Hadoop for EEG Storage and Processing: a Feasibility Study

Ghita Berrada, Maurice van Keulen, and Mena B. Habib

University of Twente, The Netherlands
{g.berrada,m.vankeulen,m.badiehhabibmorgan}@utwente.nl

Abstract. Lots of heterogeneous complex data are collected for diagnosis purposes. Such data should be shared between all caregivers and, often at least partly automatically processed, due to its complexity, for its full potential to be harnessed. This paper is a feasibility study that assesses the potential of Hadoop as a medical data storage and processing platform using EEGs as example of medical data.

Keywords: EEG,Hadoop, medical data storage and processing

1 Introduction

The diagnosis process often involves multiple clinicians/specialists and a large number of ordered tests. As a result, huge amounts of heterogeneous data are gathered and scattered in many locations (or islands of data). Table 1 shows the scale of data produced in and spread across the healthcare system. To further compound the problem, different locations often use non-interoperable systems and file formats, if the data is indeed digitized. A McKinsey Global Institute (MGI) report on the US healthcare system ([1]) shows that up to 30% of data that includes medical records, laboratory and surgery reports, is not digitized and that the video and monitor feeds that make up most of the clinical data produced are not stored but used real time. Such a setting makes it hard for caregivers to access a patient's full history and get a full picture of his/her condition. As it stands, needless tests may be ordered and diagnoses delayed and/or missed, not to mention data security more easily breached. The prevalence of misdiagnoses is estimated to be up to 15% in most areas of medicine ([2]). And a study of physician-reported diagnosis errors ([3]) finds most cases are due to testing (44%) or clinician assessment errors (32%).
A case from The Washington Post exposes all those issues ([4]). A patient struggling with depression is diagnosed with a meningioma [1], unrelated with the patient's depression and not in need of monitoring, according to the attending clinician at the time. Four years, many moves across US states and many consultations (with other clinicians) later, and with her condition steadily worsening, the patient is hospitalized and the meningioma, gone under the radar for years, is finally rediscovered and pinpointed as the cause of the patient's near-fatal

[1] a brain tumor

condition. This case stresses the necessity of care continuity and easy access to patient history: had the meningioma been known to clinicians after the initial diagnosis, the patient may have been spared years of misery, a possible fatal outcome and enjoyed a better quality of life.
To solve such issues, authorized caregivers need fast and reliable access to a shared medical data repository containing tests' data and their interpretations. An international data repository would ideally be needed but is unlikely to be created in the foreseeable future for legal reasons. So national scale repositories should at least be created. The MGI report cited earlier ([1]) argues that sharing medical data offers huge premiums such as a drastic reduction of healthcare costs and waste and improved patient outcomes and quality of life through allowing remote patient monitoring, easing comparative effectiveness studies and clinical decision systems deployment and increasing data transparency.
Sharing data would also provide a trove of data on which competing automated medical data interpretation methods can easily be tested, compared, interpreted and reproduced. So far, the automated medical data interpretation methods aiming at reducing the clinicians' workload and easing the diagnosis process have been of limited use as they are tested on distinct, usually small data, making them hard to reproduce and interpret with any certainty.
The MGI report ([1]) also points out there are critical technical hurdles to overcome before medical data can be shared, analyzed properly and its full potential uncovered,e.g standardizing formats, ensuring systems' interoperability, integrating pre-existing, fragmented and heterogenous datasets and providing sufficient storage. So any potential design for a medical repository should take into account the distributed nature of the data [2], its heterogeneity and size and the diversity of file formats and platforms used across healthcare institutions. The data should also be easy to access for further, complex processing.
In this paper, we show that a rather low cost technical solution (and possible storage platform for medical data) that fits those constraints and requires minimal changes to current state of the art storage and processing techniques already exists: the Hadoop platform. In what follows, we will take EEG data as example of medical data.
The rest of this paper is organized as follows. We introduce Hadoop, explain why it is a good fit for medical data storage and show how EEGs can be stored with Hadoop (Section 2). The example of EEG feature selection by exhaustive search is then used to lay out why complex data processing should also be done with Hadoop (Sections 3 and 4).
Contributions In this paper, we give a proof of concept for an EEG repository by :

- explaining why Hadoop fits the constraints imposed on potential medical data repositories
- showing how to store EEG data in a Hadoop framework
- proving that EEG data can be analyzed on national scale on Hadoop by designing and benchmarking a representative machine-learning algorithm

