FINAL REPORT: SLEEP STATES DETECTION (SLEEPSENSE)

Yingge Hu
Student# 1001541165
yingge.hu@mail.utoronto.ca

Joshua Chau
Student# 1008344720
joshuatt.chau@mail.utoronto.ca

Zhuoran Wang
Student# 1007646674
zr.wang@mail.utoronto.ca

Nathan Hung
Student# 1008382180
nathan.hung@mail.utoronto.ca

ABSTRACT

This report introduces SleepSense and outlines the project's progress. It provides an overview of the project's background and concept, detailing efforts to develop a neural network model that predicts sleep patterns with computer vision. The report provides evaluation of the model in quantitative and qualitative ways and compared it to baseline models. Further discussions on experiences gained and ethical considerations are also included.
<Total Pages: 9>

1 INTRODUCTION AND MOTIVATION

Sleep is fundamental to human health: it influences mood, health, and cognition. Precise sleep monitoring can unravel insights into individual mood and behavior. Many existing models can accurately analyze sleep data, but most rely on brain scans or EEG data, which requires costly medical device.

Our project, SleepSense, aims to automate sleep monitoring with only accelerometer data from smart wristbands, detecting sleep phase onset[1] and wake up times. This allows sleep monitoring with just a low-cost wristband, making sleep monitoring more accessible and offering a low-cost diagnostic tool for health professionals. Figure 13 shows a straight-forward example where there is a clear distinction between sleep and awake periods, marked by high and low activity. However, most samples are more complicated with periods of movements during sleep and wake periods of little movement. This complexity and the nonlinear relationship between the variables drives our decision of using deep learning.

The accelerometer data contains two important features: ”enmo” and ”anglez.” They represents the acceleration magnitude and z-axis rotation. Our model uses them to predict.

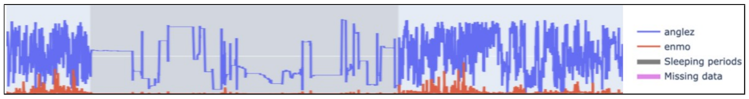


Figure 1: An straight-forward example of sleep window (Fu)

2 BACKGROUND AND RELATED WORK

Several achievements were made using heuristic-based algorithms and neural network in sleep monitoring.

[1] Onset means the beginning of sleep (time of falling asleep, the opposite of wake up time)

Heuristic-based solutions, commonly utilized in smartwatch software, uses features like rotation to identify sleep windows (Van Hees et al., 2018) (Van Hees et al., 2015).

2.1 Empirical Feature Selection Based Solutions

Some researchers have explored empirical feature selection with basic machine learning models. For example, the study by Hassan & Subasi (2017) applies signal decomposition techniques and random forest algorithms to data, yielding promising results across various datasets. Similarly, Lajnef et al. (2015) employs a multi-class support vector machine (SVM) classification, leveraging features validated by previous studies like permutation entropy (Olofsen et al., 2008).

2.2 Neural Network-based Solutions

On the neural network front, approaches vary. Phan et al. (2019) views the problem as sequence-to-sequence classification, using a combination of filter bank layers, attention-based recurrent layers for short-term modelling, and recurrent layers for long-term epoch sequencing. Contrastingly, Chambon et al. (2019) draws inspiration from object detection in computer vision. Their DOSED model uses convolutional neural networks for feature representation from raw EEG signals, focusing on predicting and selecting the most probable non-overlapping sleep events.

Blending these ideas, Supratak et al. (2017) introduces a hybrid approach utilizing Convolutional Neural Networks (CNNs) and bidirectional-Long Short-Term Memory (LSTM) networks. This method enables the learning of transition rules among sleep stages directly from EEG epochs without relying on hand-engineered features, with CNNs extracting time-invariant features and bidirectional-LSTMs encoding temporal information.

