Detection of Sleep States

P16 : Ashish Chander, Harsh Shelar, Bhavya Ahir
Department of Computer Science, North Carolina State University
Raleigh, NC 27695
{achande3, hshelar, bdahir}@ncsu.edu

1 Introduction

The project focuses on detecting sleep states using wrist worn accelerometer data. This problem statement is an ongoing competition on kaggle (ending on 5th December 2023). The problem statement is provided by the Child Mind Institute. The dataset can be found here.

1.1 Problem Statement

The objective of the problem is to detect sleep onset (marks the start of the sleep) and wake events(signifies end of the sleep). The data is collected from wrist worn accelerometer. The data is annotated with onset and wake-up events. It is in the form of time series. We are given with 277 series and each series has on average 84 nights of data. Each series is a continuous recording of accelerometer data for a single subject spanning over a period of many days. For each recording in the series we have to predict the likelihood of it being an onset or wake-up event.

1.2 Data Description

The dataset contains approximately 500 extended recordings of wrist-worn accelerometer data, with annotations for two types of events: "onset," which marks the start of sleep, and "wakeup," which marks the end of sleep. Task is to identify when these two events occur in the accelerometer data series. We will define sleep in terms of accelerometer data as the longest single period of inactivity while the watch is being worn.

Training Data: Training data is given in series format, where each series is a continuous recording of accelerometer data for a single subject spanning over a period of many days.

Below is the table summarizing data under train_series.parquet file.

Parquet - A Parquet file is a columnar storage format commonly used in big data analytics, known for its efficiency, compression, and support for schema evolution.

Attribute	Description
series_id	Unique identifier for each accelerometer series. Unique within a series.
step	Timestep(int) for each observation within a series.
timestamp	Datetime in the format:%Y-%m-%dT%H:%M:%S%z.
anglez	It is a common metric for detecting sleep and represents the arm's angle relative to the body's vertical axis.
enmo	ENMO - Euclidean Norm Minus One is a measurement that considers the overall acceleration from various accelerometer signals. It removes negative values and is one of several frequently calculated features, even though it's not a standard acceleration measure.

Second training file train_events.csv. This file will be used to generate the label data for the train series file described above.

Attribute	Description
series_id	Unique identifier for each accelerometer series. Unique within a series.
night	A list of potential onset/wakeup event pairs, only one of these event pair can occur for each night.
event	Event type - onset or wakeup.
step	Timestep(int) for each observation within a series.
timestamp	Recorded time of the occurrence of the event in accelerometer series in the format:%Y-%m-%dT%H:%M:%S%z.

Testing Data : The test file contains series data with the same attributes as the training file. The aim is to predict the event occurrences within these series.

2 Background

Sleep plays a vital role in regulating mood, emotions, and behavior across all age groups, especially on children. By effectively identifying periods of sleep and wakefulness using data from accelerometers worn on the wrist, researchers can glean valuable insights into sleep patterns and gain a clearer understanding of disruptions in children's sleep.
This effort aims to enhance researchers' capacity to analyze accelerometer data used in sleep monitoring, facilitating larger-scale sleep studies. Ultimately, this competition's outcomes could enhance understanding and recommendations regarding the significance of sleep.

3 Method

3.1 Exploratory Data Analytics

The data set's "step" values, evenly spaced at 5-second intervals, exhibit distinct temporal patterns. Some recordings encompass a full 24-hour period, while others cover shorter time frames. On average, the data set spans about 14.7 days, with up to 75% of the data extending to 20.7 days.The summary for this attribute is shown in Figure 1.

ENMO measurements, likely expressed in gravity units (g), suggest opportunities for sleep pattern analysis. Clipping extreme ENMO values at 1.5 or 1.0 gravity units is suggested to improve sleep prediction accuracy. The summary and description for this attribute is shown in Figure 2 and Figure 3 respectively.

```
step max is 1433879.0
step min is 0.0
step mean is 254802.140625
step standard deviation is 177892.96875
step median is 234519.0
step first quantile is 115812.0
step third quantile is 357196.0
```

Figure 1: Summary of "Step" attribute values

```
enmo max is 11.433699607849121
enmo min is 0.0
enmo mean is 0.04131503403186798
enmo standard deviation is 0.101828932762146
enmo first quantile is 0.0013000000035390258
enmo median is 0.01720000058412552
enmo third quantile is 0.043699998408555984
enmo 99 percentile is 0.39430001378059387
```

Figure 2: Summary of "ENMO" attribute values

Notably, 'ANGLEZ' metrics, measuring arm angle relative to the body's vertical axis, reveal intriguing patterns. Negative 'ANGLEZ' values predominate during nighttime hours (22:00 to 05:00), indicating distinct sleep-related arm positioning behaviors. The summary for this attribute is shown in Figure 4. This combination of 'step,' ENMO, and 'ANGLEZ' metrics provides valuable insights for a comprehensive analysis of activity and sleep patterns.