[2] healthcare institutions are unlikely to let their data be stored externally

Table 1: Medical data statistics (2009 data, last year for which records are available) from [5]

	Netherlands	**USA**	**OECD**[4]
EEG[2]	100,000/167GB	N/A	N/A
MRI[3]	726,000/15.9TB	28 million/614TB	42 million/921TB
CT[3]	1.1 million/36.7TB	70 million/2.3PB	104.5 million/3.4PB

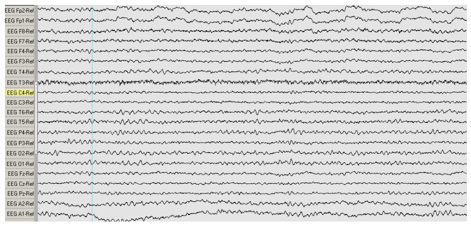


Fig. 1: EEG showing an adult's normal eyes-closed EEG segment

1.1 Related work

Hadoop has been found a viable solution for storing and processing big data similar to medical data, such as images in astronomy ([6]) or power grid time series, which unlike medical time series, are unidimensional time series ([7]).
[8] is, to the best of our knowledge, the first paper to consider storing medical data and EEGs in particular with Hadoop and show it is a promising solution in need of more testing. [8] suggest exploring the "design and benchmarking of machine learning algorithms on [the Hadoop] infrastructure and pattern matching from large scale EEG data." and this is one of the goals of our paper.

2 Hadoop: a good fit for medical repositories' constraints

2.1 Introduction to Hadoop

Hadoop, an open source platform managed by the Apache open source community, has 2 core components: the Hadoop Distributed File System (HDFS) and the job management framework or MapReduce framework. The HDFS is designed to reliably store huge files on all cluster machines. Each HDFS file is

[2] Assuming standard 20-minute EEGs in EDF+ format. File average size: 13.7MB
[3] Assuming average size of 23MB per MRI and 35MB per CT
[4] Based on data from OECD countries with available data from exams performed in and outside of hospitals i.e the USA, Greece, France, Belgium, Turkey, Iceland, Luxembourg, the Netherlands, Canada, Denmark, Estonia,the Czech Republic, the Slovak Republic, Chile, Israel and South Korea

cut into blocks and each block then replicated and stored at different physical locations in the cluster to ensure fault tolerance. The HDFS has a master/slave architecture with one master server called *Namenode* managing the filesystem namespace and regulating the file access by clients and multiple slave servers (one per cluster node) called *Datanodes* managing the storage in the nodes they run on. The *Namenode* maps the file blocks to the *Datanodes* and gives the *Datanodes* instructions to perform operations on blocks and serve filesystem clients' read and write requests. The Hadoop MapReduce framework also has a master/slave architecture with a single master called *jobtracker* and several slave servers (one per cluster node) called *tasktrackers*. MapReduce jobs are submitted to the *jobtracker*, which puts the jobs in a queue and executes them on first come/first serve basis. The *jobtracker* assigns tasks to the *tasktrackers*with instructions on how to execute them.

2.2 Hadoop and parallel data processing: the MapReduce model

MapReduce is a programming model for data-intensive parallelizable processing tasks (introduced in [9]) designed to process large volumes of data in parallel, with the workload split between large numbers of low level commodity machines. The MapReduce framework, unlike parallel databases, hides the complex and messy details of load balancing, data distribution, parallelization and fault-tolerance from the user in a library, thus making it simpler to use the resources of a large distributed system to process big datasets. The MapReduce model relies on 2 successive functions to transform lists of input data elements into lists of output data elements: a *mapper* function and a *reducer* function. Each input data element is transformed into a new output data element by the *mapper*. The transformed elements are then aggregated by the *reducer* to return a single output value. A simple example is files word count: in this case, the *mapper* associates a number of words to each of the input files while the *reducer* function sums the values obtained during the mapping step.

