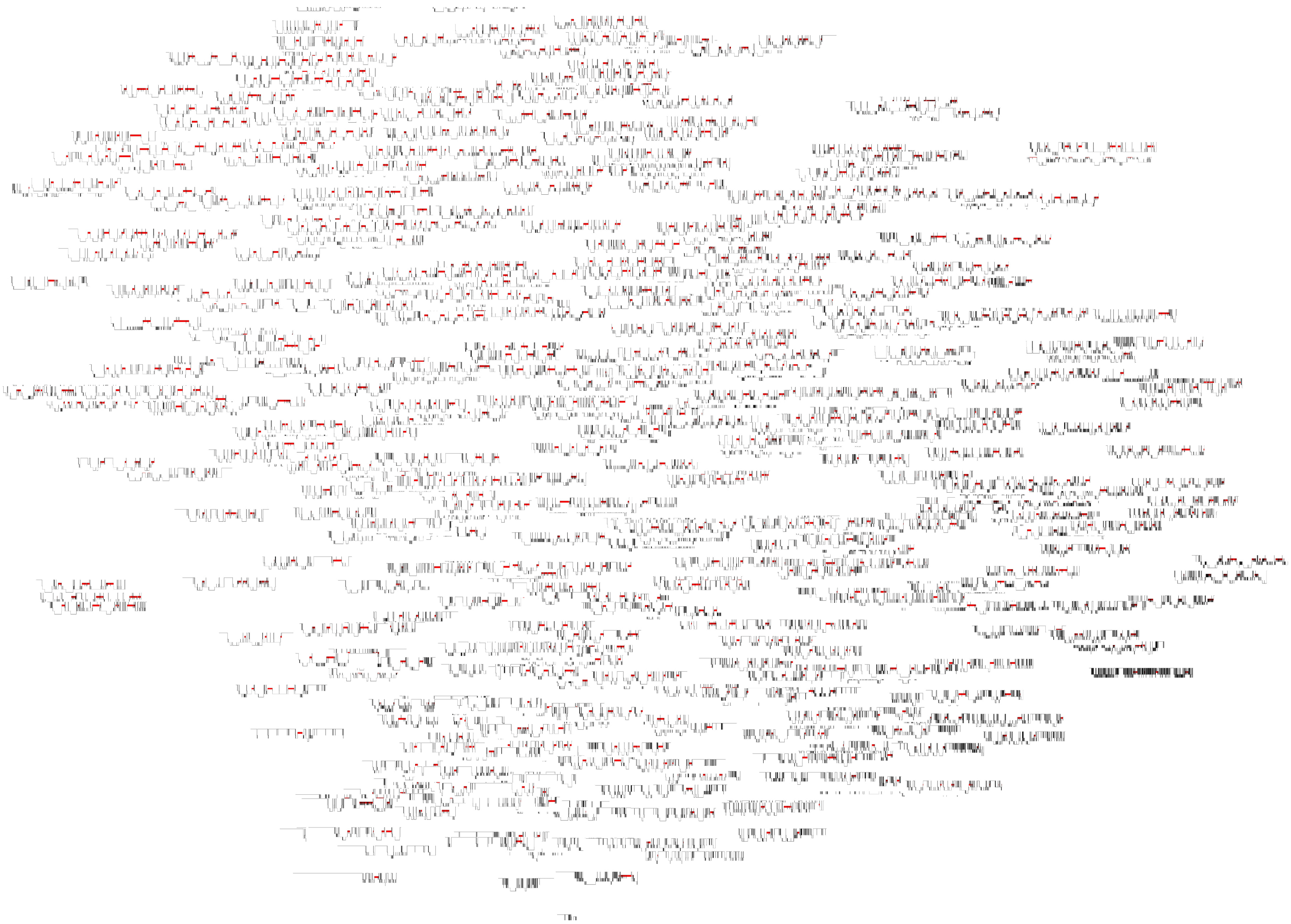


Figure B.13: T-SNE projection of feature representation for 500 randomly sampled hypnograms. The original cropped and scaled hypnograms are used as markers.