diseases

MDPI

Article

A Magnetoencephalographic/Encephalographic (MEG/EEG) Brain-Computer Interface Driver for Interactive iOS Mobile Videogame Applications Utilizing the Hadoop Ecosystem, MongoDB, and Cassandra NoSQL Databases

Wilbert McClay [1,2,3,4,5,6]

[1] Information Systems Department, Northeastern University, Boston, MA 02115, USA; mcclay.w@husky.neu.edu; Tel.: +1-571-272-8012
[2] Department School of Social Work, Tulane University School of Medicine, New Orleans, LA 70112, USA
[3] Department of Information Assurance, Northeastern University, Boston, MA 02115, USA
[4] Lawrence Livermore National Laboratory, Livermore, CA 94550, USA
[5] Department of Information Assurance, Brandeis University, Waltham, MA 02453, USA
[6] Department of Mathematics, Brandeis University, Waltham, MA 02453, USA

Received: 3 June 2018; Accepted: 21 August 2018; Published: 28 September 2018

Abstract: In Phase I, we collected data on five subjects yielding over 90% positive performance in Magnetoencephalographic (MEG) mid-and post-movement activity. In addition, a driver was developed that substituted the actions of the Brain Computer Interface (BCI) as mouse button presses for real-time use in visual simulations. The process was interfaced to a flight visualization demonstration utilizing left or right brainwave thought movement, the user experiences, the aircraft turning in the chosen direction, or on iOS Mobile Warfighter Videogame application. The BCI's data analytics of a subject's MEG brain waves and flight visualization performance videogame analytics were stored and analyzed using the Hadoop Ecosystem as a quick retrieval data warehouse. In Phase II portion of the project involves the Emotiv Encephalographic (EEG) Wireless Brain–Computer interfaces (BCIs) allow for people to establish a novel communication channel between the human brain and a machine, in this case, an iOS Mobile Application(s). The EEG BCI utilizes advanced and novel machine learning algorithms, as well as the Spark Directed Acyclic Graph (DAG), Cassandra NoSQL database environment, and also the competitor NoSQL MongoDB database for housing BCI analytics of subject's response and users' intent illustrated for both MEG/EEG brainwave signal acquisition. The wireless EEG signals that were acquired from the OpenVibe and the Emotiv EPOC headset can be connected via Bluetooth to an iPhone utilizing a thin Client architecture. The use of NoSQL databases were chosen because of its schema-less architecture and Map Reduce computational paradigm algorithm for housing a user's brain signals from each referencing sensor. Thus, in the near future, if multiple users are playing on an online network connection and an MEG/EEG sensor fails, or if the connection is lost from the smartphone and the webserver due to low battery power or failed data transmission, it will not nullify the NoSQL document-oriented (MongoDB) or column-oriented Cassandra databases. Additionally, NoSQL databases have fast querying and indexing methodologies, which are perfect for online game analytics and technology. In Phase II, we collected data on five MEG subjects, yielding over 90% positive performance on iOS Mobile Applications with Objective-C and C++, however on EEG signals utilized on three subjects with the Emotiv wireless headsets and ($n < 10$) subjects from the OpenVibe EEG database the Variational Bayesian Factor Analysis Algorithm (VBFA) yielded below 60% performance and we are currently pursuing extending the VBFA algorithm to work in the time-frequency domain referred to as VBFA-TF to enhance EEG performance in the near future. The novel usage of NoSQL databases, Cassandra and MongoDB, were the primary main enhancements of the BCI Phase II MEG/EEG brain signal data acquisition, queries, and rapid analytics, with MapReduce and Spark DAG demonstrating future implications for next generation biometric MEG/EEG NoSQL databases.

Keywords: brain-computer interface; machine learning algorithms; encephalographic (EEG); magnetoencephalographic (MEG); Hadoop Ecosystem; iOS Mobile Applications; MongoDB; Cassandra; Emotiv EPOC headset; OpenVibe

1. Introduction

In Phase I, "*A Real-Time Magnetoencephalography Brain-Computer Interface Using Interactive 3D-Visualization and the Hadoop Ecosystem*", Journal of Brain Sciences, 2015 [1], was developed with an overall above 92% performance on 5 MEG subjects utilizing the Variational Bayesian Factor Analysis (VBFA) Machine Learning algorithm.

Secondly in Phase II, demonstrate in Figures 1 and 2, below, we extend the MEG Subjects' brainwave [2] data storage to MongoDB for ease of use and relevance with the Internet of Things (IoTs) sensor based data acquisition [3] for analytical processing and data storage in Figure 2, extended to implementation with Apple iOS Mobile Applications.

