An Evaluation of SMILE on the TUSZ Corpus

S. McNicholas[1], A. Bryant[1], S. McKenzie[2], M. Desai[2] and J. Picone[1]

1. The Neural Engineering Data Consortium, Temple University, Philadelphia, Pennsylvania, USA
2. University of New Mexico Health Sciences, Albuquerque, New Mexico, USA
{shane.mcnicholas, ashton.bryant, picone}@temple.edu, {SAMcKenzie, MDesai}@salud.unm.edu

An open source system that enables rapid labeling of seizures and other seizure-like types of brain activity known as "ictal-interictal-injury continuum" (IIIC) patterns [1] was recently released by Jing et al. [2]. At the heart of this system, often referred to as SMILE, is a Seizures, Periodic and Rhythmic Continuum patterns Deep Neural Network (SPaRCNet) model. SPaRCNet is a PyTorch model that aims to classify IIIC events with accuracy that exceeds that of clinical experts. According to the authors, SPaRCNet was trained on "50,697 labeled EEG samples from 2,711 patients and 6,095 EEGs that were annotated by physician experts from 18 institutions." The system identifies seizures (SZs) and seizure-like events, known as ictal-interictal-injury continuum (IIIC) patterns, in EEG signals [2]. The system outputs labels for SZs, lateralized and generalized periodic discharges (LPD, GPD) and lateralized and generalized rhythmic delta activity (LRDA, GRDA). From a functional point of view, this system reads an EDF file, performs a classification of IIIC patterns, and presents the user with a GUI that enables rapid annotation of large amounts of data.

In this abstract, we present an evaluation of SMILE on the well-known Temple University Hospital Seizure Corpus (TUSZ) [3]. An extensive evaluation to assess state of the art in seizure classification, known as the Neureka^{TM} 2000 Challenge [4], was conducted using the TUSZ Corpus. The evaluation focused on a simple two-way decision – seizure/no-seizure – and used a scoring metric that combined the sensitivity and false positive (FP) rates to produce an overall figure of merit [5]. Scoring was performed using our open source package Eval EEG [6], and focused on event-based scoring (asynchronous hypotheses that includes a start time, stop time and confidence or probability). The final evaluation metric heavily weighted a system's ability to accurately detect the onsets and offsets of seizure events. To evaluate SMILE on TUSZ, which only does frame-level classification, changes to the SMILE system had to be made. A major contribution of this abstract is to discuss modifications made to SMILE to support this evaluation.

The SMILE system was compared to a real-time ResNet-18 based seizure detection system [7] developed by the authors. This innovative system allows clinicians to continually monitor a patient's EEG data while effectively managing their other clinical duties. Additionally, ResNet is offered as a non-real-time version, frequently utilized in competitive scenarios and benchmarking exercises. In this abstract, we analyze the differences in performance between SMILE and ResNet using the well-calibrated TUSZ dataset as the basis for the comparison.

The complete SMILE system consists of 11 MATLAB scripts. Each script has its purpose, whether that be processing EEG data, running SPaRCNet, or creating the annotation GUI. The first MATLAB script converts EDF data to MAT format. Most of the work of this script is done by EEGLAB, a MATLAB toolbox for processing EEG and other electrophysical data. This script reads each EDF from a directory, converting them into MAT data one at a time. The second step is a simple pre-processing script that resamples to 200 Hz and denoises with a half 40 Hz band-pass filter and a 5 Hz band-stop filter centered at the power-line frequency of 60 Hz. Most pre-processing is done through a function called *fcn_preprocess*, located in a MATLAB file of the same name. The preprocessing script also generates MAT-formatted data that the SPaRCNet model within SMILE requires as input.

