A Scalable Automated Diagnostic Feature Extraction System for EEGs

Prakhar Agrawal*†, Divya Bhargavi*†, Gokul Krishna G*†, Xiao Han*†, Neha Tevathia*†,
Abbie M. Popa†, Nicholas Ross†, Diane Myung-kyung Woodbridge†, Barbie Zimmerman-Bier‡, William J. Bosl†§
Harvard Medical School, Boston MA§; Data Science Program, University of San Francisco, San Francisco, CA†
{pagrawal4,vbhargavi,gg,xhan20,ntevathia,apopa,ncross,dwoodbridge,wjbosl}@usfca.edu
New Jersey Medical School (NJMS), Neurology, Newark, NJ‡; zimmerba@rwjms.rutgers.edu

***Abstract*—Researchers using Electroencephalograms ("EEGs") to diagnose clinical outcomes often run into computational complexity problems. In particular, extracting complex, sometimes nonlinear, features from a large number of time-series often require large amounts of processing time. In this paper we describe a distributed system that leverages modern cloud-based technologies and tools and demonstrate that it can effectively, and efficiently, undertake clinical research. Specifically we compare three types of clusters, showing their relative costs (in both time and money) to develop a distributed machine learning pipeline for predicting gestation time based on features extracted from these EEGs.**

***Keywords*-Electroencephalography, EEG, Distributed Processing, Distributed Database, NoSQL, Cloud Computing, Machine Learning**

I. Introduction and Background

Electroencephalograms (EEGs) are recorded using non-invasive electrodes placed on the scalp of participants to monitor the electrical activity of the brain. Historically, EEGs have been used to diagnose a variety of different neurological conditions, including hearing loss [1], seizure disorders [2], and Tourette's Syndrome [3]. This diagnostic process is done by board-certified experts, such as neurophysiologists or neurologists, by visual inspection of the EEG time series. With recent advances in machine learning approaches to time-series analysis, these experts may be augmented by automated expert systems to support clinical decision-making.

EEGs are not currently used to screen for or diagnose psychiatric or neurodevelopmental disorders. However, recent research suggests that some conditions may exhibit functional neural differences in EEG patterns that can be detected. These include Autism Spectrum Disorder (ASD) [4], depression [5], schizophrenia [6] and attention deficit hyperactivity disorder (ADHD) [7]. Thus far, these differences have not proven reliable enough for use as clinical tools for risk assessment or diagnosis. Most previous EEG studies have relied on linear measures such as spectral power density [8] for resting state analysis or amplitude and latency changes for detecting event-related potentials (ERPs) in response to stimuli such as images of faces or incongruent sounds [9]. ERPs require more training for administration and are time-consuming because they require many trials with a stimulus to be averaged together. Physiological signals may exhibit chaotic characteristics and important information in the time series will not be detected by linear measures [10]. Nonlinear measures may provide additional information about the dynamics of neural activity. This may be particularly useful for participants who are not able to complete a task associated with ERPs, either due to age or cognitive ability. Finally, nonlinear features are promising as digital biomarkers that can be computed from biological signals, and so are similarly promising when examining EEGs [10]. Our work builds on the foundation of linear EEG analysis, but expands on historical EEG analysis with the inclusion of nonlinear features.

Researchers frequently run into issues since computing features from an EEG time-series is computationally costly. EEG data is acquired at sampling rates of 250-1000 Hz, which produces a high volume of data for short recording periods. Most EEG systems used in clinical or research settings record from many sensors spaced evenly on the scalp. Thus, a single EEG recording contains a multitude of time series, one from each sensor, and each of these are frequently decomposed into frequency bands. A single EEG recording could easily contain hundreds of times series to analyze, depending on the number of channels (up to 128) and frequency bands (usually 5 - 12). Finally, algorithms for computing nonlinear features are often computationally intensive, further slowing down the research process. Our study documents a system that takes in multiple time series contained within an EEG and processes them via a distributed, cloud-based cluster to generate specific features and then efficiently deploys a machine learning model using the output features. We applied machine learning algorithms to the extracted features in order to predict a clinical characteristic of an infant.