3 Data Processing

We chose the dataset from the Child Mind Institute Healthy Brain Network as it is the largest publicly available dataset for wrist-worn accelerometer-annotated sleep data, aligning with the demands of deep learning for substantial training data. This dataset contains 500 sleep recordings with wrist-worn accelerometer time series, each annotated as ”onset(beginning of sleep)” and ”wake up(end of sleep)”. Although it offers a generous sample size, it necessitates cleaning and organization and exhibits storage.

3.1 Data cleaning and organization

To ensure data quality and reliability, we addressed inconsistencies, errors, and anomalies in the raw dataset. We dropped all entries with missing label significant values. Additionally, we've observed extended periods of inactivity in the accelerometer data, characterized by consecutive zero readings (Figure 2)(Figure 3), likely because the user removed their wrist-worn accelerometer. We excluded these recordings from the training data.

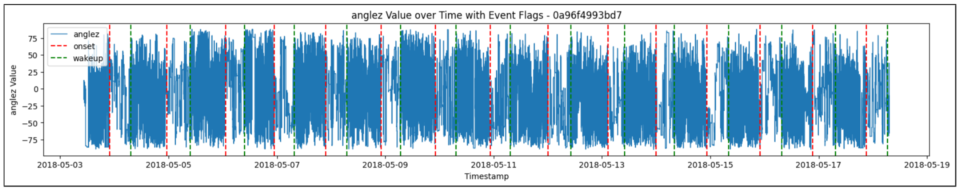


Figure 2: Normal accelerometer data.

3.2 Memory optimization

Despite containing only 3801 accelerometer recordings with floating point values, the dataset occupies a substantial 950MB, posing computational challenges. We've applied storage optimization. For instance, the 'night' variable currently uses int64 but only spans from 0 to 34. Therefore, it

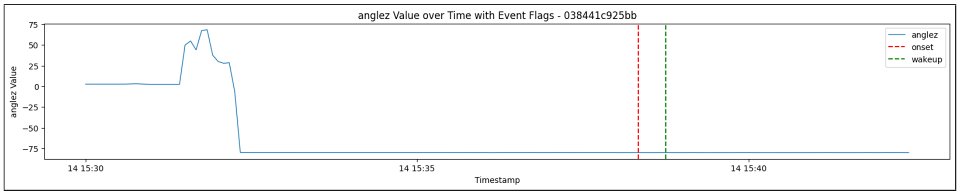
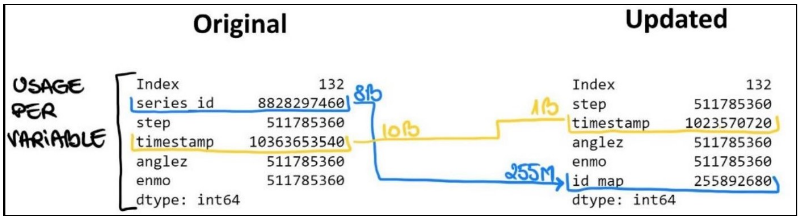


Figure 3: Abnormal accelerometer data with consecutive zero readings.

can safely be converted to uint16. Similarly, the ’event’ which can only have 3 values should use uint8 instead of a Python object. These memory reductions collectively cut dataset memory usage by 73% (See Figure 4).



```
Index                132
series_id     8828297460
step           511785360
timestamp    10363653540
anglez         511785360
enmo           511785360
dtype: int64
```

```
Index                132
step           511785360
timestamp     1023570720
anglez         511785360
enmo           511785360
id_map         255892680
dtype: int64
```

Figure 4: Summary of memory reduction techniques.

4 Quantitative measurement methodology

We use three metrics to measure performances of our primary and baseline models:

RMSE with outliers removed: RMSE have the same units as our output variable (time), making it easy to interpret the magnitude of the error by converting to minutes. We chose RMSE over Mean Average Error for its sensitivity to large errors, which is more important large errors drastically hinders usefulness in medical applications. A 20-minute error is significantly more severe than a 5-minute error, surpassing it by more than fourfold. However, we remove errors larger than 8 hours, because impossibility huge errors are much less of an concern than subtle errors. When using predicted sleep state data, medical professionals are trained to identify and rectify such errors during diagnosis, thereby avoiding blind assessments of sleep irregularities.