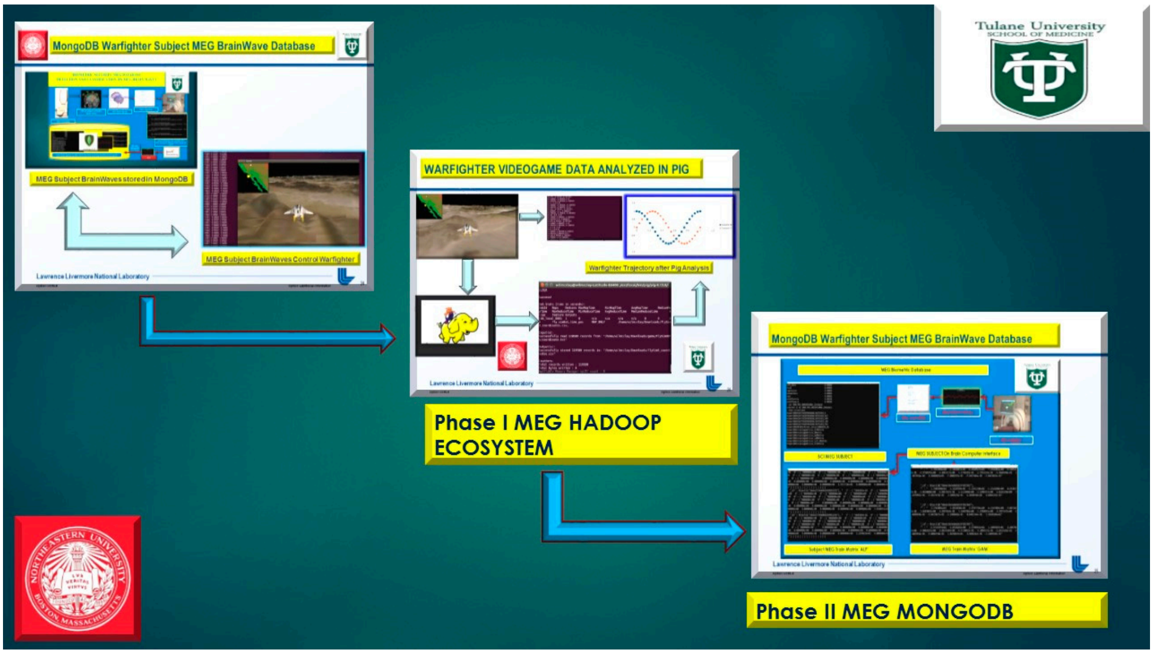

MongoDB Warfighter Subject MEG BrainWave Database
MEG Subject BrainWaves stored in MongoDB
MEG Subject BrainWaves Control Warfighter
Lawrence Livermore National Laboratory
WARFIGHTER VIDEOGAME DATA ANALYZED IN PIG
Warfighter Trajectory after Pig Analysis
Lawrence Livermore National Laboratory
Phase I MEG HADOOP ECOSYSTEM
MongoDB Warfighter Subject MEG BrainWave Database
Lawrence Livermore National Laboratory
Phase II MEG MONGODB
Tulane University School of Medicine

Figure 1. Phase II, MongoDB MEG Brain Computer Interface Database(s).

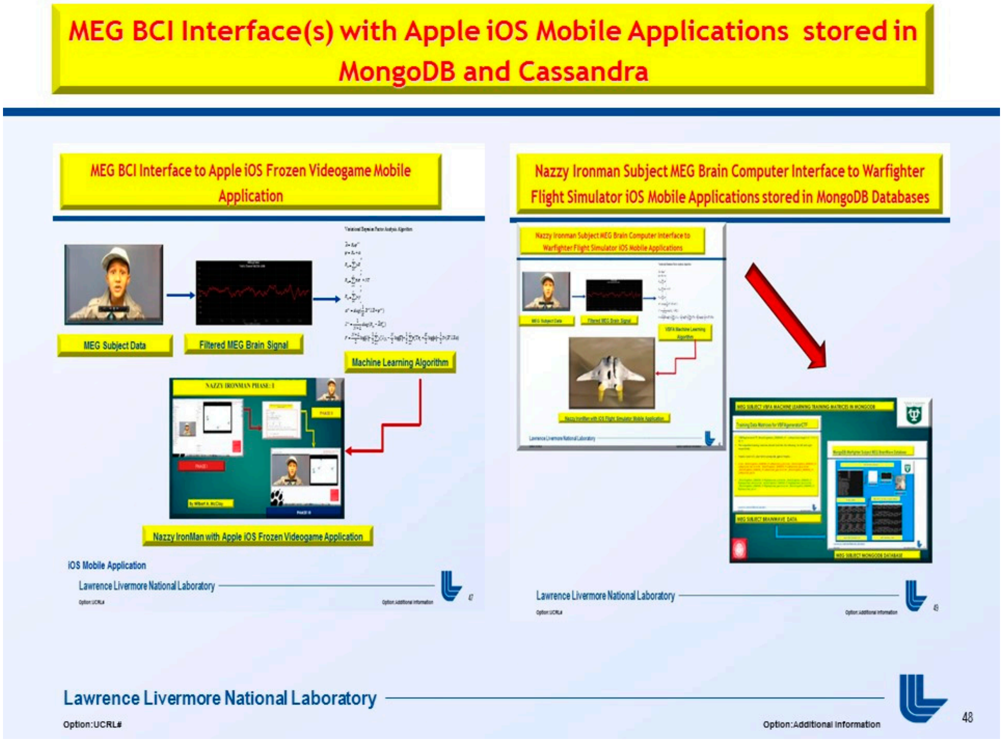

MEG BCI Interface(s) with Apple iOS Mobile Applications stored in MongoDB and Cassandra
MEG BCI Interface to Apple iOS Frozen Videogame Mobile Application
MEG Subject Data
Filtered MEG Brain Signal
Machine Learning Algorithm
NAZZY IRONMAN PHASE: I
Nazzy IronMan with Apple iOS Frozen Videogame Application
iOS Mobile Application
Lawrence Livermore National Laboratory
Option:UCRL#
Option:Additional Information
47
Nazzy Ironman Subject MEG Brain Computer Interface to Warfighter Flight Simulator iOS Mobile Applications stored in MongoDB Databases
Nazzy Ironman Subject MEG Brain Computer Interface to Warfighter Flight Simulator iOS Mobile Applications
VBFA Machine Learning Algorithm
MEG SUBJECT VBFA MACHINE LEARNING TRAINING MATRICES IN MONGODB
MEG SUBJECT BRAINWAVE DATA
MEG SUBJECT MONGODB DATABASE
Lawrence Livermore National Laboratory
Option:UCRL#
Option:Additional Information
49
Lawrence Livermore National Laboratory
Option:UCRL#
Option:Additional Information
48

Figure 2. Phase II, magnetoencephalography brain-computer interface(s) (MEG BCI) with Apple iOS Mobile Applications stored in MongoDB and Cassandra.

Thus, the acquisitioned tomographic magnetoencephalography/electroencephalography (MEG/EEG) subject brainwave signals can be uploaded locally on a single node or into a new and innovative cloud based hosting infrastructure referenced as a Cloud Service Provider (CSP) utilizing MongoDB and/or Cassandra as the NoSQL databases.

The previous tomographic MEG/EEG subject brainwave data analysis locally stored acquisitioned brainwave signals into directories in a UNIX based file system or utilizing a relational database system, such as PostgreSQL or MySQL. However, this proved unfruitful for real-time data acquisition for subject's brain signals and was focused on a RAID (Redundant Array of Disks) architecture. The main fallbacks and hazards with utilizing a Relational Database Management System for acquired subject's brain signals in real-time is if a MEG/EEG electrode array or sensor fails to acquisition signal data from the subject it will result in nulls in the Relational Database Management System (RDBMS). Furthermore, if the Database System Administrator attempts to run queries with a RDBMS filled with null values it will cause major castrophes and possible crash the entire system, particularly for interactive queries on large datasets.

