SleepFormer: A Transformer-based Sleep Classifier

Harshil Vagadia, Xintong Qu, Jake Michiels, Azeez Ishaqui
Georgia Institute of Technology
{harshil, xqu46, aishaqui3, jmichiels3}@gatech.edu

Abstract

Recently, sparse-attention mechanisms in transformers have shown promising results in NLP due to their ability to capture long sequence features. In this work, we explore the use of sparse-attention transformers in the Human Activity Recognition (HAR) task. Specifically, we work on sleep state prediction from wrist-worn accelerometer data. We're proposing SleepFormer, a novel sparse-attention transformer architecture specifically designed to deal with long high-frequency timeseries data obtained from the accelerometer. Firstly, we introduce a time-aware embedding layer which replaces the traditional positional embeddings in vanilla transformers. Secondly, we use sliding window attention in the transformer to allow our model to work on longer sequences. Thirdly, we pass the transformer representations through a Conditional Random Field (CRF) layer to improve label consistency and reduce superfluous transitions. We conduct extensive experiments and fine-tuning on various variants of transformers and other baselines (both DL and ML). Our experiments shows that our method achieves 17% higher Event Detection AP than previous SOTA-baseline.

1. Introduction

Sleep plays a fundamental role in human health and well-being. It is a complex physiological process with distinct stages, each associated with unique patterns of brain activity and body movements. Accurate and non-intrusive monitoring of sleep stages is crucial for understanding sleep disorders, developing effective treatments, and improving overall health outcomes. Polysomnography (PSG) accurately monitors sleep habits, but they are expensive, intrusive and impractical for daily use. Another promising avenue for sleep monitoring is leveraging wrist-worn accelerometer data, which capture subtle movements associated with different sleep stages. This project aims to tackle the challenge of predicting sleep states from such accelerometer data using deep learning techniques.

Formally, this is a sequence tagging problem. The inputs consist of a time series of accelerometer data, represented as $X = \{x_1, x_2, \ldots, x_t\}$ where x_i denotes the accelerometer data at timestamp i. The accelerometer data is collected at measured intervals, forming a sequential dataset representing the wearer's physical activity over a specific duration.

The outputs, on the other hand, are discrete labels indicating the sleep state at each timestamp. Formally, the output sequence is represented as $Y = \{y_1, y_2, \ldots, y_t\}$ where $y_i \in C$ is the label denoting sleep state at timestamp i. For binary sleep classification, $C = \{Asleep, Awake\}$ while for multi-stage sleep classification $C = \{Awake, N_1, N_2, N_3, REM\}$. In this project, we only work with binary sleep classification.

Accurate and real-time classification of sleep states from wrist-worn accelerometer data holds paramount importance in the fields of healthcare and well-being. Sleep disorders, affecting millions globally, contribute significantly to various health problems, including cardiovascular issues, obesity, and mental health disorders [16]. Timely detection and understanding of sleep patterns are vital for diagnosing sleep disorders such as sleep apnea and insomnia. Furthermore, precise monitoring of sleep quality aids in optimizing daytime productivity, cognitive function, and emotional well-being. By developing robust models that can discern sleep states from accelerometer data, this research not only advances our understanding of human sleep patterns but also provides a practical tool for healthcare professionals to diagnose, treat, and improve the lives of individuals suffering from sleep-related disorders. Additionally, it has the potential to empower individuals to take proactive measures to enhance their sleep quality, ultimately leading to improved overall health outcomes.

The primary objective of this project is to develop highly accurate and efficient deep learning transformer-based model capable of classifying sleep states from wrist-worn accelerometer data. Our main contributions can be summarised as follows: (1) We introduce a unique timestamp embeddings, which replaces the traditional positional embeddings for timeseries data. (2) We explore the use of sliding window attention to capture long-range dependencies in sequential data. (3) We perform extensive experi-

mentation on various transformer variants and baselines.

2. Related Work

A number of work have focused on classifying sleep states from sensory data provided by various Consumer Sleep Technologies (CSTs). The most common type of CSTs are wrist-worn devices. For example, [2] uses a finger-worn OURA ring which has Photoplethysmogram (PPG), accelerometer, and skin temperature measurement. [17] used a Samsung Gear S2 smartwatch to obtain PPG sensory readings. We focus only on accelerometer and orientation data in this work, partially due to lack of specialised sensors on vast majority of the devices and also due to lack of publicly available datasets with these measurements.