The dataset presents compelling observations about individuals' sleep patterns. It's evident that the majority of subjects typically awaken between 6-8 am and retire to bed around 8-10 pm, in line with general expectations as shown in Figure 5. Moreover, a significant portion of the dataset records sleep durations of 8-10 hours per night (day), with 9 hours being the most frequent duration as shown

```
count    532800.000000
mean          0.071785
std           0.235867
min           0.000000
25%           0.000000
50%           0.004400
75%           0.043100
max          11.433700
Name: enmo, dtype: float64
```

Figure 3: Description of "ENMO" attribute values

```
anglez max is 90.0
anglez min is -90.0
anglez mean is -8.810453414916992
anglez standard deviation is 35.52187728881836
anglez first quantile is -31.85890007019043
anglez median is -9.597900390625
anglez third quantile is 11.300200462341309
anglez 99 percentile is 78.9126968383789
anglez 1 percentile is -84.65339660644531
```

Figure 4: Summary of "AngleZ" attribute values

in Figure 6. Notably, this pattern holds consistently across a wide range of observations, providing a robust baseline for further analysis of sleep behaviors within the dataset.

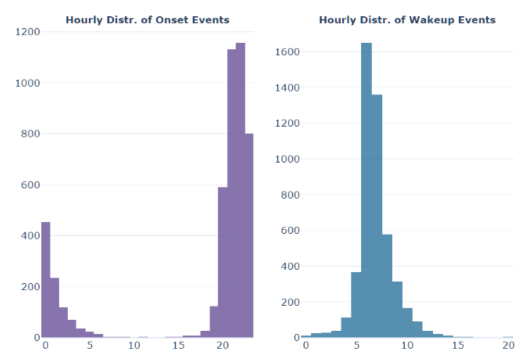

Figure 5: Hourly distribution of Sleep onset and wakeup event

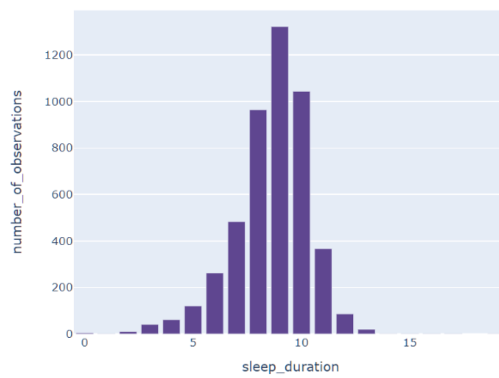

Figure 6: Distribution of sleep duration

3.2 Shrinking Data

We observed that out of the 277 series given to use only 37 of the series had fully labelled data i.e corresponding annotation for onset and wake-up event was present. For rest of the series either the annotation was missing or partially present. We figured replacing these missing values with estimated values would introduce noise to the data so we shrinked our dataset and used the 37 fully labelled series. Note here 37 series does not mean 37 rows, for each series on average 84 nights of data is present and each night would have 17280 recordings so approximately we had 37*84*17280 rows of data. Below figure shows the distribution of full, partial and missing series.

3.3 Gaussian Distribution

Gaussian distributions can be employed to represent the likelihood of sleep onset or wakeup times within a 24-hour period.

Transform the accelerometer data into 17500 windows, representing a 24-hour interval and by fitting Gaussian distributions to the data, we can model the expected time and variability associated with these events, providing a probabilistic representation of sleep patterns.

Train the model to predict probabilities of sleep events at different time intervals within a day. Interpret Gaussian distributions as probabilistic representations of event likelihoods across time intervals.