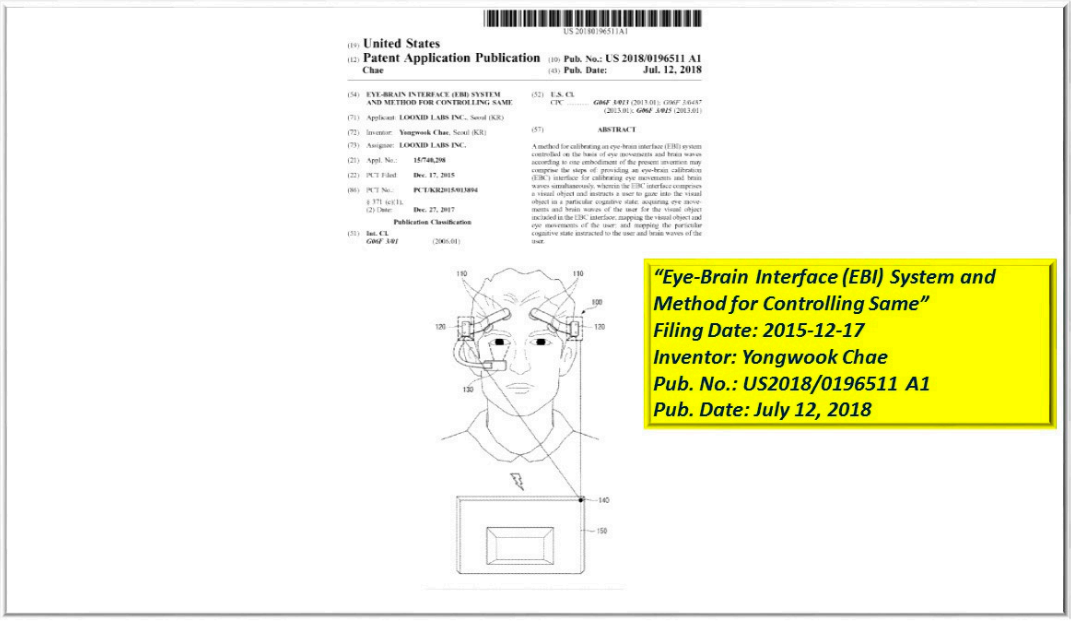
The innovative and next generation usage of NoSQL databases, referenced as MongoDB and Cassandra, where the rows and columns in the database do not need to be uniform as in a RDBMS, and is also schema-less architecture, where the rows and columns do not require uniformity.

For instance, this provides a distinct advantage for real-time brain signal acquisition, because, if a sensor fails, the MongoDB NoSQL database is not compromised, and additionally MongoDB uses "Replication and Sharding" for distributed computing where the data is separated into blocks or chunks and spread across distributed machines. Likewise, CASSANDRA NoSQL database is a column-oriented NoSQL database designed by Facebook and distinctly designed for fast reads and writes, fault-tolerance, and elasticity, and is more favorable than most NoSQL databases in that domain. In terms of certain aspects of performance, the CASSANDRA NoSQL database is column-oriented NoSQL database and has faster reads, writes, and elasticity than MongoDB, but MongoDB is more popular and versatile for the usage of Internet of Things (IoTs) devices (e.g. sensors, MEG/EEG brain signal electrode array sensors) [2,3].

The concept of extracting brainwaves and classification of user intent is referred to as brain-computer interfaces (BCIs), sometimes called mind-machine interfacing (MMI) or brain-machine interfacing (BMI), has been evolving for many years. These interfaces are used for both noninvasive procedures (such as magnetoencephalography (MEG) and electroencephalography (EEG)), as well as for invasive procedures (such as electrocorticographic (ECoG) events). What follows is a brief discussion of the history and importance of BCIs in the noninvasive procedures of MEG and EEG as they relate to recent applications, ranging from interactive video game technology to robotics and mobile applications.

One of the most dynamic current applications of these BCI developments is, "User Input Validation and Verification for Augmented and Mixed Reality Experiences", created by Aylin Climenser, Hani Awni, Frank Chester Irving, Jr., and Stefanie A. Hutka, under United States Patent Publication Number: US2018/0188807A1 [4]. The premise utilizes a head mounted display (HMD) incorporating an EEG to monitor and analyze event related potentials (ERP) to a given stimulus that was observed by the subject. Additionally demonstrated in Illustration C, Yongwook Chae with LOOXID LABS, INC., was issued patent US2018/0196511, "EYE-BRAIN INTERFACE (ERI) SYSTEM AND METHOD FOR CONTROLLING SAME", [5], on 12 July 2018, where the system is defined as providing an eye-brain calibration (EBC) interface for calibrating eye movements and brain waves simultaneously, in addition, the EBC interface is composed of a visual object and instructs a user to focus or gauze into the visual object in a particular cognitive state; thus acquisitioning eye movements and brainwaves of the user for the visual object that was included in the EBC.

Furthermore, Cruz-Hernandez at Immersion Corporation in 2018 was issued patent US 20170199569 A1, entitled, "Systems and methods for haptically-enabled neural interfaces", the processor is configured to receive a sensor signal from a neural interface configured to detect an electrical signal that is associated with a nervous system demonstrated in Figure 3, above. Additionally, the processor is utilized to detect an interaction with a virtual object in a virtual environment based on the sensor signal [4].



US 20180196511A1

(19) **United States**
(12) **Patent Application Publication**
Chae

(10) **Pub. No.: US 2018/0196511 A1**
(43) **Pub. Date: Jul. 12, 2018**

(54) EYE-BRAIN INTERFACE (EBI) SYSTEM AND METHOD FOR CONTROLLING SAME

(71) Applicant: LOOXID LABS INC., Seoul (KR)

(72) Inventor: **Yongwook Chae**, Seoul (KR)

(73) Assignee: LOOXID LABS INC.