Another body of work uses traditional machine learning (ML) algorithms like SVM and Random Forests for the classification. [3,8,17,22] uses traditional algorithms on top of hand engineered statistical features. Apart from the manual effort required in designing the features, these models fail to capture the temporal nature of the data and thus failing to learn long-term dependencies.

Deep learning (DL) models have shown promising results in sleep classification. According to the survey [9], deep learning models on raw signals perform the best in both binary and multi-stage sleep classification. [7] developed a CNN-LSTM model (which is essentially a LSTM on top of CNN feature extractor) for 2-stage sleep classification using raw accelerometer data. Works like [19] used both raw accelerometer data and its corresponding Fast Fourier Transform, thus combining both raw data and engineered data.

Transformers [26] is a prominent deep learning model that has been widely adopted in fields like Natural Language Processing (NLP) [20, 23], Computer Vision (CV) [5, 11] and speech recognition [6, 10]. However, vanilla transformers are inefficient for long sequences, owing to its expensive multi-head attention operation [15]. To alleviate this issue, many sparse-attention mechanisms have been proposed, including Star-Transformer [13], Longformer [4], Extended Transformer Construction (ETC) [1], BigBird [29]. In this work we explore sliding window attention (a.k.a band attention or local attention).

Conditional Random Fields (CRF) [24] is a class of discriminative models best suited to prediction tasks where contextual information of neighbouring state affects the current prediction. CRFs are used for various tasks such as image denoising [27], phishing detection [21], and musical audio-to-score alignment [14]. Our sleep state prediction tasks contextually depends on the neighbours, i.e. if a person was asleep in the neighbouring timestamps, they are highly likely to be asleep in current timestamp.

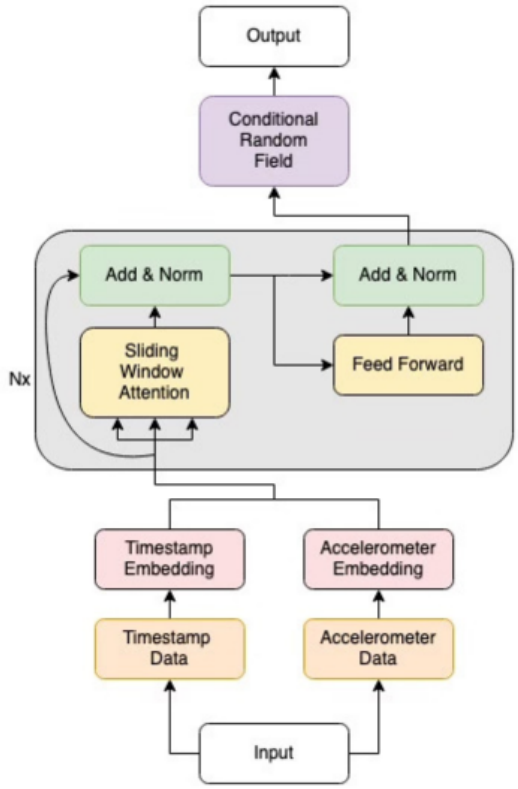

Figure 1. SleepFormer architecture for sleep state classification.

3. Technical Approach

In this section, we describe our approach to the sleep state prediction task. Inspired from the immense success of transformer-based models in NLP, we propose *SleepFormer*, a unique architecture well suited for long timeseries data from wrist-worn accelerometers.

More formally, we want to learn a function $f(.)$ which can classify input $X \in \mathbb{R}^{T \times d}$ (T is the length of timeseries and d is the size of feature vector) into a series of labels $f(X) \in C^T$ (C is the set of classes depending on the task). Here $f(.)$ is modelled by a transformer encoder based architecture described in these section. Fig [1] shows the architecture of our model.