R-squared: R2 provides a scale-independent measure of our model’s predictive power. It is widely used in similar researches (Vincent Theodoor van Hees, 2018), allowing us to easily compare our models with other models with outputs of different units and scales.

EDAP (event detection average precision) score: this is a custom metric that is specifically used by the kag when evaluating sleep state regression models. Error tolerance varies in different sleep state monitoring use cases. Tolerance are usually between 1 to 30 minutes, as errors beyond this makes the prediction useless. The EDAP score uses common required tolerances [2], calculating the precision [3] using each tolerance, and averaging the result. This robust metric accounts for potential variations in timestamp accuracy, and the different required tolerance in different applications.

5 Primary model Architecture

RNN related models are usually best for time series predictions. However, after hitting an accuracy plateau with our GRU and LSTM model, we discovered a brand new approach.

[2] 1, 3, 5, 7.5, 10, 12.5, 15, 20, 25, 30 minutes (kag)

[3] number of predictions that falls within the tolerance ÷ total number of predictions

5.1 MOTIVATION FOR USING GRAPHICAL-BASED APPROACH

How does the ”medical professionals” label the data? They look at graphed data with their eyes. Therefore, is it possible that a computer vision model can solve this problem?

When conducting literature review, we noticed how S. Chambon & Arnal (2019) introduced their CNN model DOSED that operated on EEG signal. While our signal is less complex than EEG signals, we hypothesized that a similar ”object detection” method will produce a satisfying result.

5.2 PROCESS OF CONVERTING TO GRAPHS

When converting data into graph images, we assumed that there is one wakeup and one onset per day (24 hour period). This allows us to separate the time series into 24 hours windows. While not necessarily true (as we will show later), it is true for most cases. We color the two features using red and green. An example is shown in Figure 5

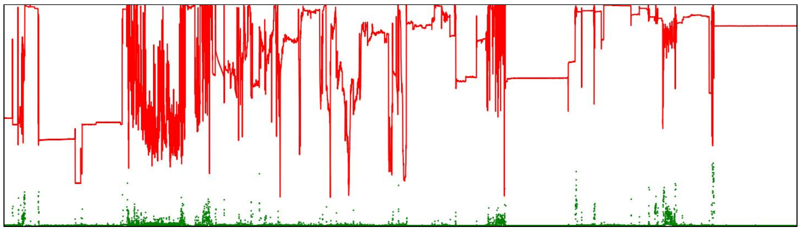


Figure 5: An example of generated image input for RCNN

In order to use the RCNN model (explained in next section), we also turn the ground truth onset and wakeup times into target boxes with width of 30 minutes.

5.3 RCNN

Faster RCNN model (Region Based Convolutional Neural Networks) is a object detection and segmentation model with great performance without needing too much computatio. Faster R-CNN is efficient in handling large input sizes and integrating region proposal networks, thus the ideal choice for our input of large time series data. We tailored it to detect sleep states from the images generated from accelerometer data.

5.3.1 ARCHITECTURE OVERVIEW

We utilized ResNet-50 with Feature Pyramid Network (FPN) for feature extraction. Then, the RPN(Region Proposal Network) generated region proposals which are classified and refined by the subsequent layers. It focuses the model's attention on areas containing the target sleep states. The Region of Interest (RoI) pooling layer extracted fixed-size feature maps from these proposals to ensure uniformity before classification. The final layers of our model comprised a classifier and a bounding box regressor. The classifier determined whether a given region contained a sleep or wake state, while the regressor tuned the coordinates of the bounding box to enclose the detected state.

