Deep learning approach for automated sleep state classification in preterm infants

Jan Werth, Mustafa Radha, Peter Andriessen, Ronald M Aarts, Xi Long

# Abstract

Preterm infant neuronal development is connected to the distribution of their sleep states. The distribution changes over the course of development. Automated sleep stage monitoring can become a powerful aid for development monitoring in preterm infants. Therefore, three datasets including 34 preterm infants and a total of 18018 30s manual annotated sleep intervals are analyzed in this publication. The annotation separated the data into active sleep, quiet sleep, intermediate sleep, wake, and caretaking. Four different recurrent neuronal network architectures are compared for bi-state, tri-state, and all-state analysis. A rather shallow model was used to compare long short term memory and gated recurrent units models. The other network architectures are based on the popular ResNet and ResNext architectures utilizing residual connection for more depth. The most important states active and quiet sleep could be separated well with a kappa of 0.43 ± 0.08. Quiet vs caretaking and wake showed kappa of 0.44 ± 0.01. Active vs quiet vs intermediate sleep resulted in a kappa of 0.36 ± 0.04 and for active vs quiet vs wake and caretaking with a kappa of 0.32 ± 0.09. In contrary, all state classification was underperforming mostly due to lack of sufficient training data for the minority classes.

# Introduction

Preterm infant sleep shows several distinct sleep states. They are defined mainly as active sleep (AS), quiet sleep (QS), and wake. As in early preterm infants sleep states cannot always be undoubtedly identified or show a slow shift from one to the other, often with patterns of both, AS and QS, intermediate or undetermined (IS) states are used [1].   
AS is often compared to the adult rapid eye movement (REM) sleep states since it shows similar increased neural activity. Nevertheless, the role of preterm infant sleep states seems to be different. It is believed that the sleep, the sleep states and the sleep cycles of the fetus, preterm and term infants all play an essential role in the sensory and cortical development. Initially, AS is providing stimulation to the fetus brain in a sensory-reduced environment triggering the development of brain regions with reduced sensory input. Furthermore, during AS the development, integration, and alignment of specific neural tasks/regions into the cortex structure is taking place. During QS, it is reasonably assumed that developmental errors are corrected, and reorganizations are conducted with the use of increased brain plasticity. QS is therefore often seen as the resting and reenergizing state.   
In contrast to the adult REM sleep, AS sleep is dominating the sleep cycle of preterm infants with about 80% at early gestational birth. QS is seen as the minority state with about 18% of the total sleep time. The distribution changes in the course of development. The states can be observed by differences in vital sign, movement, and electroencephalographic (EEG) activity. During active sleep increased cardio-respiratory activity and increased motor movement with sporadic REM phases can be observed. During QS all cardio-respiratory activities are lowered in amplitude and dynamic range.

To date, sleep is observed only manually and, more important, sporadic. At present, manual analysis of polysomnographic (PSG) data is still the clinical standard for sleep classification and analysis. Continuous, automated monitoring would enable a variety of possibilities. Some examples would be decision support to the neonatologist, optimizing workflow of the caretakers, completing the patient information scheme and most important to better safeguard the preterm infant´s developmental process. To be able to provide such a monitoring system to a neonatal intensive care unit (NICU) in the near future, no additional sensor should be introduced. Therefore, the system should concentrate on already existing, continuously monitored parameters such as electrocardiography (ECG).

## Related Work

Machine learning opens up the possibility to create an automated algorithm based only on ECG. In the adult sleep research, machine learning has already been successfully used [2], [3]. Radha et al. [4] compared several machine learning methods such as random forest and ensemble support vector machine (SVM) to show promising real-time EEG sleep analysis. Further, full polysomnographic analysis (PSG) [5], [6] and unobtrusive actigraphy methods [7] where investigated for adult sleep state separation.

Sleep analysis for preterm infants is more difficult as the states are less distinguishable. Nevertheless, machine learning for automated EEG [8] analysis, ECG for sleep vs. wake [9] and heart rate variability (HRV) for sleep states analysis [10] have been investigated demonstrating the potential of machine learning for preterm infant sleep state classification.

The next logic step in the automated analysis is the use of artificial neural networks (ANN). The combination of an increase in computation power, the liberalization of computation power due to scalability on demand, an open access to comprehensible analytic tools, and a drastic increase in the amount of data has recently led to the great success of ANNs. The application of ANNs to the topic of sleep classification is on the rise. In 2017 Bishwal et al. presented the annotation tool SLEEPNET [11] using a large dataset to train a deep recurrent neural network (RNN) reaching human level annotation performance. During the same period, Chambon et al. [12] published the implementation of an algorithm which is independent of crafted features using convolution in combination with spatial filtering for classification. A similar approach was chosen by Supratak et al. [13] using an ensemble of convolutional neural networks (CNN) and RNN networks to be able to classify sleep from raw EEG data. At the beginning of 2018 Olesen et al. [14] presented an approach with an adapted model using transfer learning from the ResNet50 architecture. In the following, Sano et al. [15] used long- and short-term memory (LSTM) classifier to identify wake vs. sleep from multimodal data. One of the most recent publications on the topic from Radha et al. [16] used LSTM classifier to classify sleep from HRV features overcoming the temporal limits of non-temporal models. Also, they used transfer-learning to enable the utilization of other signals (here photoplethysmography) with the same trained model enabling different application areas.

As sleep classification in adults is well handled with deep learning, this publication tries to investigate the possibility of deep learning approaches for the more difficult preterm infant sleep classification.

# Methods

## Population

As deep learning demands generally more data than traditional machine learning, three retrospective studies were combined. The dataset recordings have a timespan of several years in-between them. The infants were admitted to the NICU of the neonatal department at the Máxima Medical Center Veldhoven, The Netherlands. Ethical approval was given by the medical ethical committee of the hospital, and written consent was given by the patient’s parents. In those three retrospective studies, 34 (8, 9, 17) stable preterm infants were analyzed during 39 sessions. The preterm infants were born with a mean gestational age (GA) of 29 ± 4.6 weeks. They were studied at a mean postmenstrual age (PMA) of 30 ± 2.4 weeks. The patients had a mean birth weight of 1338 ± 473g.