3.1. Embedding Layers

We have two embedding layers, namely accelerometer embedding and timestamp embedding. Accelerometer embedding layer transforms the accelerometer data to a higher dimension latent embedding. Timestamp embedding layer transforms the timestamp signal into a higher dimension representation. Formally, the input $X \in \mathbb{R}^{T \times d}$ is split into accelerometer data $X_a \in \mathbb{R}^{T \times d_a}$ and timestamp data

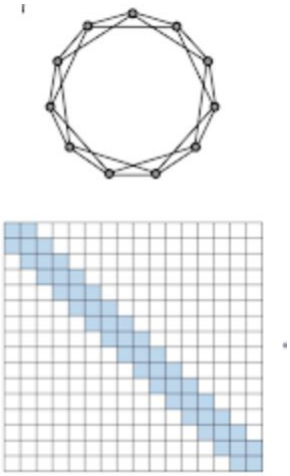

Figure 2. Visual Representation of Sliding Window Attention. Diagram Credits: [12]

$X_t \in \mathbb{R}^{T \times d_t}$. d_a is the dimensionality of accelerometer data ($d_a = 2$ in our case) and d_t is the dimensionality of timestamp vector. We use $d_t = 6$ and the vector consists of numerical values of year, month, day, hour, minute, and second of the timestamp.

Let f_a and f_t denote the accelerometer and timestamp embedding layer respectively. Then the latent representation for downstream transformer module is the concatenation of both embeddings $X_d = [f_a(X_a); f_t(X_t)]$ (X_d is the downstream latent representation). Both f_a and f_t are simply modelled by a linear transformation followed by GELU activation function.

Note that out timestamp embeddings are similar to positional embeddings used in vanilla transformer architectures. We do not use positional embeddings here. We believe timestamp is more important signal in classifying sleep states than the position in a sequence as human sleep is highly correlated with the time of the day. Further, the timeseries we deal with are very long and positional embeddings pose the practical challenge of memory explosion. In the experimental section, we explore using positional embeddings on smaller and manageable chunks of timeseries data.

3.2. Sparse-Attention Transformer Encoder

The latent representation X_d is passed into a transformer based network with multiple stacked modules of alternating attention and fully connected layers. The vanilla transformer architecture struggles with long sequences due to the quadratic time computation of attention layer in the length of input sequence. Since we are dealing with days of high-frequency accelerometer data in our task, this is a serious issue. Therefore we use a combination of sliding window attention and chunking to deal with long sequences.

Firstly, given a long timeseries input $X_d \in \mathbb{R}^{T \times d}$, we divide it into T/T_c chunks of $X_c \in \mathbb{R}^{T_c \times d}$. In our case, $T \approx 100,000$ and $T_c \approx 250$. Each chunk is essentially independent of another, and there is no flow of attention or gradients between them. Chunking is required because such long timeseries simply cannot fit into a reasonable memory. Our chunks were as large as our machine memory allowed.

Secondly, even with chunking, the chunks are significantly larger to disallow full multi-head attention. Thus, we use sliding window attention to further reduce computation and memory overhead. In sliding window attention, each item in the timeseries only attends to a few neighbouring items in a window. In our experiments, we keep out window size $W = 50$. Fig [2] shows a visual representation of sliding window attention. Note that sliding window attention is different from chunking as even though an item only attends to its neighbours, distant items also affect the current item through a chain of neighbours.

3.3. Conditional Random Field

The Conditional Random Field (CRF) serves as the ultimate layer within our model, playing a pivotal role in refining predictions. It introduces a consistency loss in addition to the prediction loss, ensuring not only that the output aligns with the correct class (prediction loss) but also that it maintains coherence with neighboring predictions (consistency loss). This dual-loss mechanism proves instrumental in minimizing unnecessary state changes, thereby enhancing the precision of event detection (described later).

A specific instance of CRF employed in our model is the linear chain CRF. In this variant, the consistency loss is selectively applied between adjacent items. The probability of generating a sequence y given an input X is expressed as:

$$P(y|X) = \frac{1}{Z(X)} exp(\Sigma_{i=1}^{t} \lambda_i f(y_i, y_{i-1}, X_i))$$

In this formula, $f(\cdot)$ represents a learnable function responsible for maintaining both prediction values and consistency, while $Z(\cdot)$ denotes the normalizing factor. There are many variants of $f(\cdot)$ available and we invite readers to see [25] for more details. During the training phase of the CRF, the loss is computed as the negative log-likelihood of generating y from X—the output of the transformer block. In the inference phase, Viterbi-trellis algorithm is employed to generate a sequence that maximizes the probability estimated by the trained CRF. All these functionalities are encapsulated within a library known as *TorchCRF*.

Feature	Description
$Mean_T$	Mean value of signal in Ts interval
Std_T	Std deviation of signal in Ts interval
$Minimum_T$	Minimum value of signal in Ts interval
$Maximum_T$	Maximum value of signal in Ts interval

Table 1. Features used in conventional ML models and their description. $T = [5, 30, 100, 1000]$

3.4. Baselines

3.4.1 Conventional Machine Learning

In this section we will explore how traditional ML techniques namely Random Forest, Logistic Regression, and SVM can be used for sleep state classification. These models will serve as baselines for our work.