Our work builds on a previous study where nonlinear and linear features were extracted from 188 infants, 99 of whom were at high risk for autism and 89 who were low risk controls. EEGs were collected for these infants at 3, 6, 9, 12, 18, and 24 months months of age. EEG measures, both linear and nonlinear, were used to predict an ASD diagnosis that was determined clinically when the participants reached 36 months of age, using support vector machines. The resulting models could detect later development of ASD with

* These authors contributed equally.

sensitivity and specificity ranging from 82 - 100%. [4].

In this study we apply a similar method to analyzing EEG data from premature infants born at different gestational ages. A predictive model was built using linear and nonlinear EEG features to determine preterm level from EEG alone. Our data set consists of 47 EEG files, representing 34 infants. For each EEG, eight channels were recorded (Fp1, Fp2, C3, C4, O1, O2, T7, T8) at a sampling frequency of 500 Hz. The EEG time-series had an average length of 8.12 minutes (minimum 2 seconds, maximum 14.2 minutes) and were recorded at 6, 12 and 24 months post-birth.

After completing our data processing system we built a series of machine learning models to validate its research potential. We compare three classifiers (one-vs-all logistic regression, decision trees, and random forests) to detect if an infant, based on their EEG features, was extremely, very, or moderately preterm. Since clinical data were not available at the time of this exercise, preterm level serves as an important proof of concept that our system works for processing EEGs for machine learning classification. Importantly, the level of prematurity may indicate later risk for negative cognitive outcomes, including ASD or developmental delay [11], so classifying this outcome validates our method for broader risk assessments in future studies. We find that a random forest model achieves the highest prediction accuracy, though it takes almost ten times as long to train as our fastest model, the decision tree. The one-vs-all classifier had the lowest accuracy as well as the largest training time.

The rest of this paper is organized as follows: Section 2 contains a system overview while Section 3 contains specifics of different computing configurations that were evaluated. Section 4 contains our machine learning results and we end with a short discussion.

II. System Overview

Our system is designed with clinicians and neuroscientists in mind and breaks up simple operations (loading and recording metadata) and more complex operations (feature extraction) to facilitate ease of use. The system thus consists of a front-end system which allows researchers to easily upload EEG files as well as document metadata about the participants from whom EEGs were recorded. Once the EEG records are uploaded, our back-end systems process the files in batch, extracting features and recording them for future use. While this system has a number of parts, here we focus on the back-end processes, as described in Figure 1.

A. Data Storage and Preprocessing

Once a user uploads an EEG data file and records the metadata using the front-end interface (Python Flask web application hosted on Amazon Web Service [12] Elastic Compute Cloud (AWS EC2)) the files are validated (making sure that they are readable) and placed in an Amazon Simple Storage Service (S3) bucket. Amazon S3 offers secured, scalable and high-speed online object storage. Our system currently permits EEG files recorded in the European Data Format ("EDF"), though the system will be extended to accept additional EEG file formats.

Table I: Instance Specification for MongoDB - All instances have 1 CPU.

Server	Number	Size	Memory	Cost (hourly)
Storage	3	t2.medium	4 GB	.4176
Config	3	t2.micro	1 GB	.0348
Routing	1	t2.micro	1 GB	.0116

The server then detects new files in the S3 bucket and downloads those files to another server which extracts the raw data using the PyEDFlib package which can encode and decode signals [13]. Each EDF file contains time-series data from eight channels as well as metadata including the sampling rate. At this stage the file undergoes error checking and additional data validation to make sure that it can be processed to extract features and build a machine learning model. Upon completion of these checks, both the metadata and time series data extracted from the EDF file are transferred and stored in MongoDB [14]. The system also creates a series of log files that include details of the process.