## Data recordings

Vital signs recordings for all studies were performed with a Philips patient monitor (Intellivue MX 800, Germany) at sampling frequencies of 500 and 250 Hz (n=2). The 250 Hz data were interpolated to meet the 500 Hz.

Each preterm infant was also video-recorded. Videos were either recorded of the face or the total body view. The used cameras were standard, medium resolution, greyscale devices.

## Annotations

Per dataset, two trained observers annotated the data based on 30 s intervals adhering the Prechtl system [17]. The observers used a reference ECG time series and video information for annotation. They annotated the following states: AS, QS, IS, wake, caretaking, and unknown (unable to annotate). The total duration of annotated data was 167h (20021 30s intervals) with a mean duration per patient of 4.28 ± 1.5h (513 ± 179 30s intervals).The overall distribution of state was: AS: 51.45 %, QS: 12.7 %, IS: 16.5%, wake: 6.6%, caretaking: 2.2% and unknown: 10.5%. Subtracting the unknown, a total amount of around 18018 30s intervals were left for analysis. The detailed distribution of all trials can be found in Table 1. After clarifying minor differences in the annotations; the observers reached consent for all datasets.

As preterm infants are mostly awake during caretaking periods, generating very similar signal structures, the labels caretaking and wake were merged under the label caretaking + wake (CTW) to equalize for the low amount of data from each state.

## R-peak detection

The R-peak detection algorithm of Wijshoff et al. [18] was used to determine the NN intervals and the resulting HRV signal. To determine the steepest ascent and descent of the QR and RS slopes they calculated the first derivative of the ECG signal. Then the peaks in the QRS complex were detected with a variable threshold. By interpolation around the detected peaks, they verified that the position of the peak is at the real max. This sub-peak detection assured that there is no shift from the real peak due to off sampling.

## Features

For each dataset 47 features from HRV, respiration, ECG, and patient information were created. The features were calculated based on 30s or 300s intervals. The HRV features include the time, frequency and nonlinear domain. The ECG features were calculated in the time and nonlinear domain, while the respiration features were calculated in the frequency and nonlinear domain. For HRV and EDR the signals are fundamentally non-equidistant in time. The Lomb-Scargle algorithm [19] was used to generate the frequency spectrum as resampling for classic Fourier transformation would have introduced extra parameters.

As the respiratory sinus arrhythmia and cardiorespiratory coupling is not very pronounced in preterm infant and can only be seen in more mature infants [20], [21], the respiration can mostly be determined via superimposed chest movement on the ECG signal. Therefore, the ECG derived respiration was calculated using the ECG envelope. In the frequency domain, the frequency band was limited to max 1,1 Hz (66 breaths per minute ) and min 0.3 Hz (18 breaths per minute) which is described in the literature as the min and max respiration rates of preterm infants [22].

The frequency bands were then separated into high (1.1 - 0.84), medium (0.84 - 0.56), and low (0.56 – 0.3) bands.

From the patient information file gestational age (GA), age at measurement (CA), and birth weight were taken. To gain the timespan between birth and data recording, CA and GA were subtracted from each other. All this information was combined into a stability score. The stability score was subdivided into three different ranges: unstable, medium, and stable. The patient was assigned to a stability score range if at least three parameters being over the threshold of that range.

All features are listed below in Table 1. The normalized features with mean zero and standard derivation of one were combined into 3D tensors which were fed as input into the deep learning models.

## Preprocessing

For the classification of the preterm infant sleep states, the neural network API Keras was used backed with the TensorFlow library from Google Brain. For sleep state classification, time series analysis was used. Therefore the input was cast in the form of a 3D time series tensor [samples, time step, features]. After testing, the time step was chosen as the total length of one recording session with the batch size set to 1. Thereby, long and short-term patterns can be recognized. To achieve uniform length, the tensors where padded to the length of the most extended session. Later, a masking layer and sample weight distribution of zero for the padded values was used to prohibit the padded values to influence the learning process. The data was separated into train and validations sets to exclude significant bias. The split was set to 70% training data 30% validation data. A *3*-fold cross-validation process was used to ensure the proper generalization of the model.

As we have majority (AS) and minority classes (QS, IS, CW) a class weight has to be calculated to balance this unequal class distribution. The sample weight was calculated for the different targets in the training set according to the percentile distribution of classes and normalized to the majority class. Sample weight was used instead of class weight as class weight is converted to sample weights inside of Keras. Using sample\_weight\_mode *temporal*, sample weight fulfills the class weight task and can as well be used for masking-padded-values with a sample weight of 0. So far, this can only be used for smaller datasets as sample\_weight\_mode does not work currently for *fit generator*.

## Model architecture

Four different model types were compared. A wide residual model, a deep residual model, a wide residual model using transfer learning from shallow models, and the more shallow models as standalone architectures.

The base residual architecture itself was adapted from the residual architectures ResNet [23] and ResNext [24]. Both approaches tackle the problem that an increase of model depth creates a sudden and rapid decrease of accuracy which is not caused by overfitting but rather shattered gradients [25]. The shattered gradient let none residual networks appear, with increasing depth, as white noise. Both networks combine multiple shallow models to one larger model. The shallow models thereby fit a residual map which is easier to optimize than a larger model. The connection between the shallow models are ensured with skip layers performing identity mapping and feeding the output of a shallow model block into the next block. This process enables large networks with rather low complexity.

The deep residual network is made of an initiation block followed by five residual blocks of five connected GRU layers (Figure 2). Each connected block ends with a dropout and dense layer. The architecture is finished with a softmax activated dense layer. The initiation block consists of first a masking layer which is needed as the data is padded. The masking layer is followed by a 1/2 dropout layer connected to a dense layer which, both combined, function as a feature selection phase. This combination first randomly reduces the input nodes trailed by reducing the dimensional space which forces the focus on the most distinguishing input information. The last layer of the initiation is a batch normalization to avoid vanishing/exploding gradients by the scale of backpropagated weights.