Essentially these classifiers will classify the state at each timestamp based on some hand-engineered aggregates for each input feature over a window of T seconds around the required timestamp. These aggregates are passed to the classifier. The classifiers are trained on paired data of aggregates and labels. Table [1] shows the statistical aggregates we use for classification, based on [22].

3.4.2 Deep Learning

Since sequence tagging is a well-researched area especially in NLP, we will use those models as baselines. Based on the success of previous work, we add three more deep learning models to our baseline repertoire, CNNs, LSTMs, and DeepActiNet [7]. We also explore the CRF versions of LSTMs and DeepActiNet to test our hypothesis. To ensure consistency, we exactly replicate the architecture of [28] and [7] for LSTM and DeepActiNet respectively.

No hand-engineered features are extracted from the Deep Learning models. The raw features from the accelerometer are directly passed to the models. To deal with large memory overheads and vanishing gradients issues, we chunk our timeseries into smaller lengths (similar to our transformer approach). We use cross-entropy loss on output logits for all the models (CRF likelihood loss in case of CRF variants).

3.5. Evaluation Metrics

The most common evaluation metrics for sleep stage classification are accuracy, precision, recall, and F1 score. Although accuracy is the most popular metric, it is not a good metric when the labels are imbalanced. This is especially an issue with multi-stage classification, where some states like N_1 are relatively short. For binary classification, accuracy is still a reasonable metric as humans spend $\sim 25\%$ of their day sleeping.

We also explore Event Detection AP as an evaluation metric, which is recommended by Child Mind Institute for their sleep state classification challenge [18]. This metric evaluates events rather than sequences. Events are defined at transition from one label to another (sleep "onset" and "wakeup"). The metric is the mean precision of event prediction averaged over different timestamp error threshold (error threshold is the tolerance for timestamp error for which ground-truth and predicted events match).

4. Dataset

Finding a dataset for sleep state classification is a challenging task, as most works collect their own dataset and do not release them publicly. Specifically, we could not find any large dataset with PPG, ECG or EEG readings. Thus, we resort to the largest accelerometer dataset we could find, released by Healthy Brain Network [18]. This dataset was released as part of a Kaggle challenge. It contain multi-day timeseries accelerometer data of 500 subjects. The dataset was labelled with sleep onset/wakeup events for each subject. The annotation is done by human guided raters instructed with several heuristics on sleep patterns.

In this dataset, the features contain accelerometer data at each timestamp represented as $x_i = (angle_z, ENMO)$. $angle_z$ is the angle of the accelerometer from the z-axis. $ENMO$ is Euclidean Norm Minus One, $ENMO = \sqrt{a_x^2 + a_y^2 + a_z^2} - 1$ where a_x, a_y, a_z are the accelerations along each axis. A separate file contains the events (onset/awake) timestamps for each timeseries and the corresponding event type. The data is collected at 5s intervals with the timestamp being accurately mapped to the nearest second. No other information revealing the subjects involved is present.

5. Experiments and Results

We perform extensive experiments on various baselines and variants of transformer architectures. We used the libraries scikit-learn to implement ML models and PyTorch for the DL models. Our experiments were done on the Google Cloud Platform. We used a NVIDIA T4 GPU with 16GB memory. Our resources were limited by the free credits received for this project. We believe the memory was a bottleneck in many transformer variants and the trends could be different with larger memory and compute.

5.1. Data Preprocessing

As mentioned, the dataset contains the sleep data in the form of events. Since we follow the sleep classification paradigm and not the event detection paradigm, we convert the events into labels for each timestamp and store it separate csv files. We divided the 500 subjects into train-test split in 80:20 ratio and normalised the data based on

train set. Due to the limited compute available, we only use $\sim 25\%$ of the train and test sets for our experiments.

5.2. Model Training and Tuning

Each model was trained separately and independently of other models in our repertoire. For the ML models, we slide a Ts window over the timeseries window over the data, calculate statistical features and train a classifier to predict the class at the center of the window.

For the DL models, the timeseries was divided into chunks and each chunk was processed independently of others. This was done to limit the memory usage of the models (ideally we want the models to process the entire timeseries, but that is not possible for such long timeseries). The DL models were trained to predict the sleep state at each timestamp.