A graphical overview of the architecture is given in figure 6

5.3.2 TRAINING

We used Stochastic Gradient Descent (SGD) with a learning rate of 0.005, momentum of 0.9, and weight decay of 0.0005. StepLR scheduler was employed to reduce the learning rate by a factor of 0.1 every 3 epochs.

Loss Components: We monitored four types of losses - loss for classification, loss for bounding box regression, loss for objectness, and loss for RPN box regression. The final performance was evaluated using Average Precision(AP) and Average Recall(AR) across various Intersection over Union(IoU) thresholds.

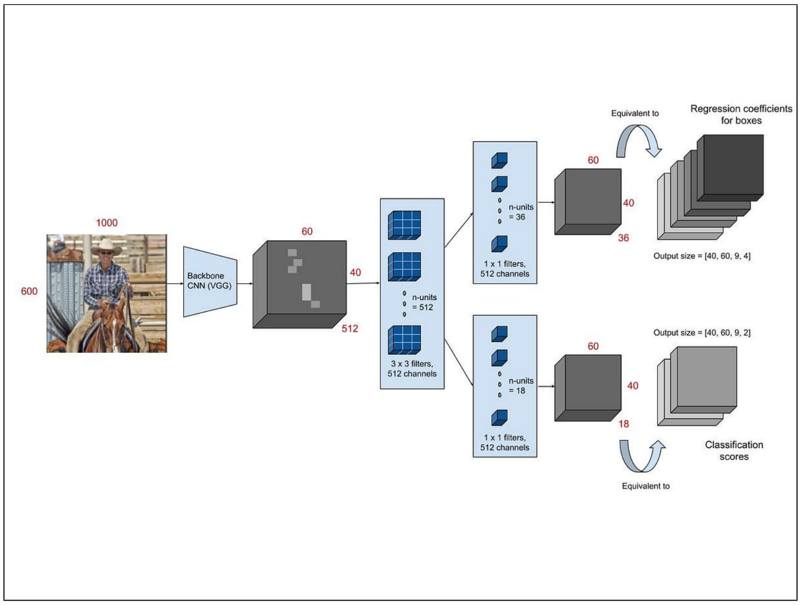

1000
600
Backbone CNN (VGG)
60
40
512
n-units = 512
3 x 3 filters, 512 channels
n-units = 36
1 x 1 filters, 512 channels
60
40
36
Equivalent to
Regression coefficients for boxes
Output size = [40, 60, 9, 4]
Output size = [40, 60, 9, 2]
n-units = 18
1 x 1 filters, 512 channels
60
40
18
Classification scores
Equivalent to

Figure 6: An overview of faster RCNN

6 Primary Model Result

6.1 Quantitative result

Our final model performed extremely well compared to the baselines (Table 3). Our model was able to accurately predict both onset and wakeup time, but had slightly larger and more varied error for onset. Moreover, the distribution is bimodal, with the model sometimes underpredicting by around 80 minutes. This is because sleep onset time has larger variability in dataset, as most adults wake up at similar time in the morning for work, but sleep at vastly different times. Moreover, there are often movement activities right after sleep onset due to shallow sleep, causing our model to misclassify. Wakeup time predictions have slightly higher correlation. Both have similar tolerance vs precision plot. Overall, we achieved a moderate error of 51 minutes, and a high EDAP precision score of 0.79. Using tolerance threshold of 10 minutes, 86.4% of the predictions are correct. At 20 minutes tolerance, the precision is 91.0%. This demonstrated great usability for usecases with large tolerances (discussed later).

RMSE (minutes)	R2	EDAP Score
51.38908	0.90721	0.79034

Table 1: Mean Predictor quantitative metrics

6.2 QUALITATIVE RESULTS

In our model design, we assumed that there is only one onset and wakeup per day, which is true for almost all days. For those samples, our model was able to correctly predict most samples with very high precision (Figure 7). This is a randomly drawn example of correct prediction, to ensure we do not cherry pick.