(21) Appl. No.: 15/740,298

(22) PCT Filed: Dec. 17, 2015

(86) PCT No.: PCT/KR2015/013894
§ 371 (c)(1),
(2) Date: Dec. 27, 2017

Publication Classification

(51) Int. Cl.
G06F 3/01 (2006.01)

(52) U.S. Cl.
CPC *G06F 3/013* (2013.01); *G06F 3/0487* (2013.01); *G06F 3/015* (2013.01)

(57) **ABSTRACT**

A method for calibrating an eye-brain interface (EBI) system controlled on the basis of eye movements and brain waves according to one embodiment of the present invention may comprise the steps of: providing an eye-brain calibration (EBC) interface for calibrating eye movements and brain waves simultaneously, wherein the EBC interface comprises a visual object and instructs a user to gaze into the visual object in a particular cognitive state; acquiring eye movements and brain waves of the user for the visual object included in the EBC interface; mapping the visual object and eye movements of the user; and mapping the particular cognitive state instructed to the user and brain waves of the user.

"Eye-Brain Interface (EBI) System and Method for Controlling Same"
Filing Date: 2015-12-17
Inventor: Yongwook Chae
Pub. No.: US2018/0196511 A1
Pub. Date: July 12, 2018

Figure 3. Yongwook Chae, "EYE-BRAIN INTERFACE (ERI) SYSTEM AND METHOD FOR CONTROLLING SAME", US2018/0196511.

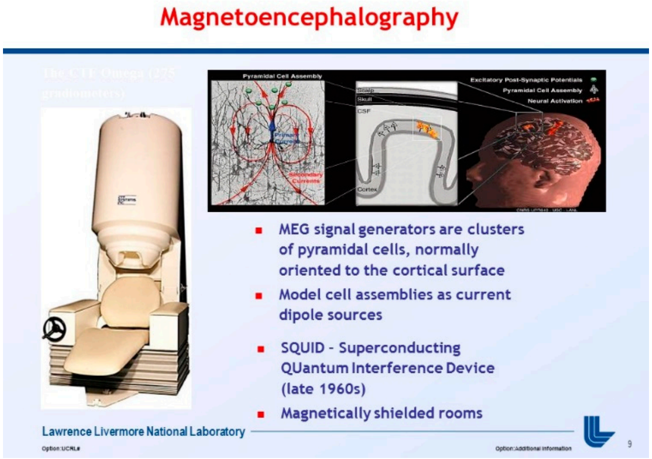
As a result of the innovation of the next generation MEG and EEG BCI applications, this technology is potentially applicable to other types of cloud based architectures utilizing Big Data analytics to reduce computational time and expense. The general consensus is that such methods could yield the same result with less time lag and more compact BMI devices [2].

At the moment, the integration of the gaming industry, mobile, and Big Data analytics was approximated at $137.9 billion for 2018 [6]. The use of NoSql databases, such as Cassandra, MongoDB, and the Hadoop Ecosystem yields keen competitive advantages over legacy relational transactional databases, and web-based games are now the go-to standard platform with the brisk adoption of mobile games [7]. Thus, an MEG based Brain-computer Interface (BCI) utilizing videogame analytics attracts two primary audiences: (1) the neuroscience and neuro-engineering scientific community and (2) gaming and Big Data analytics industry. The market revenue for BCI applications interfaced to videogames has unparalleled future market revenue for avid gamers and diligent research scientists. Furthermore, the videogame analytics and processing were stored in the Hadoop Ecosystem demonstrated in Figures 1 and 2 (above), Figure 4, and below in Figures 5 and 6.

Currently, the telemedicine & healthcare industry has now adopted gamification, or the utilization of game mechanics and design, to inspire people for motivation and behavioral influence that is focused on wellness and healthy behaviors [7].

2. UCSF MEG System

At the University of California, San Francisco (UCSF), MEG technology is being used to study multimodal and multiscale imaging of dynamic brain function as well as cortical spatiotemporal plasticity in humans [1,8]. Thus, several novel signal processing and machine learning algorithms had to be constructed, and UCSF had to fully utilize its twin 37-channel bio magnetometer. This machine uses 275 channel SQUIDS-based detectors, housed in a magnetically shielded room (MSR), to noninvasively detect tiny magnetic fields generated by neuronal activity in the brain, as demonstrated in Figure 4, below.



Magnetoencephalography

- MEG signal generators are clusters of pyramidal cells, normally oriented to the cortical surface
- Model cell assemblies as current dipole sources
- SQUID - Superconducting QUantum Interference Device (late 1960s)
- Magnetically shielded rooms