Each model was tuned using intelligent grid search of hyperparameters. Our search parameters include the depth and width of the models, chunk length, learning rate etc. All the weights of our model were randomly initialised and tuned with the Adam optimizer. We report the results of the best hyperparamters obtained. For the speepformer model, we found having 4 attention blocks, 8 heads per attention, 128 embedding size, 0.0005 learning rate, and 8 batch size gave the optimal performance.

5.3. Data Postprocessing

Through our experiments we observed that the models had good sleep prediction scores, but poor event detection AP. Even with the presence of CRF layer, the models were predicting superfluous sleep regions of short duration which leads to false events during event conversion. To remedy this, we tried several smoothing functions.

The first method used two passes of a rolling window. during the first pass any label that did not match a proportion of the labels to the left and right was flipped to match it's neighbors. This should remove all incorrect labels that occur far away from true event boundaries and leave labels clustered around event boundaries. During the second pass the window is placed over individual clusters of labels and the best label is probabilistically chosen from all of the labels that don't match the previous label. This method did not perform very well on the outputs from more complex models because incorrect labels usually occurred together, so they weren't filtered out by the first pass and became candidates for edges during the second pass.

Next we used a Savitzky–Golay filter with a large window to fit points using quadratic regression. This method performs much better than the previous method and consistently approximates the true sleep regions, however it struggles to maintain sharp edges, so event detection performance suffers in some instances.

Lastly, we attempted to implement edge preserving smoothing methods commonly used in computer vision applications, but they failed to outperform the Savitzky–Golay filter or they took too long to execute on the very large series in this dataset.

5.4. Ablation Studies

To accurately judge the effect of each component of our transformer architecture, we perform in-depth ablation study of our model. Formally, we create and evaluate the following variants of our transformer architecture:

- *SleepFormer*: Our primary architecture with timestamp embedding, sliding-window attention and CRF layer.
- *SleepFormer-NoCRF*: SleepFormer architecture with the final CRF layer replaced by a fully connected layer. This model was trained using cross-entropy loss.
- *SleepFormer-FullAttention*: SleepFormer architecture with sliding-attention replaced by global attention i.e. each item attends to every other item in the timeseries. Note that global attention requires larger compute and memory, hence the sequence length inevitable has to be reduced.
- *SleepFormer-Positional*: SleepFormer architecture which uses trainable positional embeddings instead of timestamp embeddings. Note that positional embedding is based on the position in the chunk, not the entire timeseries. The timeseries is too long to have a positional embedding for each position.

5.5. Results

All models underwent training using 80% of the total dataset, designated as the training set. The performance metrics on a held-out test set are detailed in Table [2], while Figure [3] visually presents qualitative outcomes for our SleepFormer model before and after post-processing.

In summary, our findings reveal a significant performance advantage for Deep Learning algorithms over traditional Machine Learning models. Notably, SleepFormer surpasses the performance of state-of-the-art (SOTA) baseline models, achieving a maximum Event Detection AP score of 0.61. CRF models outperform their non-CRF counterparts, underscoring the importance of neighboring consistency enforced by CRF in enhancing performance. The utilization of positional embedding, as opposed to timestamp embedding, leads to a performance decline, emphasizing the relevance of timestamp information over positional cues within a chunk. Additionally, employing full attention necessitates a reduction in chunk size, resulting in a performance decrease.

Model	Accuracy	Precision	Recall	F1	Event Detection AP
Random Forest	0.901	0.882	0.833	0.853	0.202
Logistic Regression	0.788	0.707	0.626	0.635	0.124
SVM	0.769	0.661	0.597	0.603	0.065
CNN (250)	0.827	0.817	0.827	0.820	0.232
LSTM (250)	0.909	0.922	0.909	0.912	0.344
LSTM-CRF (250)	0.910	0.917	0.910	0.912	0.366
DeepActiNet (250)	0.915	0.927	0.911	0.914	0.411
DeepActiNet-CRF (250)	0.915	0.922	0.915	0.917	0.444
SleepFormer (250)	**0.936**	0.939	**0.936**	0.936	**0.618**
SleepFormer-NoCRF (250)	0.935	**0.941**	0.935	**0.937**	0.580
SleepFormer-FullAttention (100)	0.909	0.909	0.909	0.909	0.512
SleepFormer-Positional (250)	0.924	0.927	0.922	0.924	0.597

Table 2. Evaluation of all ML/DL models based on various metrics. The value in the braces denotes the sequence length of the chunks for DL models.