In the rare occasion of samples with only a wakeup time but no onset time, our model correctly predicted the wakeup time but misspredicts an extra onset time, causing false positive (Figure 8) .

We also have rare false negatives. Almost all of them occurs along with zero readings during wake periods. For Figure 9, the target around 200 is surrounded in what seems like a sleeping period. While in Figure 8, the sudden drop around 900 might represent a period where the user has taken the device off, but without the context does seem like a ”onset” scene.

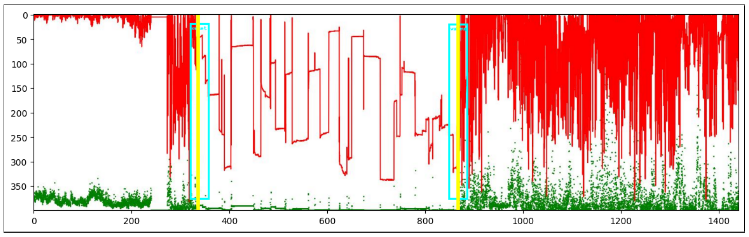


Figure 7: Example sample which correct prediction by our model. True labels shown in yellow, predictions shown in blue.

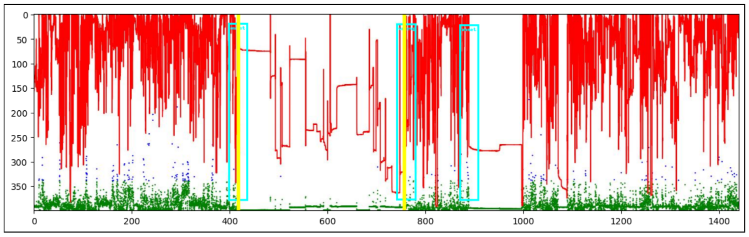


Figure 8: A sample with prediction where no target is related

7 Baseline Model

We implemented two baseline algorithms for the primary model to compare against: Mean Predictor and Random Forest.

7.1 Mean Predictor

The mean predictor serves as a simple baseline for sanity check. We found the average onset and wakeup time from the training label [4]. Then, the model always outputs this average for any input. As expected, the result was very poor (Table 2): there no correlation since our output is constant, and a large RMS error of 168 minutes.

RMSE (minutes)	R2	EDAP Score
168.41509	0.0	0.04217

Table 2: Mean Predictor quantitative metrics

7.2 Random Forest with Temporal Statistical Feature

This is a simple model with good performance on timeseries data, and has been proven to work on sleep accelerator data. Random Forest is robust in handling high-dimensional data and capturing complex relationships in the dataset. However, pure Random Forest is unsuitable for time series data. Therefore, we focused on feature engineering using temporal statistics. The preliminary results obtained from the Random Forest model (as shown in Figure 10) served as a valuable benchmark for our project.

Feature Engineering: The feature engineering phase captures temporal dynamics pertinent to sleep detection. We generated rolling means and maximum values of ’enmo’[5] over time windows ranging from 5 minutes to 8 hours. Additionally, the first variations of ’enmo’ and ’anglez’[6] were computed over these time windows to capture the rate of change in these variables, which could be indicative of transitions between sleep and wake states. The absolute values of these engineered features were

[4] The average wakeup and onset time is 07:19 AM and 10:40 PM, an unimaginable dream to undergrads

[5] Magnitude of acceleration

[6] gyroscopic rotation

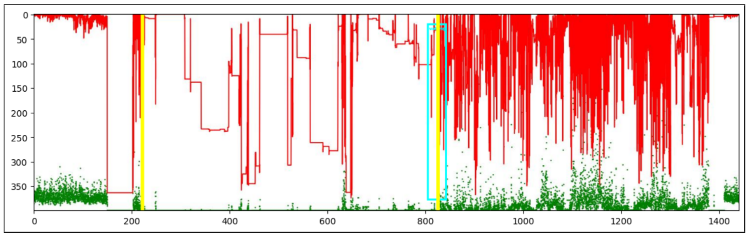
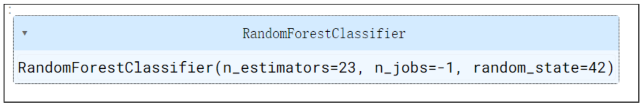