Lawrence Livermore National Laboratory

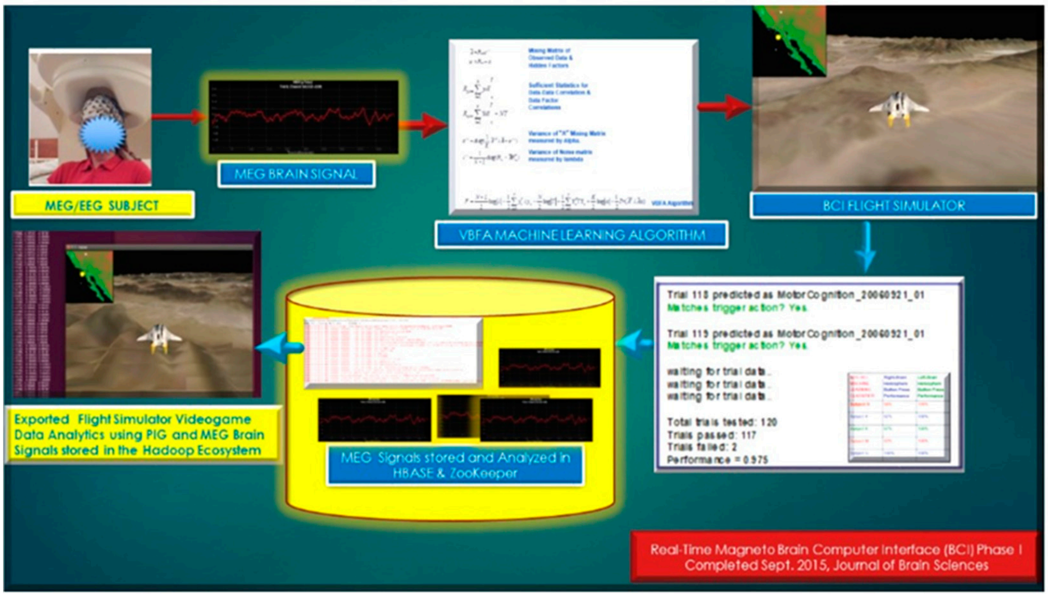
Option:UCRL#

Option:Additional information

9

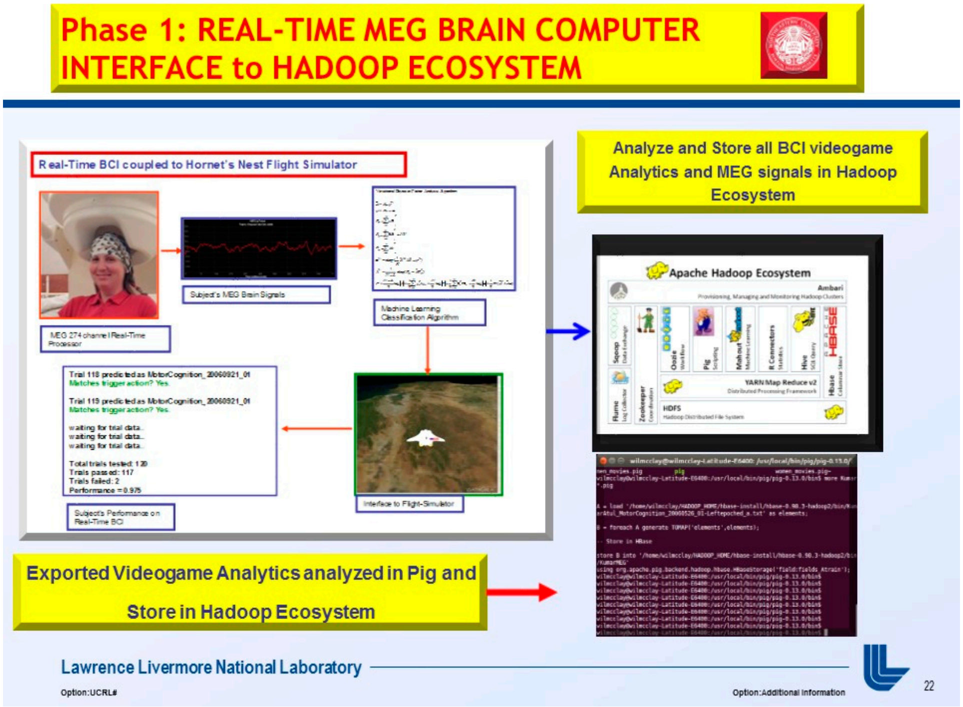
The CTF Omega (275 gradiometers)
Pyramidal Cell Assembly
Primary Current
Secondary Currents
Scalp
Skull
CSF
Cortex
Excitatory Post-Synaptic Potentials
Pyramidal Cell Assembly
Neural Activation

Figure 4. University of San Francisco in California (UCSF) MEG Scanner with Superconducting Quantum Interference Device (SQUID) detectors.


MEG/EEG SUBJECT
MEG BRAIN SIGNAL
VBFA MACHINE LEARNING ALGORITHM
BCI FLIGHT SIMULATOR
Exported Flight Simulator Videogame Data Analytics using PIG and MEG Brain Signals stored in the Hadoop Ecosystem
MEG Signals stored and Analyzed in HBASE & ZooKeeper
Real-Time Magneto Brain Computer Interface (BCI) Phase I Completed Sept. 2015, Journal of Brain Sciences

```
Trial 118 predicted as MotorCognition_20060921_01
Matches trigger action? Yes

Trial 119 predicted as MotorCognition_20060921_01
Matches trigger action? Yes

waiting for trial data.
waiting for trial data.
waiting for trial data.

Total trials tested: 120
Trials passed: 117
Trials failed: 2
Performance = 0.975
```

Figure 5. Phase I, "A Real-Time Magnetoencephalography Brain-Computer Interface Using Interactive three-dimensional 3D-Visualization and the Hadoop Ecosystem", Journal of Brain Sciences, 2015.


Phase 1: REAL-TIME MEG BRAIN COMPUTER INTERFACE to HADOOP ECOSYSTEM
Real-Time BCI coupled to Hornet's Nest Flight Simulator
Subject's MEG Brain Signals
Machine Learning Classification Algorithm
MEG 274 channel Real-Time Processor
Subject's Performance on Real-Time BCI
Interface to Flight-Simulator
Analyze and Store all BCI videogame Analytics and MEG signals in Hadoop Ecosystem
Apache Hadoop Ecosystem
Ambari
Provisioning, Managing and Monitoring Hadoop Clusters
Sqoop
Data Exchange
Oozie
Workflow
Pig
Scripting
Mahout
Machine Learning
R Connectors
Statistics
Hive
SQL Query
HBASE
Hbase
Columnar Store
YARN Map Reduce v2
Distributed Processing Framework
Flume
Log Collector
Zookeeper
Coordination
HDFS
Hadoop Distributed File System
Exported Videogame Analytics analyzed in Pig and Store in Hadoop Ecosystem
Lawrence Livermore National Laboratory
Option:UCRL#
Option:Additional Information
22