The third and final script evaluates the pre-processed data with SPaRCNet. The SMILE system depends on the user to create a Python virtual environment using the Anaconda3 Python distribution to handle the dependencies of SPaRCNet. In addition to packages included with Anaconda, SMILE requires the user to install the hdf5storage, mne, and PyTorch packages to their SMILE virtual environment. With the necessary environment set up, this third MATLAB script activates the virtual environment and calls the runSPaRCNet.py Python script. The runSPaRCNet Python script loads all the pre-processed data and evaluates it with SPaRCNet, which is contained inside the model_1130.pt PyTorch file. The runSPaRCNet script is also responsible for reshaping, montaging, and filtering the EEG data before it is evaluated. The system outputs individual CSV files corresponding to each EDF file. Each CSV file line contains a probability for the six IIIC classes: SZ, LPD, GPD, LRDA, GRDA [2], and Other. Each line is the prediction for a ten-second segment, starting from the beginning of the EDF file. Each entry in the CSV file represents an increment of two seconds forward from the start of the last segment. For example, if the first line is a $0 - 10$s segment, the second line will represent a segment extending from $2 - 12s$.

To facilitate running the modified SMILE system, a Python wrapper was created. This Python wrapper calls each step of the SMILE system and post-processing consecutively, creating the necessary directories along the way. Finally, the wrapper evaluates the results using our open source evaluation system [6]. The Python wrapper efficiently runs a complete evaluation of SMILE and creates the scoring results.

However, SMILE only runs single-threaded out of the box. While this is not a problem with a small data set, large databases such as TUSZ can take several days to complete. The solution is to utilize a high-performance compute cluster, like the NEDC NeuroNix server [8], in tandem with a workload manager, like Slurm [9]. Through a compute cluster and a workload manager, the SMILE wrapper can be run across the hundreds of CPU cores on the NEDC NeuroNix server, completing the database in only a few hours. The TUSZ database was split into slices containing about 25 EDF files, with each slice running as its own, independent process. To make this work, SMILE had to be modified so that its intermediary file locations were no longer hard coded, with each EDF slice storing its output in a unique directory. From here, the slices are regrouped and scored as development, evaluation, and training datasets.

Before the SMILE system can evaluate TUSZ data, the channels that SMILE EDFs use must be appropriately mapped to channels that TUSZ uses. The TUH Corpus contains over 40 unique channel combinations and four different electrode configurations [10]. SMILE's lookup table found in the "channel_mappings.mat" file was modified to include channel labels found in TUSZ data. These labels are mapped in SMILE's preprocess step which prepares TUSZ data for use of SMILE's custom montage.

To do event-based scoring, we had to adapt the SMILE system by implementing a postprocessor that converts frame-level output to event-level output. We adapted the same postprocessor that we use in our open source seizure detection system [7]. For an event to be labeled as a seizure, the seizure probability must exceed a certain threshold, which will be discussed later. From here, an algorithm is deployed to group together consecutive frames, converting the frame-based data into event-based. The final step is to remove any events that do not meet a minimum duration requirement. If the model predicts a seizure with a duration of only $2s$ while surrounded by background events, that event is not a seizure and will be converted to a non-seizure, or background, event. This transforms the synchronous output of SMILE to an asynchronous, or event-based output needed for scoring.

Before delving into the outcomes of the SMILE experiments, it is appropriate to comment on the exclusion of seizures lasting less than $2s$. The significance of seizures lasting a short duration remains a topic of debate within the field of neurology, often with the consensus that short seizures primarily affect young children rather than adults. The decision to omit short duration seizures was made as a strategic compromise between sensitivity and FP rates. Inclusion of seizures under $2s$ substantially inflates the false positive rate while offering minimal improvement in sensitivity, primarily due to the scarcity of children experiencing such short seizures in the TUSZ dataset. For example, it was observed that no children aged $0 - 2$ in TUSZ had encountered seizures lasting less than $2s$. While the precise threshold for defining a "short" seizure can be debated, excluding very brief seizures is a fact of life with most machine learning approaches since they tend to produce a significant number of FPs.