2.3 Hadoop for medical data storage

The Hadoop platform provides a solution to the technical hurdles outlined by the MGI report ([1]) described earlier (Section 1).
First of all, Hadoop was designed to scale with large data. It is currently being used at Facebook to store about 100PB of user data, i.e data much bigger than national scale medical data which ranges from dozens of terabytes (eg the Netherlands) to petabytes of data (eg the USA) annually as shown in Table 1. So Hadoop can easily handle national scale amount of medical data.
Moreover, Hadoop can store heterogeneous formats of data, in particular unstructured data, and if there is a method to extract the data from the files that store it [5], the data can then be fed to Hadoop MapReduce for further analysis

[5] Such methods currently exist at the sites where the different types of data are stored. There is,at most, a need to translate those methods into Java, Python, Perl or any other language that can be interfaced with Hadoop.

and processing.
Hadoop is also tolerant to node failure. The HDFS relies on replication (by default 3 copies on 3 Datanodes per file block) to ensure file blocks are not lost if a data server fails. If a Datanode fails and some data blocks have less than a set minimum of copies, the Namenode orders the replication of the affected blocks in some available Datanodes to bring back the replication factor of the blocks to safer levels. The probability of losing a block in a 4000 nodes' cluster in a day (respectively in a year) in the case of uncorrelated failures of multiple nodes is about 5.7×10^{-7} (respectively 2.1×10^{-4}) ([10]). At Yahoo! in 2009 for example, only 641 blocks were lost out of 329 million on 17720 nodes i.e a loss rate of $1.9 \times 10^{-4}\%$ ([10]). The only problem left is the Namenode as the HDFS is unusable if the Namenode fails. Namenode crashes rarely occur though ([11])(1 in 4 years at Facebook) and solutions limiting the crash impact are already being deployed. One such solution is the AvatarNodes in use at Facebook: 2 AvatarNodes, an active and standby one, replace the unique Namenode and receive the Datanodes messages in its stead. The Standby AvatarNode thus contains up-to-date information about block locations and can be started in under a minute to replace the Namenode (or Active AvatarNode) if it fails. This solution cuts cluster planned downtime by 50%. Data stored with Hadoop will therefore be constantly available.
Hadoop was built for parallel processing (via MapReduce described in Section 2.2) and we study the feasibility EEG data processing with Hadoop with the example of feature selection by exhaustive search in Section 3.

2.4 Hadoop and EEG storage

An EEG is a multidimensional time series obtained by capturing the brain's electric activity with scalp electrodes. Figure 1 shows an example of EEG. The increasingly popular EDF+ format is used to store EEGs and contains all the information about the EEG recording, both metadata in a header encoded in UTF-8 and raw data in binary format. The metadata includes patient information and EEG signal technical attributes (eg equipment details and sampling rate). Annotations on the EEG, such as context of recording or EEG events labels, may also be stored in the EDF+ file. See [12] for format details.
HDFS does not call for any set file format, so we store EEGs in EDF+ in HDFS. We anonymize EEGs before storage for security reasons. Keeping EEGs as EDF+ files has many advantages. No additional data formatting is needed and existing tools for EDF+ files, eg. visualization tools, can still be used. And as EDF+ files are mainly binary files, the size of the stored EEGs is small: 2500 EDF+ files (dataset 1 in Section 4 and Table 2(a)) i.e to about 2 years of EEG data at the local hospital take up 46.5GB whereas the same data [6] would take up 1TB when in a relational database.

[6] with one table for metadata, one table for raw data and one tuple per raw data point

3 EEG feature selection with Hadoop

EEG interpretation is arduous even for trained specialists due to the mass of data to interpret [7] and non-specific, age or context-dependent patterns and artifacts. For example, the patterns for a chewing artifact and an epileptic seizure are similar. Machine learning-based methods ([14, 15]) are being developed to ease the interpretation for clinicians, though the methods' scalability remains an issue. Instead of reducing algorithm complexity as in most studies aiming to lower the computational cost of machine-learning methods, we opt for using more commodity hardware with Hadoop and show, here, with EEGs as example, that parallelizable machine learning tasks and translatable to a sequence of *map/reduce* can be run in manageable times.

3.1 Feature selection as example EEG machine learning algorithm

Most automated EEG data interpretation methods classify or cluster EEGs and select suitable features for classification/clustering (eg. fractal dimension in [14, 16]) prior to it. Other approaches ([17]) select, quantify, visualize some "relevant" EEG features through time and present them to a practitioner who then interprets them and their variations to derive conclusions on the EEG. So the key task in the automated interpretation of EEG is feature selection so we pick a feature selection algorithm on EEG as example of machine-learning algorithm to determine whether Hadoop is suitable for medical data processing compared to other more traditional frameworks. We purposely choose an algorithm with exponential complexity for feature selection (exhaustive search) as achieving manageable execution times with Hadoop for this worst-case algorithm would entail achieving even more reasonable execution times for more common less computationally expensive algorithms. The goal of this study is not to evaluate the accuracy of the feature selection algorithm but to test whether running feature selection (as a sample machine-learning algorithm) on Hadoop has any benefits compared to using more traditional processing platforms.