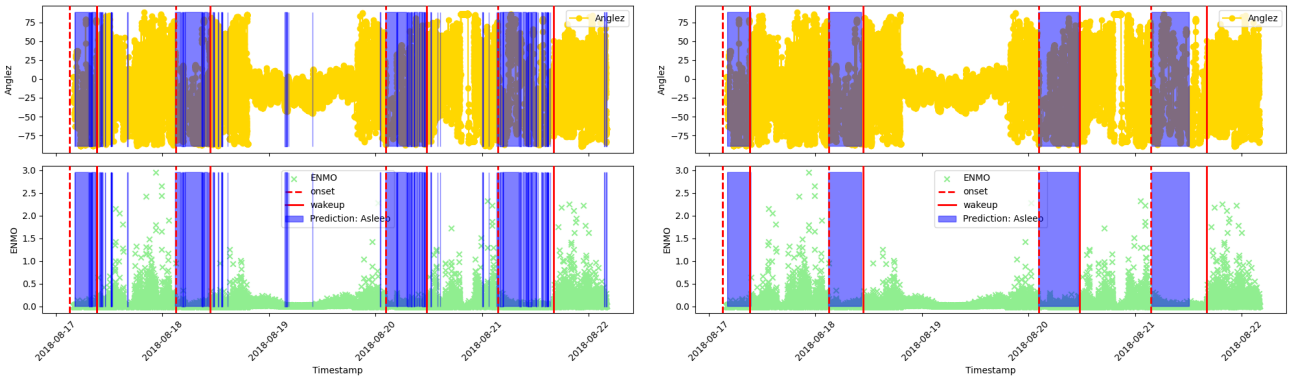

Figure 3. Visualisation of predictions of SleepFormer before (left) and after (right) post-processing. Red lines are ground-truth events and blue region is the sleep predicted region.

While achieving significant improvements, the event detection AP is considerable lower than the corresponding F1 scores. Visualizations in Figure [3] depict the model's ability to identify general sleep regions but struggle with precise localization of transition points. This highlights the necessity for post-processing techniques. The model outputs without post-processing exhibit multiple events within a short timeframe, emphasizing the importance of our post-processing methods in smoothing out these occurrences.

6. Conclusion and Future Work

In this work, we introduced SleepFormer, a novel transformer-based architecture to classify sleep states from wrist-worn accelerometer data. SleepFormer uses timestamp embedding to replace the traditional positional embeddings which are unsuitable for long sequences. SleepFormer also uses sliding-window attention to deal with longer sequences. Finally we use CRF layer to maintain consistency among the predicted outputs. Our model achieves an event detection AP of 0.61 and conclusively beat the previous baselines (both ML and DL).

Our work was significantly limited by the memory available to us. This restricted us to smaller sequence lengths of the chunks processed by the transformer models. Right now our models are able to look ~ 30 minutes in the past/future. We believe extending this to ~ 24 hours would result in significant performance improvement as the model could look at the neighbouring days' sleep to make predictions. This can be achieved by a combination of more memory and efficient implementation of sparse-attention. Furthermore, improving the smoothing method could drastically improve event detection performance. Future work should be focused on adapting edge preserving smoothing techniques commonly found in computer vision to the very long one dimensional sequences in this data-set.

Code: https://github.com/HV007/SleepFormer

7. Contributions

Team Member	Contributions
Harshil Vagadia	Implemented Deep Learning Baselines, Performed SleepFormer Ablation Studies, Designed Poster
Xintong Qu	Implemented Machine Learning Baselines, Designed Diagrams, Designed Poster
Jake Michiels	Implemented Machine Learning Baselines, Designed Postprocessing Pipelines
Azeez Ishaqui	Implemented Machine Learning Baselines, Tuned Deep Learning Models, Designed Poster

References

[1] Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. ETC: Encoding long and structured inputs in transformers. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284, Online, Nov. 2020. Association for Computational Linguistics. 2

[2] Marco Altini and Hannu Kinnunen. The promise of sleep: A multi-sensor approach for accurate sleep stage detection using the oura ring. *Sensors*, 21(13), 2021. 2