MongoDB is a distributed schemaless database that stores data in JSON document format. MongoDB divides and stores data in multiple shards to provide high scalabilty. For each shard, MongoDB maintains a primary node and secondary nodes that have duplicated data from primary nodes to enhance reliability in case the primary node fails. Further, MongoDB is schemaless, which makes it especially useful for storing data with different structures, such as EEG time series (with different formats and lengths), metadata, and computed features.

To store our data, we launched 13 AWS EC2 instances to build the MongoDB distributed data system (specifically, the Amazon Linux 2 AMI (HVM) and MongoDB 4.0.3). Instance details are shown in Table I. Horizontal sharding was used to improve performance of our parallelized, downstream feature processors while providing redundancy that protects the system in the event of node failures.

Once copied to MongoDB, the files are ready for feature extraction and a feature-extraction-tracking log table is created in the database. The purpose of this table is to keep track of which features have been successfully extracted from the EEG files. Infant data, such as gestational age, preterm level, gender and clinical data are also parsed and stored within the MongoDB database. The entire pre-processing and data storage step takes around 5 seconds for an average-length EEG file from this dataset. At this stage, the EDF file has been broken down into time-series data from each channel and is ready to be processed to extract features.

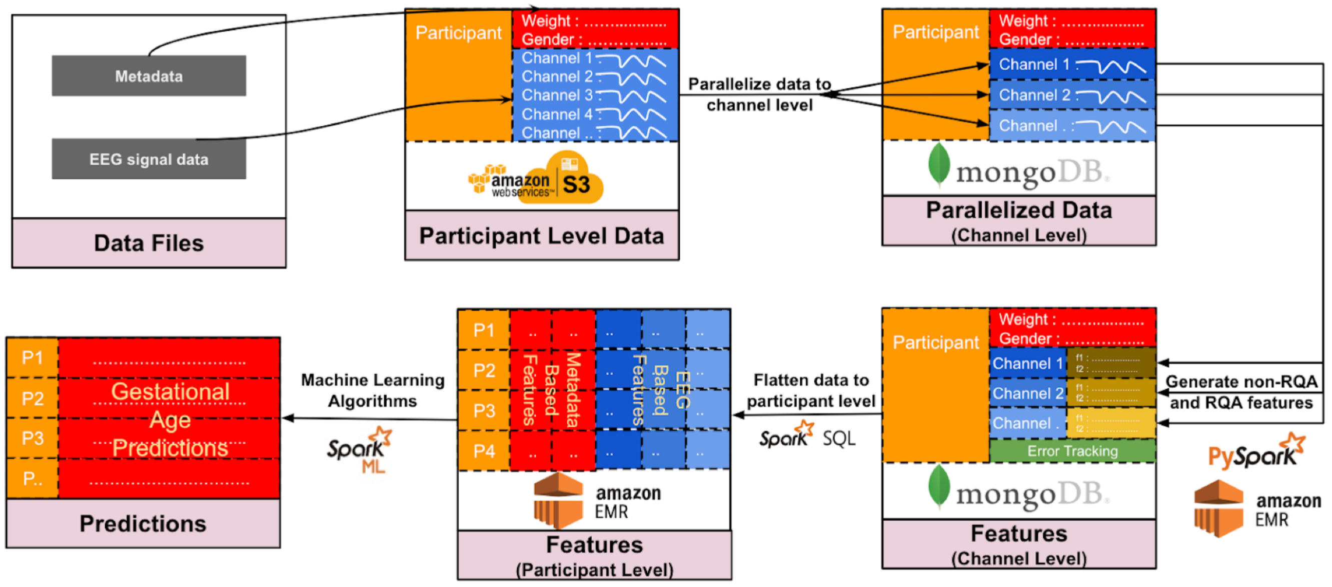


Figure 1: Data Processing Architecture

Table II: Feature Descriptions, with RQA features in bold.