3.2 Tested features and rationale for the choice of features

To test the feature selection algorithm, we choose a mix of 9 clinically-relevant and more general time-series features shown to be relevant for EEG processing in literature: 4 features computed in the time domain (fractal dimension, mean amplitude, amplitude standard deviation, normalized Hjorth mobility and complexity[8]) and 5 in the frequency domain (frequency bands percentages (α band,β band,θ band,δ band)[9], the α to δ ratio, high to low frequency ratio

[7] a routine 20 minute EEG fits in 109 A4-pages with the guidelines of the American Clinical Neurophysiology Society [13]

[8] 2-dimensional feature

[9] The EEG waves are grouped by frequency in 4 main bands: δ band for frequencies from 0.5 to 4 Hz, θ band for frequencies from 4 to 7 Hz, α band for frequencies from 7 to 12 Hz and β band for frequencies from 12 to 30 Hz. The frequency band percentage is therefore a 4-dimensional feature.

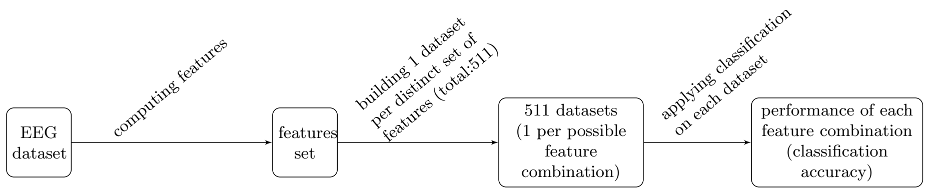

EEG dataset
computing features
features set
building 1 dataset per distinct set of features (total:511)
511 datasets (1 per possible feature combination)
applying classification on each dataset
performance of each feature combination (classification accuracy)

Fig. 2: EEG feature selection steps

(high frequency being frequencies above 25Hz), brain symmetry index (BSI) and spectral entropy). These features detect many pathologies and patterns: EEG asymmetries as in focal seizures or hemispheric ischemia with the BSI defined in [18], temporal lobe seizure with the Hjorth mobility and complexity([15]), high frequency artifacts with the high to low frequency ratio ([19]), hypofunctional patterns with the α to δ ratio and iso-electric ([19]), low-voltage EEGs with the mean amplitude ([19]). The fractal dimension separates normal sequences and other sequence types ([16]) and normal EEGs and Alzheimer patients EEGs ([14]). An extra feature, the nearest neighbour synchronization (mNNC) (defined in [17]), used to detect seizures ([19]), sleep or encephalopathies ([17]) [10] is computed in the feature computation step (to measure scalability) but not used for classification.Each of the 9 features can be picked alone or in combination with a variable number of the other features. So there are $\sum_{i=1}^{9} C_9^i = 511$ distinct possible ways to pick a feature set from the 9 features. This paper doesn't aim to assess the classification performance of the chosen features. The features were only picked as sample EEG features for scalability tests so others may have been selected for this study.

3.3 Performing EEG feature selection with exhaustive search

We evaluate each of the 511 possible feature combinations to select the best feature combination for our classification problem. Figure 2 summarizes the feature evaluation steps. For simplicity, we choose KNN as classifier but the same principle applies to other classifiers. We then implement this algorithm in 4 steps in MapReduce:

1. Map: Extract the segments of interest from the original EEG files and compute all features for each of the segments
2. Reduce: Build one dataset per feature combination
3. Map: Train the classifier and assess its performance for each feature set
4. Reduce: Choose the feature set that maximizes mean accuracy (for all classes).

Details on the classifier and EEG segments of interest are found in Section 4.1.

[10] the mNNC value increases in seizures and decreases in sleep or encephalopathies

4 Experiments

This section describes the experiments performed and their setup. Table 2 summarises the hardware and software properties of the experimental servers.

4.1 Details on EEG classification

EEG labeling hinges on properties such as sequence type and patient age so feature selection can be done only on segments of similar properties. Only eyes-closed segments from adult EEGs are used in this paper. The feature selection principle is unchanged for other age groups and segment types. We use KNN as a classifier. We assess a feature set's performance by the mean classification accuracy (mean of the accuracy for all classes) and run 3 rounds of the Shuffle and Split cross-validation, with 30% of the data used as training set per iteration, to reduce overfitting and minimize the prediction error. We have 3 EEG classes: normal, normal but for increased β wave (often due to medication) and abnormal.