A performance assessment of SMILE was conducted using TUSZ's evaluation and development datasets. TUSZ [3] is divided into three distinct datasets, namely, evaluation, development, and training based on an attempt to balance a number of demographic and metadata features of the corpus. The training dataset is typically used to adjust parameters to optimize performance on the development dataset. The evaluation set is treated as a blind dataset and is exclusively employed for model accuracy evaluation. Parameters are not adjusted to maximize performance on the evaluation set.

It is worth reiterating that SMILE is pre-trained on proprietary data, meaning that the TUSZ training set was not used to train the model. However, our experience with cross-training scenarios for EEG research is that performance across systems is robust [11]. Before the results can be finalized, the seizure threshold and minimum event durations must be tuned to optimize the performance of SMILE. The original seizure threshold was set at 90%, with the minimum seizure and background durations being $20s$ and $40s$, respectively. These parameters were based on optimizations performed in various experiments and challenges [4][7] for the TUSZ Corpus. To find the ideal parameters for SMILE, a grid search approach was used, as summarizes in Table 1. A reasonable operating point for SMILE is shown in green though meaningful performance comparisons to ResNet must consider the entire operating range using a Receiver Operating Characteristic (ROC) [12] or Detection Error Tradeoff (DET) curve [13].

Table 1. Postprocessor Tuning Results

Parameters			Dev			Eval		
seiz_th	**min_bckg**	**min_seiz**	**Sens**	**Spec**	**FPs**	**Sens**	**Spec**	**FPs**
0.90	120	20	22.59	96.96	3.04	34.33	98.50	1.50
0.88	120	20	26.61	96.59	3.41	39.87	98.40	1.60
0.86	120	20	28.21	96.22	3.78	40.94	98.25	1.75
0.80	120	20	34.40	94.75	5.25	44.14	97.62	2.38
0.76	120	20	37.27	93.92	6.08	46.27	97.07	2.93
0.72	120	20	41.63	92.90	7.10	47.76	96.67	3.33
0.70	120	20	42.78	92.63	7.37	49.68	95.98	4.02
0.66	120	20	44.61	91.83	8.17	52.03	95.27	4.73
0.65	120	20	45.18	91.58	8.42	52.24	95.11	4.89
0.64	120	20	45.41	91.33	8.67	52.24	94.67	5.33

Since SMILE doesn’t produce the type of information needed to compute an ROC curve, we use a plot of sensitivity vs. FPs as a proxy. In Table 2 and Figure 1, we show sensitivity as a function of FPs for both SMILE and ResNet on the development data (/dev). In Table 3 and Figure 2, we show the same performance on the evaluation data (/eval). These figures demonstrate that ResNet is competitive with SMILE. Both models boast an impressive sub-1.50 FP rate while still achieving a very respectable sensitivity of 35% on /eval.

These results are more impressive considering that SMILE is not implemented as a real-time system. It performs many passes over the signal. As was learned in the Neureka$^{\text{TM}}$ Challenge [4], non-real-time systems have a distinct advantage in that they can do many types of normalization and postprocessing to optimize performance. ResNet is low latency ($120s$), as it can read and predict data from streams at the accuracy mentioned earlier. Though the classifier operates with less than $1s$ of delay, its postprocessor, previously described in this abstract, introduces delay to reduce FPs. While an impressive system, SMILE has infinite latency and is meant more for offline analysis. ResNet can be further optimized if longer latency was allowed. However, clinical applications require extremely low latency, so a more productive research direction is to further reduce the latency of ResNet.

Table 2. Comparison of Performance on TUSZ /dev

SMILE		ResNet	
FP Rate	**Sensitivity**	**FP Rate**	**Sensitivity**
6.34	38.19	6.26	39.28
7.37	42.78	7.46	42.03
8.67	45.41	8.68	44.5
10.09	46.56	10.38	47.15
11.63	50.92	11.45	49.43

Table 3. Comparison of Performance on TUSZ /eval

SMILE		ResNet	
FP Rate	**Sensitivity**	**FP Rate**	**Sensitivity**
1.34	13.43	1.31	33.48
1.91	42.22	1.91	36.89
2.38	44.14	2.52	40.51
3.33	47.76	3.34	43.28
3.64	49.47	3.71	44.56