The wide residual model (Figure 1) uses three parallel paths preluded by an initiation block. The initiation block is of the same structure as in the deep residual model. The parallel paths are each made of blocks from bi-directional Gated Recurrent Unit (GRU) layers. Each GRU layer uses dropout and recurrent dropout to minimize overfitting. As direct regularization of the L2-norm a kernel constrain is used over all axis. Direct kernel constrains work well in combination with dropout [26]. Each path has two of the described blocks which are connected via skip layers in the same way as with the deep model. All parallel pathes are concatenated at the end leading to a dense layer with softmax activation to achieve final state predictions.

For the transfer learning the same wide residual model architecture was used but additional paths were added. Into this additional paths, the loaded weights of the pre-trained models where fixed Figure 3. The pre-trained models are trained on bi-class problems, always training two classes versus each other. In the concatenation step, they are then used for the decision making. To avoid any bias, the bi-models are trained separately only on a fragment of the data which is later not used for further training.

The shallow models, which are also used for pre-training, have an initial masking, dropout and dense layer which is followed by four bidirectional GRU layers. The models are closed with another dropout layer and a Dense, softmax activated layer. The hidden Units for the Dense and GRU layers where set to 32. All other parameter followed the main transfer learning model. The model architecture was compared in performance and speed to the same architecture using bidirectional Long Short-Term Memory (LSTM) layers.

## Model Parameters

The wide and deep residual model, use a range of hidden units for the GRU layers to cover the more simple and the complex information in the input data stream. The wide network uses a different hidden unit for each path ranging from 4 to 128 hidden units. The deep architectur increases the hidden units with each block from 32 to 256. The hidden units of the Dense Layers were set differently to accommodate the previously mentioned feature selection. The values ranged from 16 as roughly 1/3 of the input feature dimension and a power of 2, to a max of 47 representing the full input feature dimension.

To further handle the data imbalance, the earlier mentioned class weights were used in a weighted categorical cross entropy loss function to increase the misclassification gravity for minority misclassification. Therefore, the normalized weights were cast into a matrix representing real versus predicted class with diagonal of one. Each matrix value represents the weight impact for misclassification of each class pear. For each sample, the specific weight is multiplied with the loss of the categorical cross entropy resulting in a weighted loss.

The *Sigmoid* function was selected as the activation function for each residual GRU/LSTM block. In the ResNet, and following residual architectures, the rectified linear unit (*ReLu)* function is used as the activation function. Using the ReLu activation function is not possible with GRU/LSTM as it diverges, but also not necessary as the gating scheme of the GRU/LSTM itself deals with the vanishing gradients. Therefore, the *Sigmoid* activation, which is optimally designed for the GRU/LSTM structure, can be used.

For the optimization algorithm, the Adaptive Moment Estimation (Adam) optimizer was chosen as Adam shows to be generally very effective while also removing the manual setting of the learning rate and learning rate decay. The Adam also was tested against other optimization algorithms where it stood out superior.

As already mentioned, the *timestep* (or lookback) was set to the longest recording session and the batch size to 1.

To effectively fight overfitting, dropout, L1 and, L2 regularizes were applied. The maximum dropout value was 0.6 before the results dropped off out of proportion. Combinations of L1 and L2 regularizations were implemented as kernel- and activity regularizers in different places. L1 was implemented mainly as additional feature selection in the first dense layer with a value between 0.0001 and 0.001. The L2 norm was used mainly on each LSTM/GRU layer. The L2 norm for direct kernel constrain was set to 0.3 on each layer.

# Results

When looking at classification bi-state classification, tri-state classification and all-state classification are of interest. All state classification is more difficult as the training on minority classes is reduced and misclassification towards the majority state is present despite class weighting. Bi-state classification is simpler as for specific state combinations the features show a more distinct differentiation. Tri- state classification difficulty is, due to its nature in-between.

## Bi-state classification

The bi-state classification with a shallow architecture shows promising results for using GRU and LSTM layers. Both show similar mean results (Table 2). Unexpectedly, the use of LSTM layers is faster than the GRU layers with a mean time difference of 94 ± 467min. The performance difference is minimal with 0.13 ± 0.15 between the two layer types (Table 2).

The most robust performance is reached with the majority states AS and QS a mean kappa over the folds ranging from 0.43± 0.07 to 0.40± 0.06 (Figure 4) and between QS and CTW with a mean kappa of 0.44 ± 0.01. Then the combinations AS-IS and IS-CTW show similar results with 0.33 ± 0.03 and 0. 32 ± 0.03. AS-CTW and QS-IS classification have the lowest performance of 0.28 ± 0.005 and 0.25 ± 0.03 (Table 2 and Table 3)

## Tri-state classification

The classification of only three states shows intermediate results (Figure 5) compared with bi- and all state classifications. The majority classes AS and QS were compared with the minority classes IS and CTW. The mean performances in Table 5 indicates that the Majority classes are better differentiable together with IS rather than with CTW. The deep model showed, and the shallow model performed best.

## All state classification

Both wide and deep residual models show similar results for all state classification (Table 6). Same as with the shallow model, the majority states AS and QS are separated best, followed by QS and CTW. The overall performance is lower than for bi-class classification. The use of transfer learning did not improve the performance. In contrary, it showed the lowest overall performance (Table 6)

# Discussion

## Population and Annotations

The total amount of 18018 30s periods is considered a small dataset regarding the vast spectrum of development and health status in this fragile patient group. Especially the health status can vary on a daily bases. Unfortunately, it is challenging to gather data in the high-risk environment of a NICU. The mean age of 29 ± 4.6 weeks GA is relatively young regarding sleep state separation. The sleep stages are much harder to distinguish for early preterm infants, making automated classification difficult. Nevertheless, this challenge offers the opportunity to test the performance in more difficult cases ensuring that the model will be adequate for a general NICU population. The annotation of the dataset was performed by trained annotators but different ones for each set. For this reason the ground truth shows a probably higher disagreement than usual which may limit the performance of an automated system from the very beginning. On the other hand, the trained model incorporates the different experiences and knowledge of different annotators creating a more stable, integral, and reliable model again. Optimal would be a wide range of annotators and annotation styles, backed with sufficient data, to incorporate the derivations of different annotation techniques in the model.