4.2 Dataset description

We use a dataset of 2500 EEGs for the experiments. This amount of data is about 30% of the EEG data collected monthly[11] in the Netherlands and about 2 years of data from the local hospital[12]. All EEGs in the dataset were recorded on patients in a hospital setting following the International 10/20 System with Ag/AgCl electrodes and using a common average reference. Only the 19 channels common to all EEGs are kept for calculations, with each channel sampled at 250Hz. All 9 features from Section 3.2 and mNNC are computed on the whole dataset (hereafter named dataset 1-Table 2(a)) to check the scalability of feature computation. To test feature selection by exhaustive search, we use a subset of 1000 files from dataset 1 for which the class label is known precisely (hereafter named dataset 2). The EEGs in both datasets predominantly represent standard EEGs (15 to 40 minutes' EEGs) i.e the most common EEGs in clinical practice (91.6% of the EEGs recorded per year at the local hospital).

4.3 Benchmarking the EEG exhaustive search feature selection

Setup We test EEG feature selection with python and with Hadoop Streaming. To speed up the python code, we use the joblib library to parallelize parts of the feature selection: features are computed EEG by EEG with several tasks running concurrently and several feature combinations are tested for classification at the same time. The number of jobs running concurrently is RAM-bound.
We selected Hadoop Streaming as Hadoop interface as we can write python code with it. This allows us to reuse most of the code from the python with joblib approach, thus easing the performance comparison between both approaches

[11] and about a third of the annual Dutch data in filesize
[12] Medisch Spectrum Twente, Enschede, The Netherlands

Table 2: Server and EEG test file characteristics

(a) Characteristics of experimental datasets

Dataset	**Number of files**	**Total size of files**	**Minimum EEG duration**	**Maximum EEG duration**	**Number of files of duration**					**Number of values**
					<15mn	**15 to 40 mn**	**40mn to 1h**	**1 to 2h**	**>2h**	
dataset 1 (feature computation only)	2500	46.51GB	10s	3h 9mn	204 (7.4% of files)	2201 (79.5% of files)	90 (3.25% of files)	253 (9.14% of files)	19 (0.69% of files)	578,648,474,500
dataset 2 for classification subset of dataset1)	1000	16.06GB	10s	2h 8mn 50s	73 (5.6% of files)	909 (69.9% of files)	33 (2.54% of files)	35 (2.69% of files)	1 (0.08% of files)	6,828,505,000

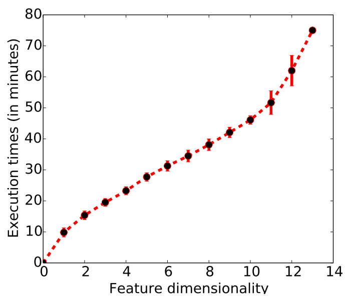
(b) Characteristics of the servers used in the experiments

Server	**OS**	**Software used**	**Processor**	**RAM**	**Number of nodes**
Server for Parallel Python experiments	openSUSE 12.3 Milestone 2(x86-64) Kernel version 3.6.3-1-desktop	Python 2.7.3 with joblib 0.7d library scikit-learn 0.14 scikit-learn 0.14	AMD Opteron® Processor 4226 (6 cores) 2 processors	64GB	1
Hadoop cluster	Ubuntu 12.04.2 LTS(x86-64) Kernel version 3.2.0-40-generic	Python 2.7.3 with scikit-learn 0.10 Hadoop streaming jar from Cloudera Hadoop CDH3u6	Intel®Xeon® CPU E3110@3.00GHz (2 cores) 1 processor	7.8GB	15

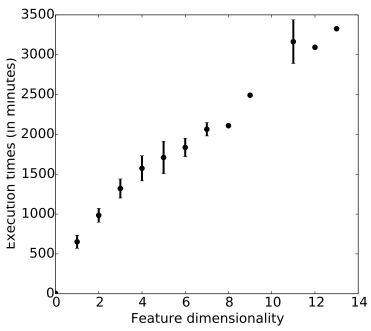
tested. There are 30 available map slots in the Hadoop cluster (2 maps per node) so that up to 30 maps run at the same time until the Hadoop map jobs are done. Similarly there are 30 possible reduce slots. Unless otherwise stated, we run 2 maps per node for the Hadoop Streaming jobs. We compute all features over windows of 1800 ms in both Hadoop and Python approaches. 1800 ms of EEG data equals 450 points per channel with the standard frequency of EEG signal i.e 250Hz and about 9 eye blink artifacts (shortest known EEG events).