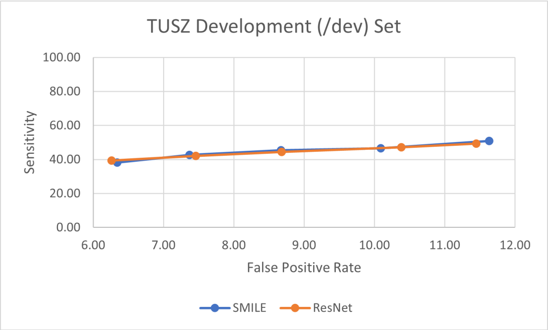


Figure 1. Sensitivity as a Function of FPs on TUSZ /dev

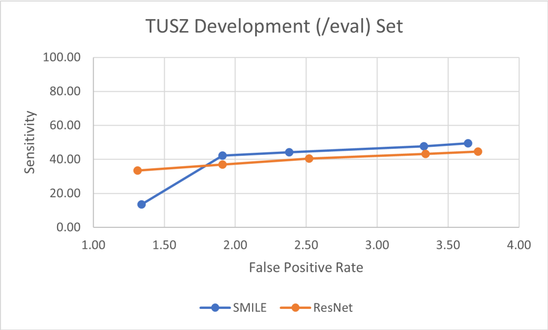


Figure 2. Sensitivity as a Function of FPs on TUSZ /eval

Both systems are accurate and competitive in predicting IIIC patterns. Some users may find the SMILE system a bit unwieldy to use and modify. Not only is ResNet a real-time system with low latency, but it is also a far simpler system to implement and use. ResNet can be run from start to finish in a single script and provides detailed documentation that walks users through its use. Unfortunately, SMILE is a complex system that requires running multiple MATLAB scripts to make predictions for EDF files. Making changes to SMILE is not trivial as documentation about the system is limited.

Acknowledgements

This material is based on work supported by the Temple University College of Engineering's Summer Research Experience for Undergraduates program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Temple University.

References

[1] J. Jing, "Rapid IIC Labeling GUI Multiple EEGs" 2023. doi: *https://github.com/bdsp-core/Rapid_IIIC_Labeling_GUI_MultipleEEGs*.

[2] J. Jing et al., "Development of Expert-Level Classification of Seizures and Rhythmic and Periodic Patterns During EEG Interpretation," *Neurology*, vol. Publish Ahead of Print, Mar. 2023, doi: *10.1212/WNL.0000000000207127*.

[3] V. Shah et al., "The Temple University Hospital Seizure Detection Corpus," Frontiers in Neuroinformatics, vol. 12, pp. 1–6, 2018, doi: 10.3389/fninf.2018.00083.

[4] Y. Roy, R. Iskander, and J. Picone, "The Neureka^{TM} 2020 Epilepsy Challenge," NeuroTechX. 2020. doi: *https://neureka-challenge.com/*.

[5] Y. Roy, R. Iskander, and J. Picone, “The Neureka 2020 Epilepsy Challenge,” in A NeuroTechX Webinar, Philadelphia, Pennsylvania, USA, 2020. doi: *www.isip.piconepress.com/publications/ presentations_misc/2020/neureka_challenge/.*

[6] V. Shah, M. Golmohammadi, I. Obeid, and J. Picone, “Objective Evaluation Metrics for Automatic Classification of EEG Events,” in Biomedical Signal Processing: Innovation and Applications, I. Obeid, I. Selesnick, and J. Picone, Eds., 1st ed.New York City, New York, USA: Springer, 2021, pp. 223–256. doi: *10.1007/978-3-030-67494-6_8*.