## Features

The feature set was partly used in an earlier publication [10]. It was reused and adapted as it showed a promising explanation of the underlying processes of preterm infant sleep. Patient information was added to the set as it was noticed in previous tests that outlying values of patients can seriously influence the performance of classification. Such values are in general not random but often occur at very young, immature preterm infants and/or with low birth weight. Also, it makes a huge difference in the development at which age a preterm infant was born and at which timespan after birth the data were recorded. If the measurement takes place at the same time with different GA at birth (or vice versa), the development state and consequently the feature appearance can look sufficiently different to influence the learning. The same hold for the birthweight. A heavier baby tends to be more stable. As the values of age, age difference, and weight are almost continuous data, they were categorized into a stability score between 1 and 3. With a significantly larger dataset either the values can be used directly, or a finer grid can be used to categorize the preterm infants. In previous tests, it was noticed that the use of respiration devices influence the classification performance. Unfortunately, it was not possible to gather this information for all patients.

## Model Parameters

To handle overfitting, settings were experimented with dropout, recurrent dropout, L1 and, L2 norm for activity (Ar) and kernel (Kr) regulation. Dropout and activity regularization had the most effect on overfitting. Different combinations of this regularizations were useful in reducing overfitting. Here only two are mentioned as examples. Either using overall lover dropout (e.g. 0.3) in combination with a stiffer Kr and Ar L2 norm (e.g. 0.01) without any other regularization in the initiation block helped fighting overfitting. Another variant is to use an L1 norm as kernel regularization in the first dense Layer to help with feature selection in the initiation block in combination with an overall dropout/recurrent dropout of 0.5-0.6 but no other Kr or Ar in the following layers. Choosing too high values for the dropout and /or regularization would lead to a drastic reduction in overall performance on the validation data. There were plenty of combinations which all resulted in reducing overfitting. Nevertheless, further investigation and proper comparison will go beyond the scope of this publication as overfitting was not the primary problem in this analysis.

The lookback was chosen as the total duration of one session for the LSTM/GRU as long-term sleep cycles can influence the overall learning process. As LSTMs/ GRUs can only learn the variations in time on the information of one batch, long-term patterns such as total sleep cycles or specific sleeping patterns need at least 30 min of data. In regular cases, sleep states changes follow the pattern wake-AS-QS-AS-wake with IS patterns in between. Irregular patterns are for example wake-QS-(AS)-wake. This pattern is called a stress sleep pattern showing signs of the preterm infant’s immediate need for rest. With more of such recorded patterns outside the norm, future research could try to detect anomalies in sleep cycle patterns to inform the responsible caretakers. Either regular or irregular cycle patterns cannot be learned with batches of insufficient length.

## Results

### Classification

The classification between the majority classes AS and QS show high performance and generally promising results. This is a very positive result as bi-state classification is crucial for clinical aid. As the state distribution of AS and QS is one of the primary indicators for neuronal development in early preterm infants, the bi-state classification for AS and QS can be utilized for neural development indication. As the minority states naturally occur less, they are of less importance to the course of development in the early stages of preterm infancy. In term infants, wake versus sleep becomes more important, but at that point, wake also has a more significant presence which can be utilized for training.

The lower classification of IS - CTW have to be explained with the minimal amount of data for CTW compared to AS and QS as preterm infants are almost not awake and caretaking takes only a portion of the day. Interestingly, QS is underrepresented among the three datasets but show enough difference to AS to be sufficiently distinguishable. Generally, CTW shows differences in the patterns to the QS and IS states resulting in heightened performance for QS - CTW and IS - CTW classification despite the lack of data. The activity in both AS and CTW, and thereby signal similarity, makes it harder to classify resulting in the lowest performance. Another influence could be wrong annotations, as during CTW the preterm infant moves similarly to AS. If the eyes are not open or caretaking cannot be directly observed in the video frame, CTW could be mistaken for AS. AS - IS are better separable than QS – IS which could be due to the reduced breathing and movement during IS. This reduction result in similar patterns for IS and QS, making the correct classification more difficult. Same as before, another reason could be the manual annotation. As changes to the heart rate variability indicating a state change without visible clues like twitches, eye movements or rapidly changing breathing, IS could be easily mistaken for the onset or continuation of QS.

The tri-state classification is expected to show slightly lower performance than the bi-state counterparts. Here, the more difficult states, IS and CTW, reduce the combined performance. The slightly higher performance between AS - QS - IS despite IS being a more difficult state to differentiate has to be explained with a higher amount of training data. Despite the lack of data, AS – QS - CTW classification shows only slightly reduced performance as noise, instability, and increased movement dominate the ECG patterns and creating a more clear differentiation.

The performances on all state classifications using the residual approach are underwhelming. Nevertheless, the use of a simpler model also did not generate reasonable results. Here again, the data to train on, especially for the minority classes, was considerable small with very early, unstable, and fragile patients. However, due to higher performance on the majority classes AS and QS, general different feature modalities between the single sates, and the very high difference in the amount of data between majority and minority classes shows that the problematic performance is directly linked to the data amount and not fundamental problems with the used model architectures. This generally indicates the correct track utilizing deep learning for preterm infant sleep classification and displays a bright prospect for an increased all state classification performance as more data becomes available. This prospect is justified as deep learning has mostly a higher performance potential than machine learning with increasing data size. Generally, all-state classification is not of main importance for early preterm infant development monitoring but is vital for a holistic view on the patient's sleep rhythm and possible predictions of sleep patterns.

To summarize, the separation of AS and QS show the general potential with using deep learning for sleep classification only using ECG derived features. Nevertheless, for a complete picture and overall sleep monitoring a wider study is necessary to gain a stable model including training on extreme outliers.

### Model architectures

Recently, GRU networks were found to have similar performance as LSTM networks. The GRU uses less computational power as it generates fewer parameters. Nevertheless, both units perform almost equally, and one cannot be generally favored over the other. We tested architectures with both units and found that in our case both, LSTMs and GRUs layer use, performed equally (see Table 2). The surprising lower LSTMs calculation time could maybe be explained with changing compute server workload during the different runs.