Experiment 1: Feature computation In the first set of experiments, we only perform the first step of feature selection (described in Section 3.3), i.e EEG segment extraction and feature computation, on part or all of dataset 1. For each experiment, execution times are recorded. Figures 4(a), 4(b), and 4(d) were obtained using all of dataset 1. Figure 4(d) explores the evolution of feature computation times when the number of cores of the Python server is made to vary. Feature computation execution times grow linearly with the size of processed files for both Hadoop and Python solutions (Figure 4(e)) but the Python execution times grow 4.5 times faster than the Hadoop ones. Therefore, feature extraction with Hadoop is especially beneficial for large files and scales to a national scale amount of data. Based on the interpolations of Figure 4(e), extracting the 10 features from Section 3.2 for the whole annual Dutch EEG data(i.e 167GB-Table 1) would take about 11 hours and 7 minutes with Hadoop compared to more than 2 days with Python. The Python execution time decreases exponentially with the number of active cores/CPUs (Figure 4(d)) but an infinite number of CPUs would be needed to reach the same performance as Hadoop!

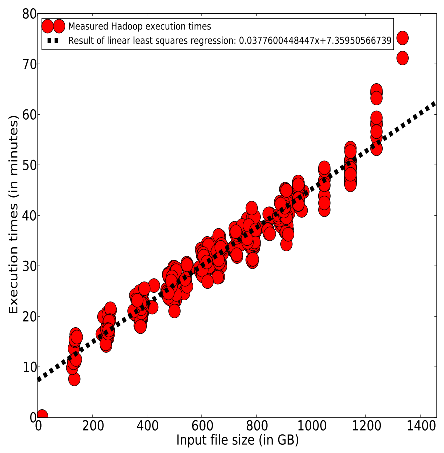
Fig. 3: Impact of several factors on features' computation and classification execution times



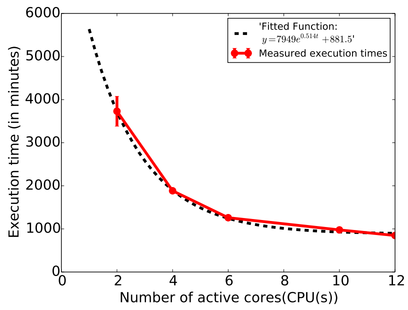
(a) Hadoop solution



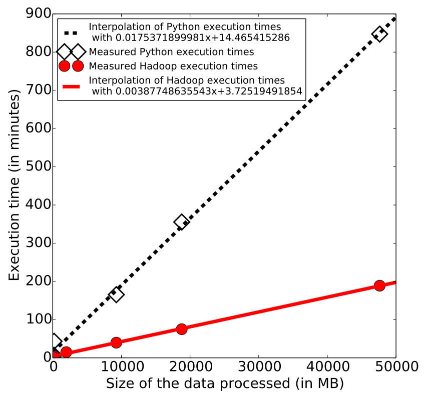
(b) parallel Python solution



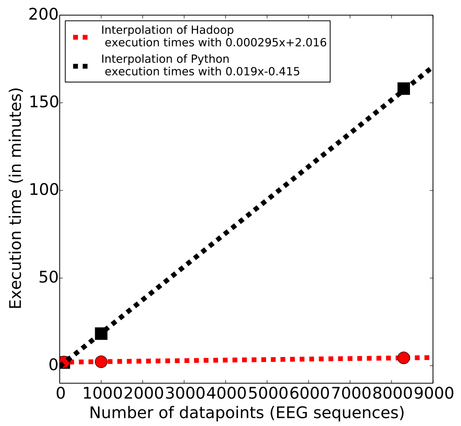
(c) Evolution of Hadoop classification execution times with classification input file size (1 file per combination)



(d) Evolution of features' computation execution times (parallel Python) with number of active cores



(e) Variation of feature computation execution times with EEG file size



(f) Variation of classification execution times with number of data points (test and training sets)