Figure 9: A sample with a target not predicted

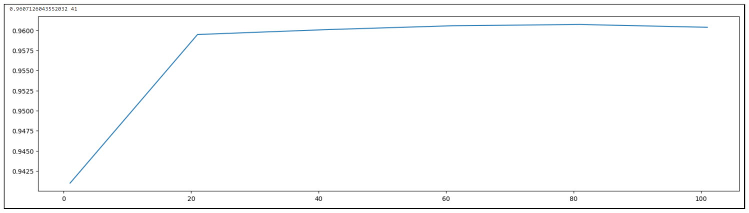


```
RandomForestClassifier
RandomForestClassifier(n_estimators=23, n_jobs=-1, random_state=42)
```

Figure 10: Random Forest Model

computed to ensure data consistency. These engineered features were appended to the original dataset to improve the model's ability to accurately detect sleep states.

Model Tuning: The learning curve guided tuning of the Random Forest classifier's number of estimators. Figure 11 indicates optimal performance at 23 estimators, balancing training and overfitting. Consequently, 23 estimators were selected for the classifier to ensure general and efficient model operation.



```
0.9607126043552032 41
```

Figure 11: Learning curve for Random Forest model with n estimators

Result: Our approach got a result of EDAP=0.349, R2=0.563, RSME=134.6 minutes, much better than the mean predictor.

8 EVALUATION OF MODEL ON NEW UNSEEN DATA

8.1 ACQUISITION OF NEW DATA

To ensure our model can generalize on unseen data, we used a dataset from the University of Newcastle (van Hees et al., 2018). It contains wrist-worn accelerometer data from 28 sleep clinic patients along with labeled ground truths.

8.2 EXPLANATION OF DISTRIBUTION DIFFERENCE

We identified three series with tiny changes in 'anglez' and 'enmo' readings compared with others. It indicated user taking off the wrist band or sensor malfunction, and thus exclude from our analysis.

We also noticed that the distribution of the feature was very different from our original dataset. For 'enmo', the original data mean was 0.0413, with quartiles at 0.0013, 0.017, and 0.043. The new data mean was 7X lower at 0.0067, with quartiles at 0.0023, 0.0036, and 0.0051. This indicates much lower overall movement intensity in the new dataset. This may be attributed to different sensors, or the different sampled population of sleep clinic patients rather than healthy individuals.