The wide and deep residual model architectures show similar results Table 3. The total amount of layers after the initiation block is similar with ten layers in the deep model and 12 for the wide model. Both architectures consider low and high complexity relations between the features and sleep states with increasing hidden units. In the ResNext model approach, the idea was also to introduce cardinality, an increase of parallel structures per residual block. At this point, we only used a cardinality of one as the model architecture could not be enhanced further due to overfitting spiraling out of hand. A deep model with 25 GRU layers after the initiation block was run with massive overfitting problems. Compared to a model with 4 GRU layers after the initiation block, the residual structured models showed weaker performance (Table 2). The overall disappointing results of the residual architectures show that probably they overreach with the complexity of the analysis on that task at hand. The complexity cannot be put to use as too few training examples for the more complex feature conjunctions are at hand. Additionally, the increased complexity tends to lead to overfitting due training onto complex appearing noise structures. Regularization and dropout have to be set in place which can lead to a performance restriction. A solution to finding the right model architecture might be an evolutionary approach for architecture search.

Unfortunately, transfer learning did not work. Probably this due to the even more reduced data size as the data pool for pre-training had to be separate from the main pool for bias control. Also, the saved weights from the pre-trained models were taken from a single fold. Even though they showed good results (Table 4), they lacked generalization on the validation data.

# Suggestions

As the main reason for the low performance of the all-state classification can be linked with the low amount of data, considering the vast difference in preterm infant stability and development, we suggest that more preterm infant data has to be gathered to surmount the threshold where data size becomes not the primary influence on performance.

Even though the transfer learning approach did not show the intended results, we suggest that term infant data instead of rare preterm infant data is used for pre-training as signal patterns and sleep architecture are still very similar to preterm infants and much more data is available for this patient group.

Another approach could be to look at unsupervised learning for preterm infant sleep staging. So far, we rely on human annotations which are in itself not perfect and show tremendous interrater variability. The general shift of data patterns from unsupervised learning could indicate brain development in the same way as classified state distributions from supervised learning. Unsupervised learning would demand even more data but will reduce the necessity of manual annotation. Not annotated, preterm infant sleep data is already freely available for example from the CHIEME study [27].

# Conclusion

Active and quiet sleep can be separated using a deep learning approach solely using ECG derived features. Nevertheless, all state classification is, so far, not possible and is hindered mostly by limited preterm infant training data as well as training data of very young and unstable patients. There is a level of data which has to be reached so that the data amount is not the significant factor for performance.

# Literature

[1] J. Werth, “Unobtrusive sleep state measurements in preterm infants – A review,” *Sleep Med. Rev.*, vol. 32, pp. 109–122, Apr. 2017.

[2] P. Fonseca, “Sleep stage classification with ECG and respiratory effort,” *Physiol. Meas.*, vol. 36, no. 10, pp. 2027–2040, Oct. 2015.

[3] P. Fonseca, “Cardiorespiratory Sleep Stage Detection Using Conditional Random Fields,” *IEEE J. Biomed. Heal. Informatics*, vol. 21, no. 4, pp. 956–966, Jul. 2017.

[4] M. Radha, “Comparison of feature and classifier algorithms for online automatic sleep staging based on a single EEG signal,” in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014, pp. 1876–1880.

[5] X. Long, “Detection of nocturnal slow wave sleep based on cardiorespiratory activity in healthy adults,” *IEEE J. Biomed. Heal. Informatics*, pp. 1–1, 2015.

[6] A. Procházka, “Multi-Class Sleep Stage Analysis and Adaptive Pattern Recognition,” *Appl. Sci.*, vol. 8, no. 5, p. 697, May 2018.

[7] X. Long, “Actigraphy-based sleep/wake detection for insomniacs,” in *2017 IEEE 14th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, 2017, pp. 1–4.

[8] N. Koolen, “Automated classification of neonatal sleep states using EEG,” *Clin. Neurophysiol.*, vol. 128, no. 6, pp. 1100–1108, Jun. 2017.

[9] A. Lewicke, “Sleep versus wake classification from heart rate variability using computational intelligence: consideration of rejection in classification models,” *IEEE Trans. Biomed. Eng.*, vol. 55, no. 1, pp. 108–118, 2008.

[10] J. Werth, “Unobtrusive assessment of neonatal sleep state based on heart rate variability retrieved from electrocardiography used for regular patient monitoring,” *Early Hum. Dev.*, vol. 113, pp. 104–113, Oct. 2017.

[11] S. Biswal, “SLEEPNET: Automated Sleep Staging System via Deep Learning,” pp. 1–17, Jul. 2017.

[12] S. Chambon, “A Deep Learning Architecture for Temporal Sleep Stage Classification Using Multivariate and Multimodal Time Series,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 4, pp. 758–769, Apr. 2018.

[13] A. Supratak, “DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, 2017.

[14] A. N. Olesen, “End-to-End Deep Learning Model For Automatic Sleep Staging Using Raw PSG Waveforms,” *Sleep*, vol. 41, no. 1, p. A121, 2018.

[15] C. W., “Multimodal ambulatory sleep detection using recurrent neural networks,” *Sleep*, 2017.

[16] M. Radha, “LSTM knowledge transfer for HRV-based sleep staging,” pp. 1–11.

[17] H. F. R. Prechtl, “The behavioral states of the newborn infant (a review),” *Brain Res.*, vol. 76, no. September 1973, pp. 185–212, 1974.

[18] R. W. C. G. R. Wijshoff, “Reduction of periodic motion artifacts in photoplethysmography,” *IEEE Trans. Biomed. Eng.*, vol. 64, no. 1, pp. 196–207, 2017.

[19] T. Ruf, “The Lomb-Scargle periodogram in biological rhythm research: analysis of incomplete and unequally spaced time-series,” *Biol. Rhythm Res.*, vol. 30, no. 2, pp. 178–201, Apr. 1999.