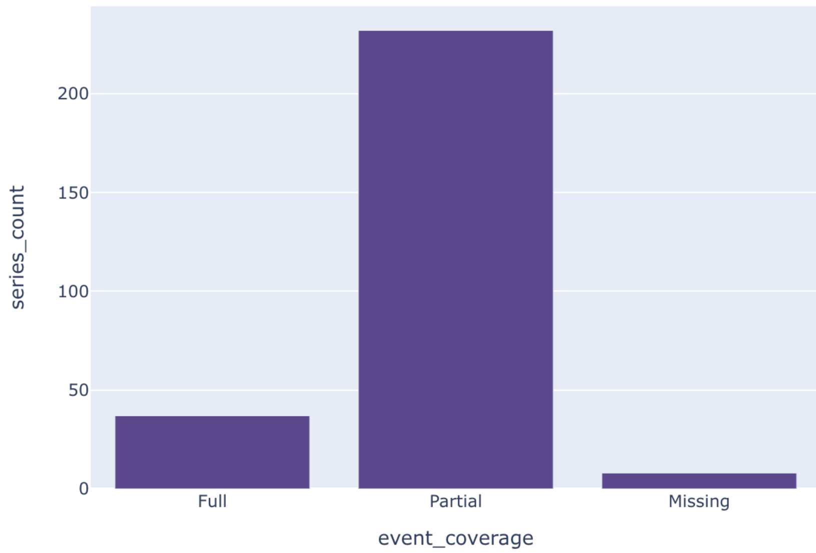

Figure 7: Number of series with full, partial and missing event level coverage

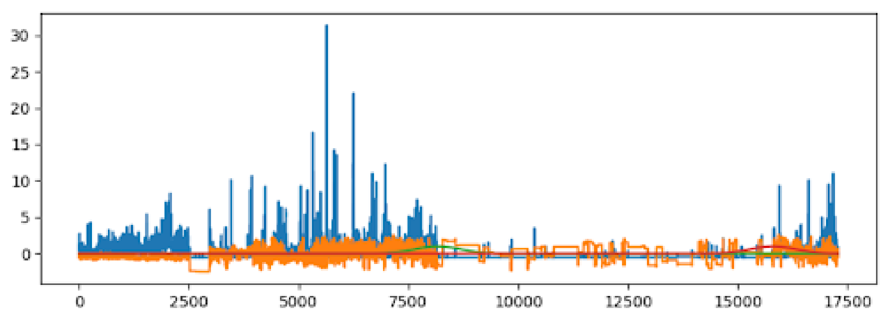

Figure 8: Hourly distribution of Sleep onset and wakeup event

3.4 Residual GRU

We make use of a gated RNN with a residual connection to allows gradient flow without losses. The model has layers such as GRU, Fully Connected layers and Norm Layer, Hidden layers with ReLU activation and the Residual connection as shown in Figure 8.

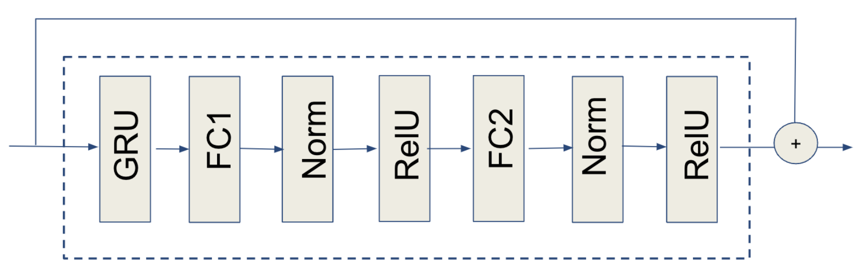

Figure 9: Residual GRU model

With respect to the GRU Layer, we process input in both forward and reverse direction to capture past and future relation in data.GRUs, like LSTMs, employ gating mechanisms to control the flow of information within the network. However, GRUs use a simpler architecture with two gates: the reset gate and the update gate.

Fully Connected Layers and LayerNorm introduces non-linearity and normalization to the processed output.Each neuron in the fully connected layer is associated with a weight for each input feature and a bias term.Layer Normalization normalizes the input of a layer across the feature dimension.

ReLU introduces non-linearity by setting negative values to zero.ReLU function is zero for any negative input and equal to the input for any non-negative input.

During backpropagation, if the gradient becomes very small as it propagates through the layers, the skip connection provides an alternate path for the gradient to flow.With the skip connection, original input sequence is added to the processed output. This allows the network to learn the difference between the input and output.

3.5 Evaluation

We obtain the probability score as the output from the RNN model. This score is considered the confidence score for onset and awake for each data point(frame). We then make use of a rolling window to go through all the confidence scores in the given time series and find the frame with maximum confidence score. This is performed for both values, the onset frame and the wakeup event frame from the time series.

To evaluate the effectiveness of the model we use an evaluation metric called Event Detection Average Precision [4]. This metric is useful for event detection in time series data.