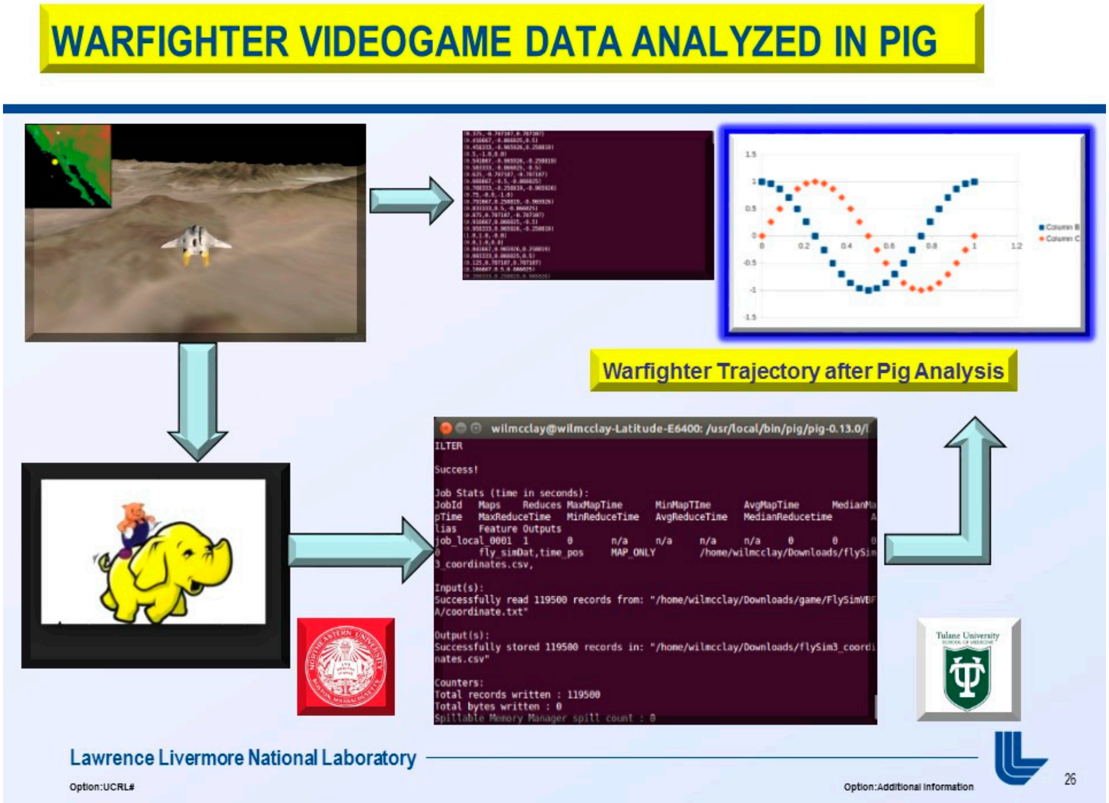
```
Trial 118 predicted as MotorCognition_20060921_01
Matches trigger action? Yes

Trial 119 predicted as MotorCognition_20060921_01
Matches trigger action? Yes

waiting for trial data.
waiting for trial data.
waiting for trial data.

Total trials tested: 120
Trials passed: 117
Trials failed: 2
Performance = 0.975
```

```
wilmcclay@wilmcclay-Latitude-E6400: /usr/local/bin/pig/pig-0.13.0/
men_movies.pig          pig                   women_movies.pig~
wilmcclay@wilmcclay-Latitude-E6400:/usr/local/bin/pig/pig-0.13.0/bin$ more Kumar
*.pig


A = load '/home/wilmcclay/HADOOP_HOME/hbase-install/hbase-0.98.3-hadoop2/bin/Kum
arAtul_MotorCognition_20060526_01-Leftepoched_a.txt' as elements;

B = foreach A generate TOMAP('elements',elements);

-- Store in HBase

store B into '/home/wilmcclay/HADOOP_HOME/hbase-install/hbase-0.98.3-hadoop2/bin
/KumarMEG'
using org.apache.pig.backend.hadoop.hbase.HBaseStorage('field:fields_Atrain');
wilmcclay@wilmcclay-Latitude-E6400:/usr/local/bin/pig/pig-0.13.0/bin$
wilmcclay@wilmcclay-Latitude-E6400:/usr/local/bin/pig/pig-0.13.0/bin$
wilmcclay@wilmcclay-Latitude-E6400:/usr/local/bin/pig/pig-0.13.0/bin$
wilmcclay@wilmcclay-Latitude-E6400:/usr/local/bin/pig/pig-0.13.0/bin$
wilmcclay@wilmcclay-Latitude-E6400:/usr/local/bin/pig/pig-0.13.0/bin$
wilmcclay@wilmcclay-Latitude-E6400:/usr/local/bin/pig/pig-0.13.0/bin$
wilmcclay@wilmcclay-Latitude-E6400:/usr/local/bin/pig/pig-0.13.0/bin$
wilmcclay@wilmcclay-Latitude-E6400:/usr/local/bin/pig/pig-0.13.0/bin$
wilmcclay@wilmcclay-Latitude-E6400:/usr/local/bin/pig/pig-0.13.0/bin$
```

Figure 6. Phase I, "A Real-Time Magnetoencephalography Brain-Computer Interface Using Interactive 3D-Visualization and the Hadoop Ecosystem", flowchart process of BCI analytics in the Hadoop Ecosystem.

From these signals, computational modeling allows a spatiotemporal view of the time course and spatial patterns of neuronal activity. The UCSF lab also uses digital 64-channel EEG and three-dimensional (3D) computing facilities [1].

The manuscript utilizes the Variational Bayesian Factor Analysis (VBFA) machine learning algorithm to classify and detect MEG/EEG brainwave classification [9,10] to define the user's intent while viewing a flight-simulator videogame in real-time [1], demonstrated in Figures 7 and 8. Furthermore, prestigious neuroscientists, such as Srikantan Nagarajan at University of San Francisco in California (UCSF), utilized location bias and spatial resolution in the reconstruction of a single dipole source utilizing various spatial filtering techniques in neuromagnetic imaging. The analysis of location bias for myriads of representative adaptive and non-adaptive spatial filters that are based on their resolution kernels validating standardized low-resolution electromagnetic tomography referenced as (sLORETA) using a minimum-variance spatial filter for MEG source reconstruction [9–14].