[20] P. Indic, “Assessment of cardio-respiratory interactions in preterm infants by bivariate autoregressive modeling and surrogate data analysis,” *Early Hum. Dev.*, vol. 87, no. 7, pp. 477–487, 2011.

[21] S. Reulecke, “Autonomic regulation during quiet and active sleep states in very preterm neonates,” *Front. Physiol.*, vol. 3, no. April, pp. 1–9, 2012.

[22] K. Cross, “The respiratory rate and volume in the premature infant.,” *J. Physiol.*, vol. 116, no. 2, pp. 168–74, Feb. 1952.

[23] K. He, “Deep Residual Learning for Image Recognition,” *arXiv Prepr. arXiv1512.03385*, 2015.

[24] S. Xie, “Aggregated residual transformations for deep neural networks,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017–Janua, pp. 5987–5995, 2017.

[25] D. Balduzzi, “The Shattered Gradients Problem: If resnets are the answer, then what is the question?,” 2017.

[26] N. Srivastava, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[27] “Collaborative Home Infant Monitoring Evaluation,” 1998. [Online]. Available: http://slone-web2.bu.edu/ChimeNisp/.

# Figures

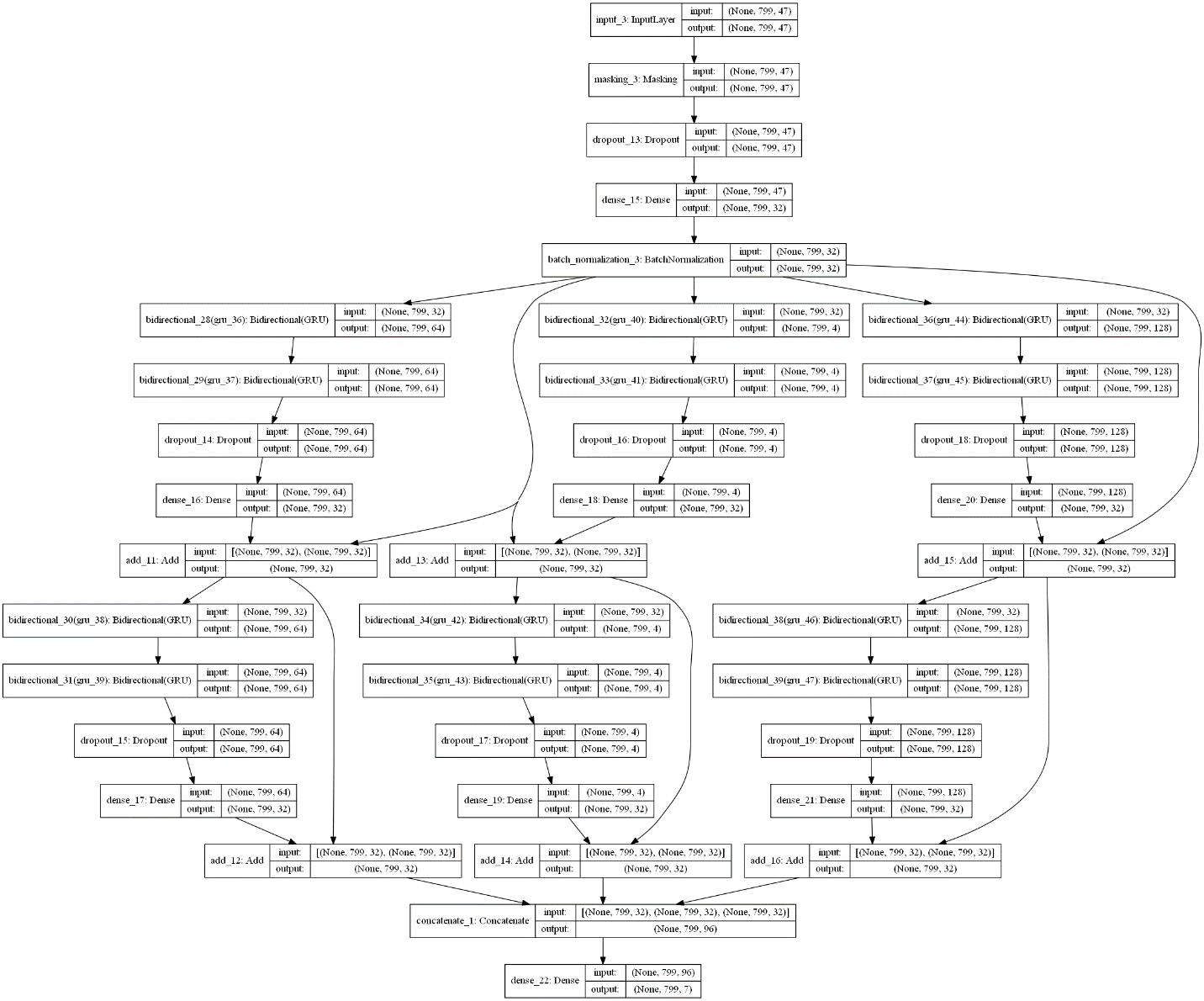


Figure 1 Exemplary Wide residual model

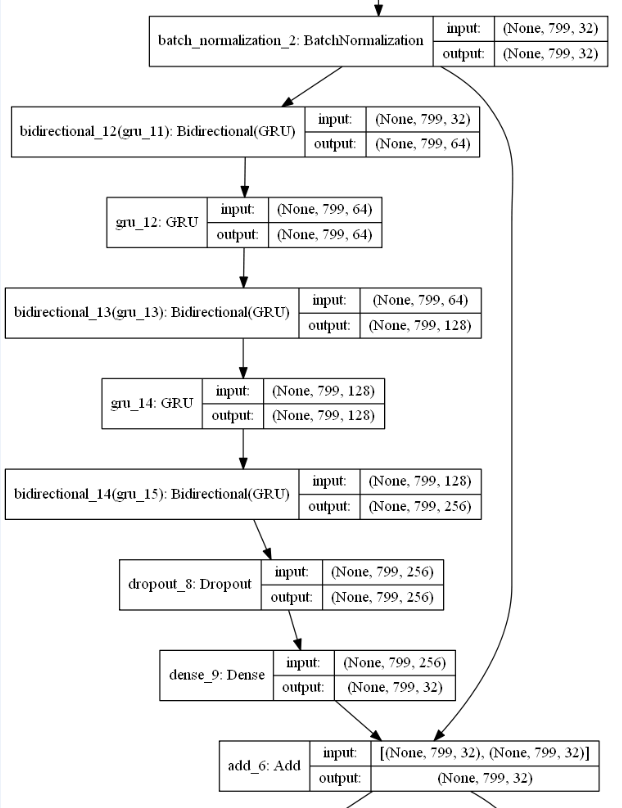


Figure 2 Residual block of deep model

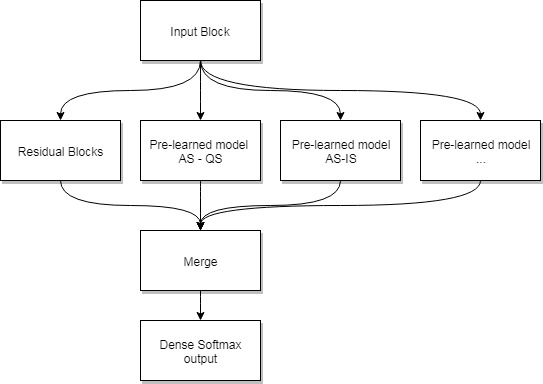


Figure 3 Architecture of the residual model utilizing transfer earning

Table 1 State distribution per dataset

|  |  |  |  |
| --- | --- | --- | --- |
| States | Dataset 1 | Dataset 2 | Dataset 3 |
| Unknown | 1,6 | 37,9 | 0,0 |
| AS | 65,4 | 47,7 | 46,0 |
| QS | 7,5 | 6,7 | 18,9 |
| Wake | 2,0 | 1,6 | 11,9 |
| CT | 8,4 | 0,0 | 0,0 |
| IS | 15,1 | 6,0 | 23,3 |

Table 2 Overview of the used ECG and HRV features for classification

|  |  |  |
| --- | --- | --- |
| NR | Feature [unit] | Description |
| 0 | BpE | Beats per Epoch / mean Beats per Epoch |
| 1,2 | LL, aLL [mV] | Line Length / mean Line Length |
| 3-6 | NNx [count] | The number of pairs of successive R-R intervals that differ by more than 10, 20, 30 or 50 ms of a defined window length. |
| 7-10 | pNNx [%] | The proportion of NNx divided by total number of R-R intervals of a defined window length. |
| 11 | RMSSD [ms] | Root mean square of successive differences between adjacent R-R intervals of a defined window length. |
| 12 | SDALL [mV] | Standard derivation of averaged line length |
| 13 | SDANN [ms] | Standard Deviation of averaged NN intervals |
| 14 | SDLL [ms] | Standard derivation of line length |