Feature Name	Description
Power	Measure of the signal's intensity
Sample Entropy	Measure of self-similarity of the signal
Hurst Exponent	Measure of the long statistical dependencies in the data
De-trended Fluctuation Analysis	Measure of statistical self-affinity of the signal
Lyapunov Exponent	Rate of separation of two infinitesimally close trajectories
Recurrence Rate	Probability that the system state recurs in a finite time

B. *Feature Extraction from Time Series*

Features are computed on the second 30-second segment of data from each time-series. The very beginning of an EEG signal often has some noise as the electrodes are settling, so the second 30 second segment should avoid this issue. For time-series under 60 total seconds in length, the first 30 seconds were used so that these infants would not need to be excluded. 30 second segments are sufficient to detect the features we are interested in, in fact, often even shorter segments have been used effectively [15]. That said, an area of future research is to select the segment for analysis in a more sophisticated manner, perhaps by finding the section of the time-series with the highest signal to noise ratio.

We compute two types of features: Recurrence Quantification Analysis ("RQA") and non-RQA features. RQA features are used to quantify repetition in highly nonlinear and dynamical systems. Importantly for clinical researchers, RQA features entail orders of magnitude more computational complexity than computing linear features and thus can be computationally prohibitive to evaluate. Table II includes the full list of features we compute and a short explanation, with Recurrence Rate being the only RQA feature computed for this exercise.

We use the PySpark API [16], which leverages the Apache Spark architecture, to control operations across the EMR cluster. Spark adopts MapReduce concepts to divide data into smaller chunks across different nodes, and subsequently maps and processes a task such as filtering and sorting, in parallel. The output of a mapped task becomes the input of a reduce operation, which performs a summary operation. This highly-effective model allows users to design programs with successive Map and Reduce operations, and is a popular and powerful programming paradigm.

Spark runs 100 times faster in memory and 10 times faster on disk than the original Hadoop MapReduce [17]. Spark provides an efficient means of distributing complex computational tasks to worker nodes in a fault-tolerant framework using a dependency graph of the planned computations. If a worker nodes fails, Spark supports re-computation of the lost partition from the original computations and thus recovers any lost data. This is especially useful for generating features out of EEG signals considering the number of features to be computed and the difficulty of computation. The incorporation of a tracking log in this algorithm allows for real-time monitoring of the calculation process.

Feature extraction is parallelized by channel and by frequency band using Spark [18]. By parallelizing the feature extraction process by channels and frequency bands, the system's extraction time is significantly decreased. For each EEG file there are approximately 72 time series from eight channels by nine bands, which are processed in parallel.

The pseudo code in Algorithm 1 documents the specific procedures within the feature extraction process. This algorithm is developed in Spark and deployed via an AWS Elastic MapReduce ("EMR") cluster [12]. AWS EMR provides a Hadoop framework supporting efficient parallel processing of large data sets in a distributed environment across multiple EC2 instances.

Algorithm 1 Feature Extraction

```
Data: Raw Data (Channel Level), Tracking Log
Result: Extracted RQA and Non-RQA Features
while (log shows new or failed channels (attempts <5)) do
    update each channel's status in log to "processing"
    for (channel in channels) do
        if (channel duration <60 seconds) then
            select first 30 seconds of time series
        else
            select 30-60 seconds of the time series
        end
        extract frequency bands and labels for selected data
        for each band do
            compute RQA and Non-RQA Features
        end
    end
    Update log for failed and successful channels
    Write features for successful channels
    Partial write for failed channels
end
```

III. Feature Extraction Cluster Experiments

While the previous section documented how we interacted with our cluster, this section details how we determined our optimal cluster configuration. The primary constraint that we identified when experimenting with cluster configuration was caused by the nonlinear RQA features, which were nearly un-computable on a personal computer without a GPU instance. In particular, computing the RQA features for the full dataset of 47 EEGs (3,384 time series) required 425 minutes and 35 seconds using a c4.xlarge machine and only 5 minutes 20 seconds using a GPU enabled p2.xlarge. Given that the GPU enabled box was 80 times faster but only 5 times costlier, we elected to use the GPU enabled box when computing RQA features.