To calculate the metric, we compare the detections made by our model and match them to the ground truth events within error tolerances. We then, calculate precision and recall, sort the confidences in descending order, plot the values, and finally calculate the area under the graph for the plotted values. This area under the graph is the average precision for one label and threshold value.

To calculate the overall average precision we calculate the average precision for both labels (onset and wakeup), and for each label, we calculate the average precision for a set of error tolerances (defined in number of steps) and take the average of all the values. This final value is the event detection average precision for our model.

For both event classes, we define error tolerance thresholds 12, 36, 60, 90, 120, 150, 180, 240, 300, and 360 in steps (1 step is 5 seconds).

The caculation of the metric is visualized in Figure 10.

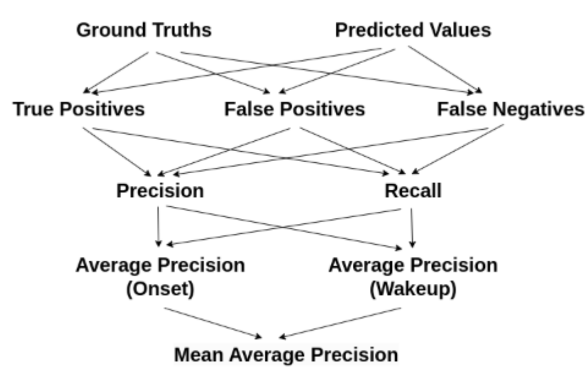

Figure 10: Event Detection Average Precision

4 Experiments

4.1 Data Preparation and preprocessing

The training data is present in the files train-events.csv and train-series.parquet. We tried reading the files and realized the files were large in size and could be reduced.

First, we checked the train-events.csv file and got the results (shown in Figure 7).

```
sleep_train.dtypes

series_id      object
night           int64
event          object
step          float64
timestamp      object
dtype: object

# Memory usage of each column in bytes
sleep_train.memory_usage(deep=True)

Index           128
series_id   1001052
night        116064
event        906750
step         116064
timestamp    933921
dtype: int64

# Memory in MB
sleep_train.memory_usage(deep=True).sum() / (1024 * 1024)

2.931574821472168
```

Figure 11: Data types and memory usage of original events data

```
reduced_sleep_train.dtypes

night         uint16
event          uint8
step         float64
timestamp     object
id_map        uint16
dtype: object

reduced_sleep_train.memory_usage(deep=True)

Index           128
night         29016
event         14508
step         116064
timestamp    933921
id_map        29016
dtype: int64

reduced_sleep_train.memory_usage(deep=True).sum() / (1024 * 1024)

1.0706453323364258
```

Figure 12: Data types and memory usage of new events data

We perform the following data type conversions for the train-events data columns to reduce size:

- series-id: from object -> uint16 : There are 277 unique ids. Hence ,we remapped them with an id-map from 0 to 277 for easier development and size reduction.
- night: from int64 -> uint16
- event: from object -> uint8: relabeled as follows: 'onset':'1', 'wakeup':'2'
- step: from int64 -> uint32
- timestamp: kept the same

Figure 8 shows the result after performing the conversions:

We have reduced the memory usage from 2MB to approximately 1MB. Almost a 50 percent reduction.

For the train-series.paraquet file, we have made the following data type conversions to reduce size:

- series-id: from object -> uint16 : There are 277 unique ids. Hence ,we remapped them with an id-map from 0 to 277 for easier development and size reduction.
- step: from int64 -> uint32

Figure 9 and Figure 10 show the initial and the final memory used, respectively.

```
[5]: series.memory_usage(deep=True)

[5]: Index                128
     series_id     8828297460
     step           511785360
     timestamp    10363653540
     anglez         511785360
     enmo           511785360
     dtype: int64

series.memory_usage(deep=True).sum() / (1024 * 1024)

[10… 19767.10053253174
```

Figure 13: Data types and memory usage of original events data

```
[8]: new_series.memory_usage(deep=True)

[8]: Index                128
     step           511785360
     timestamp    10363653540
     anglez         511785360
     enmo           511785360
     id_map         255892680
     dtype: int64

[9]: new_series.memory_usage(deep=True).sum() / (1024 * 1024)

[9]: 11591.818264007568
```

Figure 14: Data types and memory usage of new series data

4.2 Model Training and Evaluation

The Kaggle competition provided us with the training and testing data. We additionally split the training data into training and validation data sets, using an 80-20 split (80 percent of data as training and 20 percent as validation). We trained the model on the training set and validation dataset. Finally, we evaluated the Event Detection Average Precision for the model on the testing dataset.