Experiment 2: Brute-force classification and feature selection Experiments described here all use dataset 2 (see Section 4.2 and Table 2(a) for details) and test the time it takes to assess the classification performance of all possible 511 feature combinations [13]. 253295 EEG segments are extracted from dataset 2, i.e 113,982,750 values or 1.67% of the total values in the original files. Table 3 summarises the results of implementing the feature selection algorithm described in Section 3.3 with Hadoop Streaming and Python. Due to recurrent memory errors, only 154 feature combinations out of 511 (30.14%) were tested for classification with Python. The execution times for Python classification in Table 3 are estimates based on available data. Insufficient RAM per Hadoop node led to all 511 combinations being tested with 37 successive jobs [14] instead of one so that only 1 map would run per node and not 2. The current implementation is clearly subpar as map slots become available as the job runs but are unusable until the job ends and the next starts. This is however easily fixed, with the right user privileges, by setting the maximum number of maps per node to 15 so that at any time only one map runs per node: all 511 classifications can then run in a single Hadoop job. Table 3 shows that even this suboptimal solution evaluates the classification performance of all feature sets faster than Python. The gap in classification execution times between Hadoop and Python widens with the size of datasets to classify (Figure 4(f)). For very small datasets (33 training and 67 test points), Python outperforms Hadoop slightly (1.82 minutes for Python and 2.4 minutes with Hadoop to test all 511 combinations). Hadoop has overall a clear edge over Python as dataset size rises: the classification runs about 64.76 times faster on Hadoop. Classifying dataset 2's sequences, even in suboptimal conditions with Hadoop, runs 29.9 to 34.16[15] times faster than with Python (see Table 3). So Hadoop is more suited for large datasets' classification. Hadoop also scales linearly with the size of classification input files[16] (Figure 4(c)) and handles feature dimensions' increase better than Python (about 2 orders of magnitude faster than Python (Figures 4(a) and 4(b))).

4.4 Discussion

The experiments (Section 4.3) show Hadoop as a scalable and promising solution to process EEGs if the task at hand it parallelizable (eg feature computation) even if it is CPU-intensive and RAM-bound (classification with all possible feature combinations). It goes to prove that a cluster of commodity hardware (15 machines with Dual core processors and only 7.8GB of RAM here) is better at processing complex data than a single highly specialized powerful server if the task is a series of (semi-)independent steps that can run in parallel. Hadoop has also been shown to be able to process a national scale amount of data with a

[13] all features except nearest neighbor synchronization
[14] 36 testing 14 combinations at a time and 1 testing 7 combinations at a time
[15] compared to the estimated upper and lower bounds for the Python job respectively
[16] files obtained by extracting all eyes closed segments from the original EDF+ files and applying each of the 9 tested features on the extracted segments
[15] Result of 37 successive jobs instead of only one job testing all 511 combinations
[16] estimates based on data available

Table 3: Execution times for whole feature selection process on dataset 2 and each of its steps

		Segment extraction& feature computation only	**Feature computation and formatting for classification**	**Classification only**	**Complete feature selection selection**
Execution time	Hadoop streaming	30.35min	1h7min20s	32h25min52s[9]	33h33min12s
	parallel Python	97.9min	97.9min	estimated lower bound: 11 days 47min[10] estimated upper bound: 12 days 14h34min	estimated lower bound: 11days2h25min estimated upper bound: 12 days 16h2 min

quite small number of cluster machines. This is also a rather cheap solution: a cluster like the experimental one costs 10000 to 20000 euros i.e 1000-1500 euros per machine as compared to above 3000 euros per machine for the type of server used in the Python experiments. Owning a Hadoop cluster is in theory not needed as web services like Amazon Elastic Map Reduce (EMR) offer access to Hadoop clusters tailored for diverse processing needs. This is not doable, though, given the sensitivity of medical data. And we can boost the Hadoop performance further by optimizing the code we wrote by mostly reusing the Python one, via for example, changing the Hadoop configuration parameters to solve memory issues or using other Hadoop Python frameworks like mrjob or Dumbo that don't require map/reduce inputs and outputs to be strings passed via stdin/stdout and should thus need less processing RAM or using machine-learning algorithms optimized for the platform (Mahout library).

5 Conclusions

Hadoop is a promising solution for EEG storage and processing. Computation times for complex parallelizable machine-learning algorithms are notably reduced compared to more traditional means of computation and become manageable. The gain in computation times grows with data amount to process, Hadoop scaling easily with national scale data. So it would seem that it is better to process data with many commodity machines rather than with one extremely powerful server, when the processing task is parallelizable. In future, we would like to extend this work to other medical data types such as MRI or CT and study how to integrate data from computations run on diverse types of medical data (eg MRI and EEG). We would also like to run more tests on medical data querying (especially natural language querying). And Hadoop data security also needs to be explored further.