[7] V. Khalkhali, N. Shawki, V. Shah, M. Golmohammadi, I. Obeid, and J. Picone, “Low Latency Real-Time Seizure Detection Using Transfer Deep Learning,” in *Proceedings of the IEEE Signal Processing in Medicine and Biology Symposium* (SPMB), I. Obeid, I. Selesnick, and J. Picone, Eds., Philadelphia, Pennsylvania, USA: IEEE, 2021, pp. 1–7. doi: 10.1109/SPMB52430.2021.9672285.

[8] D. Trejo, I. Obeid and J. Picone, "Affordable supercomputing using open source software," *2015 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, Philadelphia, PA, USA, 2015, pp. 1-2, doi: 10.1109/SPMB.2015.7405431.

[9] A. B. Yoo, M. A. Jette, and M. Grondona, “SLURM: Simple Linux Utility for Resource Management,” in *Job Scheduling Strategies for Parallel Processing*, D. Feitelson, L. Rudolph, and U. Schwiegelshohn, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 44–60.

[10] S. Ferrell et al., “The Temple University Hospital EEG Corpus: Electrode Location and Channel Labels,” Temple University, Philadelphia, Pennsylvania, USA, 2020. doi: *www.isip.piconepress.com/publications/reports/2020/tuh_eeg/electrodes*.

[11] V. Shah, *Improved Segmentation for Automated Seizure Detection Using Channel-Dependent Posteriors*, PhD Dissertation, Department of Electrical and Computer Engineering, Temple University, Philadelphia, Pennsylvania, USA, 2021. url: *www.isip.piconepress.com/publications/ phd_dissertations/2021/seizure_segmentation/.*

[12] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*. New York City, New York, USA: John Wiley & Sons, Inc., 1965. doi: *https://books.google.com/books/about/ Principles_of_communication_engineering.html?id=4ORSAAAAMAAJ.*

[13] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, “The DET curve in assessment of detection task performance,” in *Proceedings of the European Conference on Speech Communication and Technology* (Eurospeech), Rhodes, Greece, 1997, pp. 1895–1898. doi: *http://www.isca-speech.org/archive/eurospeech_1997/e97_1895.html*.

NEURAL ENGINEERING
DATA CONSORTIUM
www.nedcdata.org

An Evaluation of SMILE on the TUSZ Corpus

College of Engineering
Temple University

S. McNicholas, A. Bryant and J. Picone
The Neural Engineering Data Consortium, Temple University

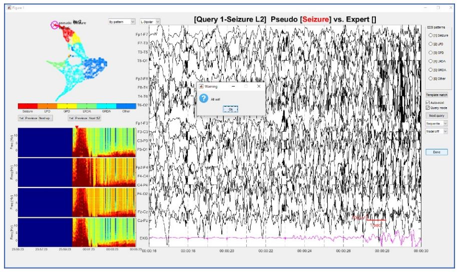
S. McKenzie and M. Desai
University of New Mexico Health Sciences

Abstract

- An open source system that enables rapid labeling of seizures and other seizure-like types of brain activity ("ictal-interictal-injury continuum" or IIIC), known as SMILE, was recently released.
- SMILE uses a PyTorch-based "SPaRCNet" model trained on roughly 50,000 annotated EEG samples.
- An alternative system, NEDC EEG ResNet, is a residual network-based seizure prediction system that offers real-time and non-real-time versions.
- TUSZ, an open-source, annotated seizure corpus, is used to evaluate performance.
- To assess SMILE's performance accurately, adjustments were made to the system's interface, transforming frame-level output into event-based.
- After modifications, SMILE performed very similarly to that of ResNet. Both systems featured a variety of respectable false-positive rates.
- Both systems are good options for IIIC prediction with each having their own strengths.