For the non-RQA features, we tested three different sized clusters to determine the optimal configuration; Table III details the results. All three clusters consisted of five instances and the total cost of running the job was between almost $4 dollars and $2.66. Given our particular requirements and estimates of the number of EEG files that we expect to process, we decided to use the c4.xlarge because, while it was not the fastest, it struck a good balance between speed and cost.

As an example, consider the paper [19]. In this paper the authors used 3,016 EEGs from the Temple University Hospital EEG Corpus of over 12,000 EEGs [20]. Running 3,016 EEGs on the c4.2xlarge instance would take 96 hours and $241 using our processing system. On the c4.xlarge it would take 128 hours and $160. Though this is a long period of time, we estimate that on a home computer the 3,016 EEGs would take weeks to run. We believe that allowing large EEG datasets to be processed in a more timely and efficient manner will encourage their use. Large datasets will produce findings that are more robust and reflect subtler effects.

IV. Machine Learning Algorithm

While we showed that the system we described was able to process the EEG files and generate features, we also wanted to demonstrate its usefulness for clinical research purposes. To that end we took our sample of EEG files, which contained information on infants, extracted features from those files using this system, and trained a series of classifiers on those resulting features to estimate the gestational age of the infant, according to the following buckets:

1) Extremely Preterm (<28 weeks)
2) Very Preterm (28 to 32 weeks)
3) Moderate / Late Preterm (32 to 37 weeks)

Eleven of the forty-seven infants had multiple EEG recordings. For these eleven we took the latest EEG that was more than sixty seconds in length. Additionally, we removed ten infants before beginning modeling. Specifically, we removed two infants who had no specified gestational age and eight with erroneous output values.

Gestational age is not necessarily a useful clinical outcome to estimate, since the physician will know when the infant was born. However, given that clinical data were not available at the time of processing, this analysis serves as a proof of concept for our system. Preterm birth is associated with higher risk for poor clinical outcomes [11], so many of the features that are important in these machine learning models will also be important in a more clinically oriented model, which could assess risk for development of ASD or other conditions.

We compared three algorithms from the PySpark Machine Learning library, MLlib: logistic regression, decision trees, and random forests. Our baseline model is a logistic regression one-vs-all model. This technique works by estimating a binary logistic regression classification for each type. In our case this generates three logistic regression models. From these three models we predict a data point by calculating

Table III: Running times, non-RQA features, by cluster type

Type	CPU	Memory	Time	Per Hour	Total
c4.xlarge	4	7.5 GB	2 hr 7 min	$1.255	$2.656
c4.2xlarge	8	15 GB	1 hr 35 min	$2.515	$3.982
m4.xlarge	4	16 GB	2 hr 25 min	$1.300	$3.141

the estimated inclusion likelihood from all three models and choosing the one with the highest score.

Our second model, a decision tree, performs recursive binary partitioning of the feature space and makes the same prediction for each bottom partition [21]. It has a tree-like structure: each leaf node stands for a decision rule and each branch represents the decision made in the previous node.

Finally, a random forest works by creating decision trees based off of a bootstrapped set of samples and features [22]. The random forest algorithm takes this ensemble of decision trees and then aggregates them, via a voting rule, in order to predict the type of an observation. The random forest model generally shows improved performance versus the single decision tree model [22].

For all models, we estimated the model using a randomly selected "hold out" or "test" set and only training our model on 80% of the data. All model fit numbers, found in the next section, are computed on this hold-out set because it did not influence the initial model training.

While classification problems of these types tend to use cross-validation, our sample size was too small to do this meaningfully and thus we used the Spark MLlib default hyper-parameters rather than completing hyper-parameter tuning. The next section details our results.