Acknowledgments The EEGs used in this paper were kindly provided by Prof. Dr. Ir. Michel van Putten (Dept. of Neurology and Clinical Neurophysiology, Medisch Spectrum Twente and MIRA, University of Twente, Enschede, The Netherlands), who we also thank for useful comments on the paper.

References

1. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A.H.: Big data: The next frontier for innovation, competition, and productivity. Technical report, Mc Kinsey Global Institute (June 2011)
2. Eta S. Berner, M.L.G.: Overconfidence as a cause of diagnostic error in medicine. Am. J. Med **121**(5) (May 2008) S2–S23 (Supplement)
3. Schiff, G.D., Hasan, O., Kim, S., Abrams, R., Cosby, K., Lambert, B.L., Elstein, A.S., Hasler, S., Kabongo, M.L., Krosnjar, N., Odwazny, R., Wisniewski, M.F., McNutt, R.A.: Diagnostic error in medicine: Analysis of 583 physician-reported errors. Arch Intern Med **169**(20) (2009) 1881–1887
4. Boodman, S.G.: Medical mystery: Depression and possible dementia masked the real problem. The Washington Post (Dec. 23 2013)
5. OECD: Medical technologies. In: Health at a Glance 2011: OECD Indicators . OECD Publishing (2011)
6. Wiley, K., Connolly, A., Gardner, J.P., Krughof, S., Balazinska, M., Howe, B., Kwon, Y., Bu, Y.: Astronomy in the cloud: Using mapreduce for image coaddition. CoRR **abs/1010.1015** (2010)
7. Bach, F., Çakmak, H.K., Maass, H., Kuehnapfel, U.: Power Grid Time Series Data Analysis with Pig on a Hadoop Cluster compared to Multi Core Systems. In: Proc. of PDP 2013. (2013)
8. Dutta, H., Kamil, A., Pooleery, M., Sethumadhavan, S., Demme, J.: Distributed storage of large-scale multidimensional electroencephalogram data using hadoop and hbase. In: Grid and Cloud Database Management. Springer (2011) 331–347
9. Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. In: OSDI, USENIX Association (2004) 137–150
10. Chansler, R.J.: Data availability and durability with the Hadoop Distributed File System. **37**(1) (February 2012) 16–22
11. Borthakur, D., Gray, J., Sarma, J.S., Muthukkaruppan, K., Spiegelberg, N., Kuang, H., Ranganathan, K., Molkov, D., Menon, A., Rash, S., Schmidt, R., Aiyer, A.: Apache hadoop goes realtime at facebook. In: Proc. of SIGMOD. (2011) 1071–1080
12. Kemp, B., Olivan, J.: European data format 'plus' (EDF+), an EDF alike standard format for the exchange of physiological data. Clin Neurophysiol **114** (2003) 1755–61
13. American Clinical Neurophysiology Society: Guideline 8: Guidelines for recording clinical EEG on digital media. J Clin Neurophysiol **23** (Apr. 2006) 122–124
14. Goh, C., Hamadicharef, B., Henderson, G.T., Ifeachor, E.C.: Comparison of Fractal Dimension Algorithms for the Computation of EEG Biomarkers for Dementia. In: Proc. of CIMED. (2005)
15. Cecchin, T., Ranta, R., Koessler, L., Caspary, O., Vespignani, H., Maillard, L.: Seizure lateralization in scalp EEG using Hjorth parameters. Clin Neurophysiol **121**(3) (Mar. 2010) 290–300
16. Berrada, G., de Keijzer, A.: An IFS-based similarity measure to index electroencephalograms. In: Proc. of PAKDD. (2011) 457–468
17. van Putten, M.J.: The Colorful Brain: Visualization of EEG Background Patterns. J Clin Neurophysiol **25**(2) (2008) 63–68
18. van Putten, M.J.: Extended BSI for continuous EEG monitoring in carotid endarterectomy. Clin Neurophysiol **117**(12) (2006) 2661–2666
19. Cloostermans, M., de Vos, C., Heida, T., de Keijzer, A., van Putten, M.: Monitoring the brain in the adult ICU. In: Proc. of the 4th Annual Symposium of the IEEE/EMBS Benelux Chapter. (Nov. 2009) 128–130