| 15 | SDNN [ms] | The standard deviation of normal to normal R-R intervals of a defined window length. |
| 16 | HF [ms2] | The power of the high frequency band between 0.15-0.4 Hz of a defined window size. |
| 17 | HFnorm [%] | HF power in normalized units HF/(Total Power-VLF) x 100 |
| 18 | LF [ms2] | The power of the low frequency band between 0.04-0.15 Hz of a defined window size. |
| 19 | LFnorm [%] | LF power in normalized units LF/(Total Power-VLF) x 100 |
| 20 | LF/HF [n.u.] | Ratio LF/HF |
| 21 | pHF1 [ms2] | The power of the high frequency band between 0.4-0.7 Hz |
| 22 | pHF1norm [%] | pHF1 power in normalized units pHF1/(Total Power-VLF) x 100 |
| 23 | TotPow [ms2] | Total power or variance of NN intervals of a defined window size. |
| 24 | pHF2 [ms2] | The power of the high frequency band between 0.7-1.5 Hz |
| 25 | pHF2norm [%] | pHF2 power in normalized units pHF2/(Total Power-VLF) x 100 |
| 26 | VLF [ms2] | The power of the very low frequency band between 0.003-0.04 Hz of a defined window size. |
| 27,28 | SE, QSE [n.u.] | Sample entropy / Quadratic sample entropy |
| 29 | SEAUC [n.u.] | Sample entropy area under the curve |
| 30 | pDEC [%] | The percentage of HR decelerations |
| 31 | SDDec [ms] | Magnitude of HR deceleration |
| 32,33 | LZNN [n.u.],  LZECG [n.u.] | Lempel-Ziv complexity measure on HRV and ECG |
| 34 | HF\_R | The power of the high frequency band of the respiration signal between 0.48-1.1 Hz of a defined window size. |
| 35 | HFnorm\_R | HF respiration power in normalized units. HF/(TotPow\_R-LF\_R) x 100 of the respiration. |
| 36 | MF\_R | The power of the medium frequency band of the respiration signal between 0.56-0.84 Hz of a defined window size. |
| 37 | MFnorm\_R | MF power in normalized units of the respiration. MLF\_R/(TotPow\_R-LF\_R) x 100 |
| 38 | LF\_R | The power of the low frequency band of the respiration signal between 0.56-0.3 Hz of a defined window size. |
| 39 | LFnorm\_R | LF power in normalized units of the respiration. LF\_R/(TotPow\_R) x 100. |
| 40 | LF\_R/HF\_R | Ratio between low and high respiration spectrum. LF\_R/HF\_R |
| 41 | MF\_R/HF\_R | Ratio between medium and high respiration spectrum. MF\_R/HF\_R |
| 42 | TotPow\_R | Total power of the respiration frequency spectrum. |
| 43 | Age difference | Difference between age at birth and age at measurement. |
| 44 | Birthweight | Weight at time of birth |
| 45 | GA | Gestational age. Age at birth calculated from the last gestation. |
| 46 | CA | Conceptional age. Age at time of measurement |