4.3 Results

4.3.1 Data Preprocessing

Using the above data preprocessing approach the memory usage reduced from 19767MB to 11591MB. Approximately a 41% decrease.

4.3.2 Evaluation of the Residual GRU model

The Event Detection Average Precision for our Residual GRU model came out to be 0.62.

The current highest value submitted in the competition is 0.797 as shown in Figure 15.

#	Team	Members	Score
1	K_mat		0.797
2	Chris Deotte		0.784
3	Ruby		0.783

Figure 15: Kaggle Leaderboard

5 Conclusion

In this work, we created a model to evaluate time series data from wrist-worn accelerometers to detect sleep events such as sleep start (onset) and sleep end (wakeup). A majority of our time was spent in pre-processing the data.

The data type conversions and the conversion of the given data to Gaussian distribution took quite some effort and time to get right. We also spent considerable time trying to shape our data in such a way that our Residual GRU model could be trained on it. In the whole process, we were exposed to a variety of different pre-processing techniques and ideas. Apart from pre-processing, we researched a lot of machine learning and deep learning models before finalizing the Residual GRU model. This enabled us to learn more about the types of models and algorithms used to evaluate time-series data.

Our model used the anglez and enmo values to determine the wake-up and onset states. We would also like to incorporate environmental factors such as sunlight exposure and outside noise as these also determine sleep start and end. In addition, daily activity (things like the amount of physical activity) could also be evaluated to predict sleep onset and wake.

Better models could be explored to capture these attributes and additional data could be collected to improve the accuracy of the prediction. By precisely identifying periods of sleep and wakefulness through the analysis of accelerometer data from wrist-worn devices, researchers can enhance their comprehension of sleep patterns.

6 References

[1] Dreem Open Datasets: Multi-Scored Sleep Datasets to compare Human and Automated sleep staging

[2] SeqSleepNet: End-to-End Hierarchical Recurrent Neural Network for Sequence-to-Sequence Automatic Sleep Staging

[3] DOSED: a deep learning approach to detect multiple sleep micro-events in EEG signal

[4] Event Detection Average Precision

[5] IoU a better detection evaluation metric

[6] A Guide to Recurrent Neural Networks: Understanding RNN and LSTM Networks

[7]Sleep classification from wrist-worn accelerometer data using random forests

[8]Estimating sleep parameters using an accelerometer without sleep diary

7 Appendix

7.1 Distribution of Work

7.1.1 Data Cleaning

hshelar: Reduced the size of the training and testing data as described in the experiments section.

bdahir: Explored the dataset and figured how given data can be shrinked(described in 3.2).

achande3: Performed exploratory data analysis to visualize relations and trends in the data. Refactored timestamp column to reduce data size.

7.1.2 Methods Development

bdahir: Explored how we can leverage gaussian distribution to apply RNN.

hshelar: Explored Random Forest algorithm to determine onset and wakeup states for the data.

achande3: Explored and implemented the residual GRU model and fine-tuned the parameters.

7.1.3 Exploration

bdahir: Explored how we can leverage gaussian distribution to apply RNN and how z-angle, enmo and other parameters impact sleep.

hshelar: Explored Random Forest algorithm to determine onset and wakeup states for the data.

achande3: Explored the benefit of using a Residual connection in the GRU model.

7.1.4 Results Analysis

bdahir, achande3: Explored different methods to evaluate time series model.

hshelar: Calculated the performance of the model using event detection average precision.

7.1.5 Conclusion Drawing

bdahir, achande3, hshelar : Derived conclusions, learnings and possible future enhancements that we can work towards.

7.1.6 Presentation Preparation

hshelar: Created the slides for model evaluation using the event detection average precision and created diagrams to better explain the concept.

bdahir : Created the slides for Introduction, Problem statement, Data description, Background and Exploratory Data analysis - Shrinking Data and Gaussian Distribution.

achande3: Created the slides and illustrations with respect to the Residual-GRU model and extracting the time frame with maximum confidence by making use of a rolling window.

7.1.7 Final Report Creation

bdahir: Helped curate Introduction, Problem statement, Data description, Background and Exploratory Data analysis - Shrinking Data.

hshelar: Helped curate Experiment Setup, Evaluation and Results sections.

achande3: Structured the entire report and curated Gaussian Distribution and Residual GRU.

8 Code

Github Link: https://github.ncsu.edu/hshelar/engr-FALL-2023-P16