A. *Model Results*

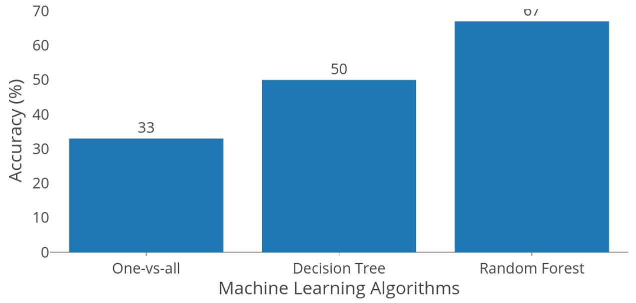
Figure 2a demonstrates the results of our models. In this table accuracy is calculated on our 20% hold out set while the reported times are measured based on how long it took to train each particular model 2b.

An important caveat to the above results is that our model is not binary, but instead requires correctly identifying one of *three* labels for a data set. In other words, the 33% accuracy is not worse than a coin flip because we are trying to predict one of three outcomes.

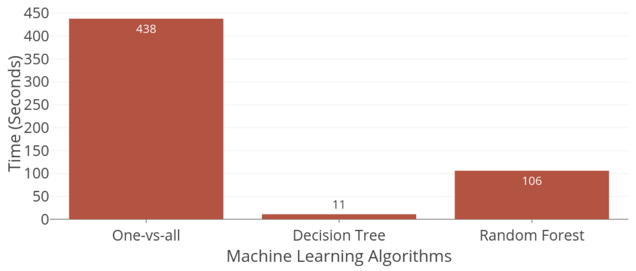
The random forest model out-performed the other two, only taking 106 seconds to train. The features of highest importance from the random forest were: Lyapunov Exponent, Sample Entropy and Power across various bands.

V. Discussion

In this research, we designed and developed a system for automatically extracting features from EEG recordings to detect an infant's gestational age for clinical research purposes. We first demonstrate that this system processes features much more rapidly than a desktop based system



(a) Accuracy of different machine learning algorithms



(b) Execution time of different machine learning algorithms

Figure 2: Accuracy and execution time of applied machine learning algorithms

while remaining affordable. Particularly for the computationally intensive RQA features using our cloud based system could make a researcher's computation time 80 times faster.

We secondly demonstrate that this system can provide an end-to-end solution, starting with a raw EEG file and developing a machine learning based classification. While the accuracy of our models was lower than illustrated in the previous paper predicting ASD, we note that we were working with a small sample size (37 infants divided into 3 groups). The fact that we achieved 67% accuracy, while preliminary, is encouraging for future work on larger samples.

For clinical neuroscience researchers, the presented system is incredibly useful due to the computational complexities around processing EEG files. First, many clinical researchers lack access to the computational power to process these high frequency multi-channel data in a rapid, efficient manner. Second, some clinical researchers do not have exposure to nonlinear features from EEG, nor do they have the time or skill to compute these features from EEG, despite their promise as diagnostic or screening measures. Indeed, in our best-performing machine learning model, the random forest, the two most important features (Lyapunov Exponent and Sample Entropy) were both nonlinear features. By creating a system which uses off-the-shelf software components and cloud-based tools, we hope that we are encouraging researchers in this field to look beyond single-node desktop based tools in order to increase their efficiency and lower their costs.

Finally, this system allows for processing large volumes of high frequency data. Many EEG studies currently have relatively small sample sizes, and these small sample sizes may result in the study failing to detect real, but small effects. Additionally, smaller sample sizes may reduce the generalizability of the results. We hope by providing a means to process large volumes of data, larger studies will be more feasible for EEG researchers.

References

[1] M. P. Paulraj, K. Subramaniam, S. B. Yaccob, A. H. B. Adom, and C. R. Hema, "Auditory Evoked Potential Response and Hearing Loss: A Review," *The Open Biomedical Engineering Journal*, vol. 9, no. 1, pp. 17–24, feb 2015. [Online]. Available: https://openbiomedicalengineeringjournal.com/VOLUME/9/PAGE/17/

[2] E. Niedermeyer and F. H. Lopes da Silva, *Electroencephalography : basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2005. [Online]. Available: https://books.google.com/books/about/Electroencephalography.html?id=tndqYGPHQdEC