[3] Mahmoud Assaf, Aïcha Rizzotti-Kaddouri, and Magdalena Punceva. Sleep detection using physiological signals from a wearable device. In *5th EAI International Conference on IoT Technologies for HealthCare*, pages 23–37. Springer, 2020. 2

[4] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *CoRR*, abs/2004.05150, 2020. 2

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham, 2020. Springer International Publishing. 2

[6] Xie Chen, Yu Wu, Zhenghao Wang, Shujie Liu, and Jinyu Li. Developing real-time streaming transformer transducer for speech recognition on large-scale dataset. *CoRR*, abs/2010.11395, 2020. 2

[7] Taeheum Cho, Unang Sunarya, Minsoo Yeo, Bosun Hwang, Yong Seo Koo, and Cheolsoo Park. Deep-actinet: End-to-end deep learning architecture for automatic sleep-wake detection using wrist actigraphy. *Electronics*, 8(12):1461, 2019. 2, 4

[8] Jishnu Dey, Tanmoy Bhowmik, Saswata Sahoo, and Vijay Narayan Tiwari. Wearable ppg sensor based alertness scoring system. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2422–2425. IEEE, 2017. 2

[9] Shagen Djanian, Anders Bruun, and Thomas Dyhre Nielsen. Sleep classification using consumer sleep technologies and ai: A review of the current landscape. *Sleep Medicine*, 100:390–403, 2022. 2

[10] Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888, 2018. 2

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. 2

[12] Avinava Dubey. Constructing transformers for longer sequences with sparse attention methods. 3

[13] Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. Star-transformer. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1315–1325, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 2

[14] Cyril Joder, Slim Essid, and Gaël Richard. A conditional random field framework for robust and scalable audio-to-score matching. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(8):2385–2397, 2011. 2

[15] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *CoRR*, abs/2106.04554, 2021. 2

[16] Goran Medic, Micheline Wille, and Michiel EH Hemels. Short-and long-term health consequences of sleep disruption. *Nature and science of sleep*, pages 151–161, 2017. 1

[17] Mohammod Abdul Motin, Chandan Karmakar, Marimuthu Palaniswami, and Thomas Penzel. Photoplethysmographic-based automated sleep–wake classification using a support vector machine. *Physiological measurement*, 41(7):075013, 2020. 2

[18] Ryan Hoolbrok Yuki Kotani Larissa Hunt Andrew Leroux Vincent van Hees Vadim Zipunnikov Kathleen Merikangas Michael Milham Alexandre Franco Gregory Kiar. Nathalia Esper, Maggie Demkin. Child mind institute - detect sleep states, 2023. 4

[19] Luis R Peraza, Richard Joules, Yves Dauvilliers, and Robin Wolz. Device agnostic sleep-wake segment classification from wrist-worn accelerometry. In *2020 IEEE international conference on healthcare informatics (ICHI)*, pages 1–3. IEEE, 2020. 2

[20] XiPeng Qiu, TianXiang Sun, YiGe Xu, YunFan Shao, Ning Dai, and XuanJing Huang. Pre-trained models for natural

language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897, 2020. 2

[21] Venkatesh Ramanathan and Harry Wechsler. Phishing detection and impersonated entity discovery using conditional random field and latent dirichlet allocation. *Computers & Security*, 34:123–139, 2013. 2

[22] Kalaivani Sundararajan, Sonja Georgievska, Bart HW Te Lindert, Philip R Gehrman, Jennifer Ramautar, Diego R Mazzotti, Séverine Sabia, Michael N Weedon, Eus JW van Someren, Lars Ridder, et al. Sleep classification from wrist-worn accelerometer data using random forests. *Scientific reports*, 11(1):24, 2021. 2, 4

[23] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. 2

[24] Charles Sutton and Andrew McCallum. An introduction to conditional random fields, 2010. 2

[25] Charles Sutton and Andrew McCallum. An introduction to conditional random fields, 2010. 3

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. 2

[27] Raviteja Vemulapalli, Oncel Tuzel, and Ming-Yu Liu. Deep gaussian conditional random field network: A model-based deep network for discriminative denoising, 2015. 2

[28] Selda Yildiz, Ryan A Opel, Jonathan E Elliott, Jeffrey Kaye, Hung Cao, and Miranda M Lim. Categorizing sleep in older adults with wireless activity monitors using lstm neural networks. In *2019 41st annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, pages 3368–3372. IEEE, 2019. 4

[29] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. *CoRR*, abs/2007.14062, 2020. 2