Rapid Labeling of EEG Events (SMILE)

- Crafted as an all-encompassing application, SMILE includes event detection and incorporates a user-friendly GUI for labeling IIIC patterns.
- SMILE uses a series of MATLAB scripts that encapsulate each component of the SMILE system.
- SMILE resamples every EEG to 200 Hz and denoises with a 40 Hz half band-pass filter and a 5 Hz band-stop filter centered at 60 Hz.
- SMILE uses an in-house trained seizure prediction model named SPaRCNet, which is trained on 50,697 annotated EEG segments. Each time frame results in a prediction from the following six classes: Seizure, LPD, GPD, LRDA, GRDA, and "Other."
- SPaRCNet outputs a CSV file that contains overlapping frame-level predictions for each class, with the first frame covering 0-10s, the second 2-12s, the third 4-14s, and so forth.
- The main focus for SMILE was a tool that supports rapid annotation of EEG data.
- SMILE does not include the capability to train new PyTorch models using user-provided EDF files. Only a decoder and an annotation tool are provided.
- A typical SMILE screen is shown below:



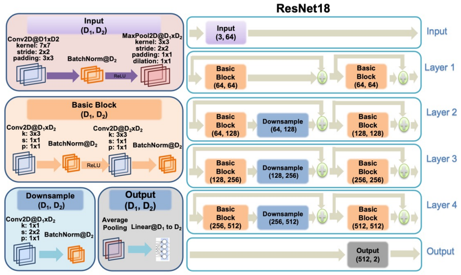
[Query 1-Seizure L2] Pseudo [Seizure] vs. Expert []

All set!

☑ Auto-next
☑ Query mode

A ResNet-18 Based Detection System

- A ResNet-18-based real-time system:

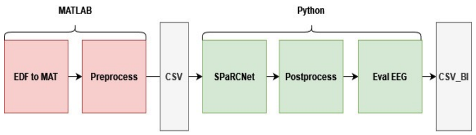


- ResNet features both real-time and non-real-time implementations with a very small difference in performance between them.
- Here we compare the non-real-time system which has slightly better postprocessing performance but also has more latency.
- This system, implemented in Python, was designed to facilitate processing large amounts of data and use multithreading. It does not include an annotation GUI.
- ResNet first makes event-based predictions (probabilities for each of the six IIIC classes):

```
# version = csv_v1.0.0
# bname = aaaaagbf_s007_t007
# duration = 1046.0 secs
# montage_file = $NEDC_NFC/lib/nedc_eas_default_montage.txt
#
channel,start_time,stop_time,label,confidence
TERM,0.0000,312.3200,bckg,1.0000
TERM,312.3200,460.8000,seiz,1.0000
TERM,460.8000,1046.0000,bckg,1.0000
```

Modifications to SMILE

- Modifications had to be made to the system to be appropriately evaluated using TUSZ and our standardized scoring tools.
- Since we were only interested in SMILE's prediction capabilities, the annotation GUI was ignored.
- To simplify the sequence of MATLAB scripts, Python wrappers were created to encapsulate each MATLAB script in an easy-to-use interface.
- To convert SMILE's frame-level classifications to event-based, a Python postprocessor was written to convert the SPaRCNet frame-level CSV predictions to an event-based CSV_BI file.
- Finally, the proper channels for TUSZ EDF files had to be mapped to what SMILE expects.
- As a note, Python wrappers were used to parallelize the modified SMILE system and speed up evaluation.
- The modified SMILE processing pipeline is shown below. The output is an event-based annotation stored in a CSV file that facilitates scoring:



Postprocessor Tuning

- We integrated the ResNet ResNet processor to convert SMILE's frame-based output to event-based output. This simple heuristic postprocessor groups similarly labeled frames into a single event.
- Before a complete SMILE evaluation can occur on TUSZ, the postprocessor's parameters must be tuned to ensure maximum performance.
- The postprocessor has three parameters: (1) a minimum probability for a seizure event, (2) a minimum duration for an event to be considered a seizure, and (3) a minimum background duration.
- Performance was optimized with respect to sensitivity, specificity, and false positive rate:

Table 1. Postprocessor Tuning Results

Parameters			Dev			Eval		
seiz_th	min_bckg	min_seiz	Sens	Spec	FPs	Sens	Spec	FPs
0.90	120	20	22.59	96.96	3.04	34.33	98.50	1.50
0.88	120	20	26.61	96.59	3.41	39.87	98.40	1.60
0.86	120	20	28.21	96.22	3.78	40.94	98.25	1.75
0.80	120	20	34.40	94.75	5.25	44.14	97.62	2.38
0.76	120	20	37.27	93.92	6.08	46.27	97.07	2.93
0.72	120	20	41.63	92.90	7.10	47.76	96.67	3.33
0.70	120	20	42.78	92.63	7.37	49.68	95.98	4.02
0.66	120	20	44.61	91.83	8.17	52.03	95.27	4.73
0.65	120	20	45.18	91.58	8.42	52.24	95.11	4.89
0.64	120	20	45.41	91.33	8.67	52.24	94.67	5.33

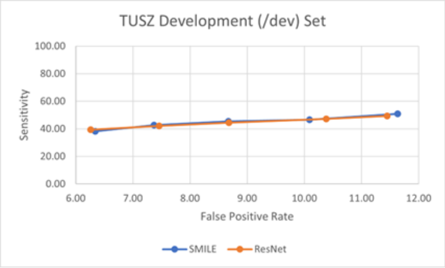
- The most effective parameter combination was a seizure threshold of 70%, a min. seizure dur. of 20 secs, and a min. background dur. of 120 secs.
- These parameters can be easily changed from an external parameter file.
- Maintaining a very low false positive rate while maximizing sensitivity is very important in clinical applications.

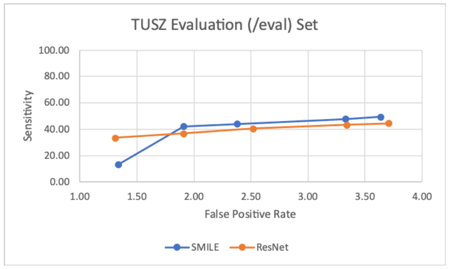
Comments about Usability

- In addition to performance, the usability of SMILE and ResNet should be considered when comparing the two systems.
- Both systems provide similar functionality in decoding and making predictions on input EDF files.
- ResNet allows training a module on annotated data, while SMILE does not.
- SMILE includes a built-in annotation GUI. SMILE is built on MATLAB scripts requiring a paid license, while ResNet is built on open-source tools.
- ResNet features a simple, shell script-based command line interface that can be run from start to finish using a single script.
- SMILE requires running multiple MATLAB scripts in sequence without much of an interactive interface.
- Unfortunately, SMILE does not include much documentation, and modifying the system can be difficult and complex. ResNet is heavily documented, and modifications to the system are encouraged.
- ResNet installation is simple while SMILE requires the installation of the 3rd-party EEGLAB MATLAB toolbox.
- ResNet features a real-time decoder that can decode EEG streams with low-latency, while SMILE only features a non-real-time decoder.

Performance Comparison

- ResNet and SMILE were evaluated using sensitivity as a function of the false positive rate:





- On the TUSZ /dev and /eval sets, performance was comparable.
- SMILE's sensitivity reduces significantly on the eval set for low false positive rates.
- ResNet was computationally faster than SMILE due to its optimized EDF file I/O in Python.
- Both systems boast an impressive sub-1.50 FP rate while achieving a respectable sensitivity of 35%.
- The low false positive rate region of these plots is of great interest in clinical applications.

Summary

- Both systems provide competitive predictions on IIIC patterns. Differences in performance are not statistically significant except when the FP rate is low.
- Some users may find the SMILE system to be a bit difficult to use and modify for research purposes.
- ResNet is a well-documented and easy-to-use system with real-time and non-real-time flavors.

Acknowledgements

- This material is based on work supported by the Temple University College of Engineering's Summer Research Experience for Undergraduates program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Temple University.

Availability

- SMILE: https://github.com/bdsp-core/Rapid_IIIC_Labeling_GUI_MultipleEEGs
- ResNET: https://isip.piconepress.com/projects/nsf_pfi_tt
- NEDC_EAS: www.isip.piconepress.com/projects/tuh_eeg