# Result Tables

Table 3 Mean performance for bi state classification using shallow model architectures

|  |  |  |
| --- | --- | --- |
| State pairs | LSTM (κ ±std) | GRU (κ ±std) |
| AS-QS | 0.43 ± 0.08 | 0.43 ± 0.08 |
| AS-IS | 0.33 ± 0.02 | 0.33 ± 0.03 |
| AS-CTW | 0.26 ± 0.003 | 0.25 ± 0.03 |
| QS-IS | 0.29 ± 0.01 | 0.28 ± 0.005 |
| QS-CTW | 0.50 ± 0.02 | 0.44 ± 0.01 |
| IS-CTW | 0.34 ± 0.03 | 0.32 ± 0.03 |

Table 4 Mean performance for bi state classification using residual model architectures

|  |  |  |
| --- | --- | --- |
| State pairs | Residual Wide | Residual deep |
| AS-QS | 0.37 ± 0.07 | 0.38 ± 0.04 |
| AS-IS | 0.31 ± 0.03 | 0.30 ±0.03 |
| AS-CTW | 0.26 ± 0.005 | 0.25 ±0.01 |
| QS-IS | 0.28 ± 0.03 | 0.27 ±0.007 |
| QS-CTW | 0.39 ± 0.005 | 0.40 ±0.001 |
| IS-CTW | 0.30 ± 0.04 | 0.29 ±0.03 |

Table 5 Kappa performance of shallow model architectures used for transfer learning

|  |  |
| --- | --- |
| State pairs | Kappa |
| AS-QS | 0.51 |
| AS-IS | 0.36 |
| AS-CTW | 0.27 |
| QS-IS | 0.29 |
| QS-CTW | 0.55 |
| IS-CTW | 0.38 |

Table 6 Mean kappa performance of different models for three state analysis

|  |  |
| --- | --- |
| State pairs | Kappa |
| AS-QS-IS | 0.36 ± 0.04 |
| AS-QS-CTW | 0.32 ± 0.09 |

Table 7 Kappa performance on all state classification for different models

|  |  |
| --- | --- |
| Model | Kappa |
| Wide Residual | 0.30 |
| Deep Residual | 0.28 |
| Shallow | 0.25 |
| Wide Residual  using Transfer learning | 0.13 |

# Result Figures

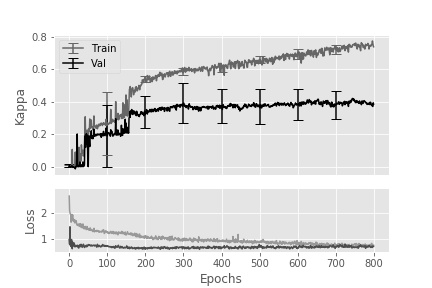


Figure 4 Mean kappa and loss of active versus quiet sleep classification over epochs

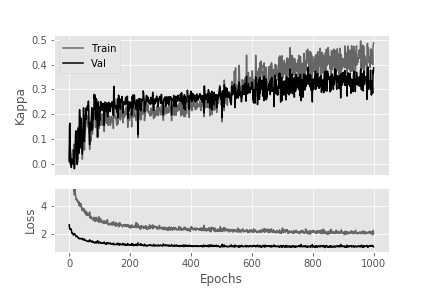


Figure 5 Kappa and loss of active vs quiet vs intermediate sleep classification over epochs.

# Captions

**Caption** Figure 1

Wide residual model with Initiation block of masking layer, dropout layer, and following dense layer. Afterward, the architecture is split into three paths, where each path consists of connected bi-directional gated recurrent unit layers which are later concatenated again. The layers are connected with skip connections to help simply fining the network optimization. Each path uses different hidden units to incorporate more and less complex relations.

**Caption** Figure 2

An exemplary block from the deep residual model. Here gated recurrent units and bidirectional gated recurrent unit layers alternate each other with increasing hidden units (neurons). Thereby, the hidden units increase from 32 to 256 covering simpler to more complex feature-state connections

**Caption** Figure 3

The input block and the Residual block are the same as in Figure 1. Also, pre-learned blocks of shallow architectures are added parallel and fixed until a various point. All weights are concatenated and added into a dense layer with softmax activation.

**Caption** Figure 4

Kappa and weighted categorical cross entropy loss over epochs using a shallow architecture. Initial dropout is 0.2, and dropout/recurrent dropout per bidirectional gated recurrent unit layer is 0.5. Kernel constrain per layer is set with max norm 0.3. Hidden units of 32 is used per layer.

**Caption** Figure 5

Kappa and weighted categorical cross entropy loss over epochs using a shallow architecture. The model used a kernel L2 regularization of 0.01 and an activity L2 regularization of 0.001. Initial dropout is 0.2, and dropout/recurrent dropout per bidirectional gated recurrent units layer is 0.5. Kernel constrain per layer is set with max norm 0.3. Hidden units of 32 is used per layer.

**Caption** Table 3

The categorical accuracy of bi-state classifications using a sigmoid activated model with four long short-term memory or gated recurrent units bi-directional layers, Adam optimizer, and unweighted categorical cross entropy loss function. The Kernel constrain was set to 3. 50 % Initial and Recurrent dropout was used with both 0.5. Initial and Recurrent dropout was used. L2 activity and kernel regularization was set to 0.01 and 0.001 per layer.

**Caption** Table 4

Kappa and weighted categorical cross entropy loss over epochs using a residual architectures. The model used a kernel L2 regularization of 0.01 and an activity L2 regularization of 0.001. Initial dropout is 0.2, and dropout/recurrent dropout per bidirectional gated recurrent units layer is 0.5. Kernel constrain per layer is set with max norm 0.3. Hidden units of 32 is used per layer.

**Caption** Table 5

Kappa results without cross validation using the same parameters as for Table 3 utilizing bidirectional gated recurrent unit layers.

**Caption** Table 6

Mean performance of tri-state classification using a sigmoid activated model with four bi-directional gated recurrent units layers, Adam optimizer, weighted categorical cross entropy loss function. The Kernel constrain was set to 3. 50 %. Initial and Recurrent dropout was used with 0.6 and 0.5. L2 kernel regularization per layer was set to 0.01.

**Caption** Table 7

Mean kappa performance over different models using the same parameters as described in the other tables.