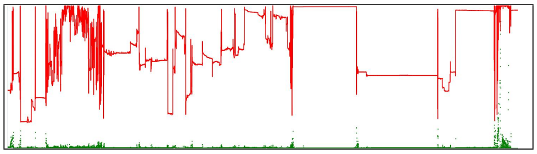


Figure 12: An example of normal series

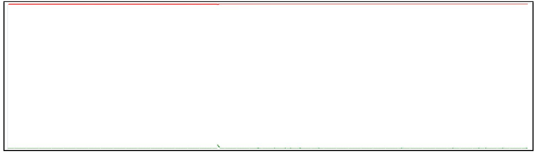


Figure 13: An example of series with tiny change

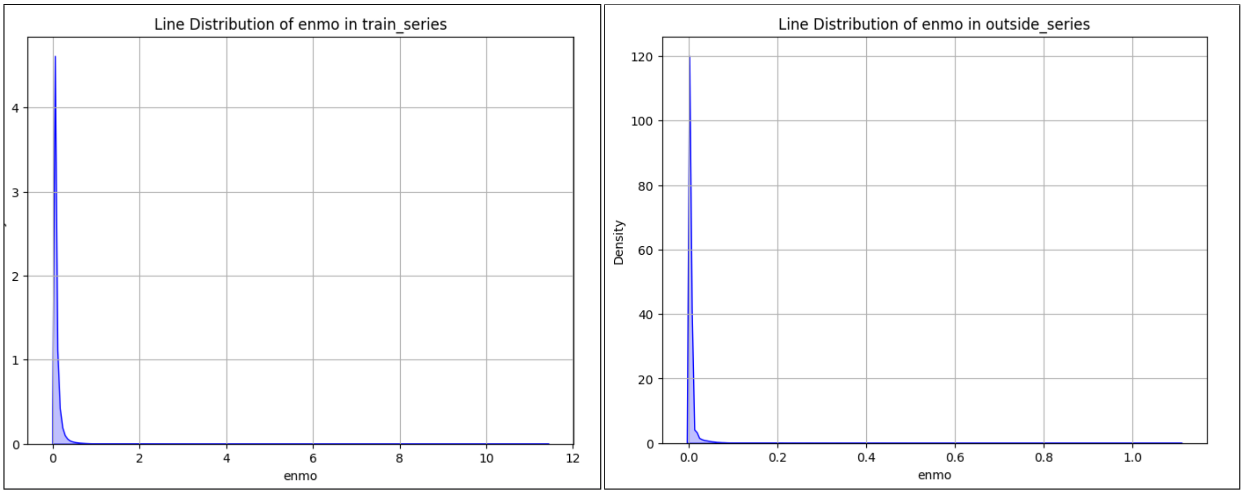


Figure 14: Comparison of enmo distribution

8.3 Interpreting The Test Result

Our model did not perform well on this unseen data. The RMSE error greatly increased to 135 minutes. This is due to having a few samples with abnormally large error. This also caused the stark decrease in R-squared. The EDAP precision score dropped to 0.61. Using tolerance threshold of 10 minutes, 64.5% of the predictions are correct. At 20 minutes tolerance, the precision is 71.7%.

RMSE (minutes)	R2	EDAP Score
135.089	0.584	0.612

Table 3: Mean Predictor quantitative metrics

9 DISCUSSION

This result suggests for adding more knowledge-based assumptions into the model when generating figures. Or raising the confidence threshold for the model to predict one as target. These tuning methods will be left for future work.

9.1 PERFORMANCE ANALYSIS

With an EDAP precision of 0.79 and 91.0% precision for tolerance of 20 minutes, our model is performing very well especially when compared to sleep state models developed by professional researches. For our exact problem, currently the best model has an EDAP of 0.84 , only 5% better than our model. For similar sleep state estimation model, the best model we found had an accuracy of 0.88. This model also used a vision based approach, but combined it with features generated with BiLSTM. This is a promising possibility for future development.

Our model demonstrates significant real-world applicability in scenarios with higher error tolerance. While 76% precision for a 5-minute error tolerance makes it unsuitable for precision-sensitive applications like sleep apnea diagnosis, its precision rises to 86.4% and 91.0% for tolerances of 10 and 20 minutes. This makes it well-suited for practical applications such as assistive sleep pattern analysis for medical professionals (as a supplementary tool) and as a sleep tracker for broader purposes, encompassing general health, fitness, or educational contexts.

9.2 Lesson learned

We learned that ”conventional ”Conventional wisdom” tell you that this problem is a time series problem and would be solved by the RNN family. Yet further investigation show that RNN is not performing well on this as each timestamp does not have its ”meaning” as a word would have.

A surprising result is that RCNN, a model originally used for object detection, offers a good result. This is a good example of how to decide which NN to use when facing a problem would require much thought and dig deeper into the problem itself.

10 ETHICAL CONSIDERATIONS

In developing our project focused on sleep monitoring through accelerometer data, we recognize the importance of ethical considerations in our work.