[3] F. R. Volkmar, J. F. Leckman, J. Detlor, D. F. Harcherik, J. W. Prichard, B. A. Shaywitz, and D. J. Cohen, "EEG Abnormalities in Tourette's Syndrome," *Journal of the American Academy of Child Psychiatry*, vol. 23, no. 3, pp. 352–353, 1984. [Online]. Available: http://dx.doi.org/10.1016/S0002-7138(09)60516-1

[4] W. J. Bosl, H. Tager-Flusberg, and C. A. Nelson, "EEG Analytics for Early Detection of Autism Spectrum Disorder: A data-driven approach," *Scientific Reports*, vol. 8, no. 1, p. 6828, dec 2018. [Online]. Available: http://www.nature.com/articles/s41598-018-24318-x

[5] J. J. B. Allen and M. X. Cohen, "Deconstructing the Resting State: Exploring the Temporal Dynamics of Frontal Alpha Asymmetry as an Endophenotype for Depression," *Frontiers in Human Neuroscience*, vol. 4, no. December, pp. 1–14, 2010. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnhum.2010.00232/abstract

[6] C. J. Leonard, S. T. Kaiser, B. M. Robinson, E. S. Kappenman, B. Hahn, J. M. Gold, and S. J. Luck, "Toward the neural mechanisms of reduced working memory capacity in schizophrenia," *Cerebral Cortex*, vol. 23, no. July, pp. 1582–1592, 2013.

[7] A. Mazaheri, S. Coffey-Corina, G. R. Mangun, E. M. Bekker, A. S. Berry, and B. a. Corbett, "Functional disconnection of frontal cortex and visual cortex in attention-deficit/hyperactivity disorder." *Biological psychiatry*, vol. 67, no. 7, pp. 617–23, apr 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20060100

[8] M. X. Cohen, *Analyzing Neural Time Series Data: Theory and Practice*. MIT Press, 2014.

[9] S. J. Luck, *An Introduction to the Event-Related Potential Technique*, 2nd ed. Cambridge, MA: Massachusetts Institute of Technology, 2014.

[10] M. Costa, A. L. Goldberger, and C. K. Peng, "Multiscale entropy analysis of biological signals," *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, vol. 71, no. 2, pp. 1–18, 2005.

[11] S. Johnson and N. Marlow, "Preterm Birth and Childhood Psychiatric Disorders," *Pediatric Research 2011 69:5, Part 2 of 2*, may 2011.

[12] Amazon Web Service. (2019) Amazon. [Online]. Available: https://aws.amazon.com

[13] PyEDFlib. (2019) Pyedflib -edf/bdf toolbox in python. [Online]. Available: https://pyedflib.readthedocs.io/en/latest/

[14] MongoDB. (2019) Mongodb for giant ideas. [Online]. Available: https://www.mongodb.com/

[15] T. M. Heunis, C. Aldrich, and P. J. de Vries, "Recent Advances in Resting-State Electroencephalography Biomarkers for Autism Spectrum DisorderA Review of Methodological and Clinical Challenges," *Pediatric Neurology*, vol. 61, pp. 28–37, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.pediatrneurol.2016.03.010

[16] Apache Spark, "Apache spark: Lightning-fast cluster computing," 2019. [Online]. Available: http://spark.apache.org

[17] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[18] J. S. Walker, *A primer on wavelets and their scientific applications*, 2nd ed. Chapman & Hall/CRC, 2008.

[19] R. Schirrmeister, L. Gemein, K. Eggensperger, F. Hutter, and T. Ball, "Deep learning with convolutional neural networks for decoding and visualization of EEG pathology," *2017 IEEE Signal Processing in Medicine and Biology Symposium, SPMB 2017 - Proceedings*, vol. 2018-Janua, pp. 1–7, 2018.

[20] I. Obeid and J. Picone, "The Temple University Hospital EEG Data Corpus." *Frontiers in neuroscience*, vol. 10, p. 196, 2016. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/27242402

[21] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, Mar 1986. [Online]. Available: https://doi.org/10.1007/BF00116251

[22] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324