10.1 Data Privacy

We prioritize data privacy and are committed to complying with international data protection HIPAA regulations. We removed the personal identifiers immediately after downloads from the data gathered in the ”New Data” section. We also kept all gather data private to prevent unathorized access.

10.2 Data Bias

We recognize that our dataset, sourced from New York City’s Healthy Brain Network, may have demographic biases. Consequently, our findings might not be globally applicable, and we refrained from making broad generalizations.

10.3 Model Limitations and Responsible Use

We emphasize that our model is an assistive tools for monitoring sleep patterns. We hereby declare that our model predictions will contain errors, and is not intended as the sole source for a diagnosis, preventing potential misuse of our model.

11 PROJECT DIFFICULTY

Sleepsense presented unique challenges, primarily stemming from the intricate nature of the accelerometer data used for sleep monitoring. Unlike standard object detection tasks, interpreting sleep states from such data is inherently complex, due to the subtle and variable patterns that are difficult to discern, even for human experts. Moreover, collecting and preprocessing additional data posed significant difficulties, as it required transforming the new accelerometer readings into a compatible format for our model, which was not straightforward due to the distinct characteristics of the ’enmo’ and ’anglez’ features. Our project also demanded an exploratory approach to model architecture, as no established solution existed for this type of analysis, leading us to experiment with various configurations before identifying a suitable architecture that performed beyond expectations. This process not only tested our technical abilities but also enhanced our understanding of deep learning applications in real-world scenarios.

REFERENCES

Child mind institute - detect sleep states competition data. https://www.kaggle.com/competitions/child-mind-institute-detect-sleep-states/data. Accessed: 2023.Dec.1.

S. Chambon, V. Thorey, and P.J. Arnal. Dosed: A deep learning approach to detect multiple sleep micro-events in eeg signal. *Journal of Neuroscience Methods*, 321:64–78, 2019.

Chun Fu. Model: Sleep state deep learning model. URL https://www.kaggle.com/code/patrick0302/viz-of-sleeping-time-series.

Ahnaf Rashik Hassan and Abdulhamit Subasi. A decision support system for automated identification of sleep stages from single-channel eeg signals. *Knowledge-Based Systems*, 128:115–124, 2017.

Tarek Lajnef, Sahbi Chaibi, and Perrine Ruby. Learning machines and sleeping brains: Automatic sleep stage classification using decision-tree multi-class support vector machines. *Journal of Neuroscience Methods*, 250:94–105, 2015.

E. Olofsen, J.W. Sleight, and A. Dahan. Permutation entropy of the electroencephalogram: a measure of anaesthetic drug effect. *British Journal of Anaesthesia*, 6:810–821, 2008.

Huy Phan, Fernando Andreotti, and Navin Cooray. Seqsleepnet: End-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 3:400–410, 2019.

V. Thorey S. Chambon and P.J. Arnal. Dosed: A deep learning approach to detect multiple sleep micro-events in eeg signal. *Journal of Neuroscience Methods*, 2019.

Akara Supratak, Hao Dong, Chao Wu, and Yike Guo. Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11:1998–2008, 2017.

Vincent van Hees, Sarah Charman, and Kirstie Anderson. Newcastle polysomnography and accelerometer data (1.0). Zenodo, 2018.

Vincent T. Van Hees, Severine Sabia, and Kirstie N. Anderson. A novel, open access method to assess sleep duration using a wrist-worn accelerometer. *PLOS ONE*, 11:e0142533, 2015.

Vincent Theodoor Van Hees, S. Sabia, and S.E. Jones. Estimating sleep parameters using an accelerometer without sleep diary. *Scientific Reports*, 1:12975, 2018.

S. E. Jones A. R. Wood Vincent Theodoor van Hees, S. Sabia. Estimating sleep parameters using an accelerometer without sleep diary. *Sci Rep*, 8(1), 12975, 2018. doi: 10.1038/s41598-018-31266-z. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6113241/.

CODE ACCESS

Link to GitHub repository: https://github.com/Nathan903/sleepSense