WARFIGHTER VIDEOGAME DATA ANALYZED IN PIG

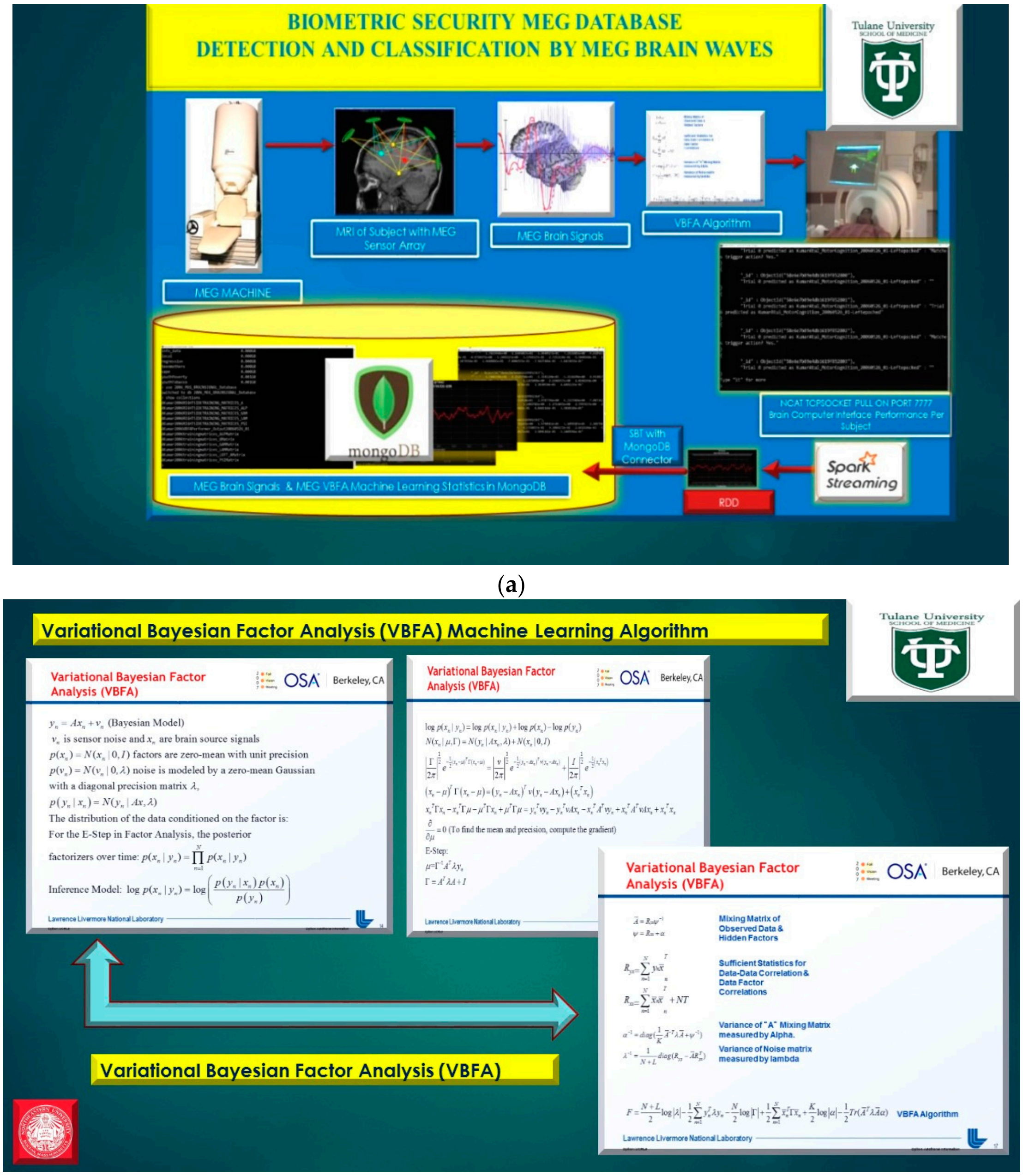
```
(0.375,-0.707107,0.707107)
(0.416667,-0.866025,0.5)
(0.458333,-0.965926,0.258819)
(0.5,-1.0,0.0)
(0.541667,-0.965926,-0.258819)
(0.583333,-0.866025,-0.5)
(0.625,-0.707107,-0.707107)
(0.666667,-0.5,-0.866025)
(0.708333,-0.258819,-0.965926)
(0.75,-0.0,-1.0)
(0.791667,0.258819,-0.965926)
(0.833333,0.5,-0.866025)
(0.875,0.707107,-0.707107)
(0.916667,0.866025,-0.5)
(0.958333,0.965926,-0.258819)
(1.0,1.0,-0.0)
(0.0,1.0,0.0)
(0.041667,0.965926,0.258819)
(0.083333,0.866025,0.5)
(0.125,0.707107,0.707107)
(0.166667,0.5,0.866025)
(0.208333,0.258819,0.965926)
```

Warfighter Trajectory after Pig Analysis

```
wilmcclay@wilmcclay-Latitude-E6400: /usr/local/bin/pig/pig-0.13.0/I
ILTER

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTIme      AvgMapTime      MedianMa
pTime   MaxReduceTime   MinReduceTime   AvgReduceTime   MedianReducetime        A
lias    Feature Outputs
job_local_0001  1       0       n/a     n/a     n/a     n/a     0       0       0
0       fly_simDat,time_pos     MAP_ONLY        /home/wilmcclay/Downloads/flySim
3_coordinates.csv,

Input(s):
Successfully read 119500 records from: "/home/wilmcclay/Downloads/game/FlySimVBF
A/coordinate.txt"

Output(s):
Successfully stored 119500 records in: "/home/wilmcclay/Downloads/flySim3_coordi
nates.csv"

Counters:
Total records written : 119500
Total bytes written : 0
Spillable Memory Manager spill count : 0
```

Tulane University
Lawrence Livermore National Laboratory
Option:UCRL#
Option:Additional Information
26

Figure 7. Phase I, "A Real-Time Magnetoencephalography Brain-Computer Interface Using Interactive 3D-Visualization and the Hadoop Ecosystem", Pig analysis for MEG Subject performance on Warfighter.



BIOMETRIC SECURITY MEG DATABASE
DETECTION AND CLASSIFICATION BY MEG BRAIN WAVES

Tulane University
SCHOOL OF MEDICINE
MEG MACHINE
MRI of Subject with MEG Sensor Array
MEG Brain Signals
VBFA Algorithm
NCAT TCPSOCKET PULL ON PORT 7777 Brain Computer Interface Performance Per Subject
Spark Streaming
RDD
SBT with MongoDB Connector
mongoDB
MEG Brain Signals & MEG VBFA Machine Learning Statistics in MongoDB

(**a**)

Variational Bayesian Factor Analysis (VBFA) Machine Learning Algorithm

Tulane University
SCHOOL OF MEDICINE

Variational Bayesian Factor Analysis (VBFA)

$y_n = Ax_n + v_n$ (Bayesian Model)

v_n is sensor noise and x_n are brain source signals

$p(x_n) = N(x_n \mid 0, I)$ factors are zero-mean with unit precision

$p(v_n) = N(v_n \mid 0, \lambda)$ noise is modeled by a zero-mean Gaussian with a diagonal precision matrix λ,

$p(y_n \mid x_n) = N(y_n \mid Ax, \lambda)$

The distribution of the data conditioned on the factor is:

For the E-Step in Factor Analysis, the posterior

factorizers over time: $p(x_n \mid y_n) = \prod_{n=1}^{N} p(x_n \mid y_n)$

Inference Model: $\log p(x_n \mid y_n) = \log\left(\frac{p(y_n \mid x_n)\, p(x_n)}{p(y_n)}\right)$

Lawrence Livermore National Laboratory

Variational Bayesian Factor Analysis (VBFA)

$\log p(x_n \mid y_n) = \log p(x_n \mid y_n) + \log p(x_n) - \log p(y_n)$

$N(x_n \mid \mu, \Gamma) = N(y_n \mid Ax_n, \lambda) + N(x_n \mid 0, I)$

$\left|\frac{\Gamma}{2\pi}\right|^{\frac{1}{2}} e^{-\frac{1}{2}(x_n-\mu)^T\Gamma(x_n-\mu)} = \left|\frac{v}{2\pi}\right|^{\frac{1}{2}} e^{-\frac{1}{2}(y_n-Ax_n)^T v(y_n-Ax_n)} + \left|\frac{I}{2\pi}\right|^{\frac{1}{2}} e^{-\frac{1}{2}(x_n^T x_n)}$

$(x_n-\mu)^T \Gamma (x_n-\mu) = (y_n - Ax_n)^T v (y_n - Ax_n) + (x_n^T x_n)$

$x_n^T \Gamma x_n - x_n^T \Gamma \mu - \mu^T \Gamma x_n + \mu^T \Gamma \mu = y_n^T v y_n - y_n^T v A x_n - x_n^T A^T v y_n + x_n^T A^T v A x_n + x_n^T x_n$

$\frac{\partial}{\partial \mu} = 0$ (To find the mean and precision, compute the gradient)

E-Step:

$\mu = \Gamma^{-1} A^T \lambda y_n$

$\Gamma = A^T \lambda A + I$

Lawrence Livermore National Laboratory

Variational Bayesian Factor Analysis (VBFA)

$\bar{A} = R_{yx}\psi^{-1}$
$\psi = R_{xx} + \alpha$ Mixing Matrix of Observed Data & Hidden Factors

$R_{yx} = \sum_{n=1}^{N} y_n \bar{x}_n^T$
$R_{xx} = \sum_{n=1}^{N} \bar{x}_n \bar{x}_n^T + NT$ Sufficient Statistics for Data-Data Correlation & Data Factor Correlations

$\alpha^{-1} = diag(\frac{1}{K}\bar{A}^{-T}\lambda\bar{A} + \psi^{-1})$ Variance of "A" Mixing Matrix measured by Alpha.

$\lambda^{-1} = \frac{1}{N+L} diag(R_{yy} - \bar{A}R_{yx}^T)$ Variance of Noise matrix measured by lambda

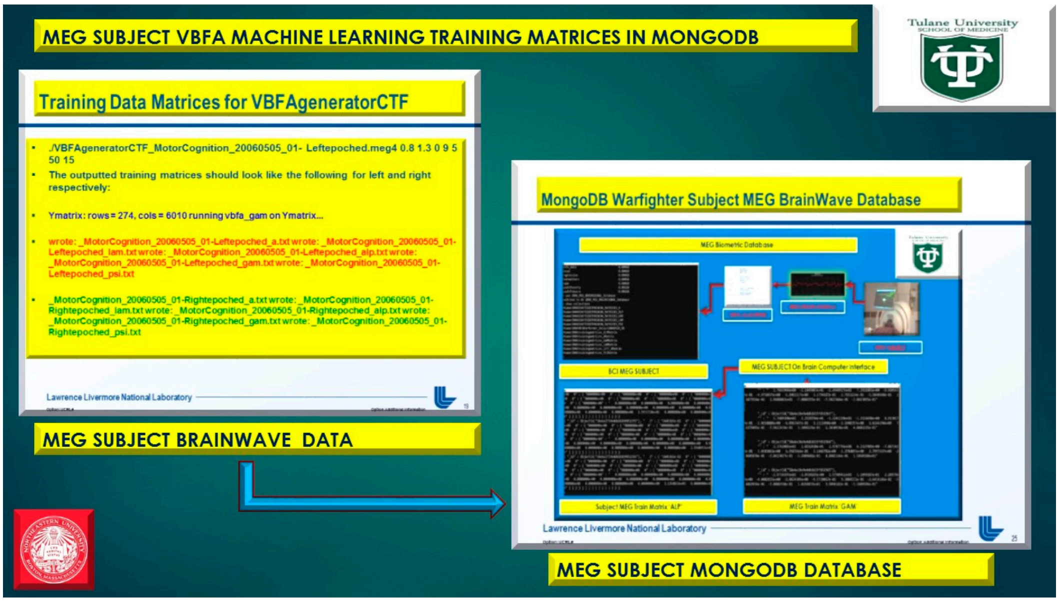
$F = \frac{N+L}{2}\log|\lambda| - \frac{1}{2}\sum_{n=1}^{N} y_n^T \lambda y_n - \frac{N}{2}\log|\Gamma| + \frac{1}{2}\sum_{n=1}^{N} \bar{x}_n^T \Gamma \bar{x}_n + \frac{K}{2}\log|\alpha| - \frac{1}{2}Tr(\bar{A}^T \lambda \bar{A}\alpha)$ VBFA Algorithm

Lawrence Livermore National Laboratory

Variational Bayesian Factor Analysis (VBFA)

(**b**)

Figure 8. *Cont.*



MEG SUBJECT VBFA MACHINE LEARNING TRAINING MATRICES IN MONGODB

Training Data Matrices for VBFAgeneratorCTF

- ./VBFAgeneratorCTF_MotorCognition_20060505_01- Leftepoched.meg4 0.8 1.3 0 9 5 50 15
- The outputted training matrices should look like the following for left and right respectively:
- Ymatrix: rows = 274, cols = 6010 running vbfa_gam on Ymatrix...
- wrote: _MotorCognition_20060505_01-Leftepoched_a.txt wrote: _MotorCognition_20060505_01-Leftepoched_lam.txt wrote: _MotorCognition_20060505_01-Leftepoched_alp.txt wrote: _MotorCognition_20060505_01-Leftepoched_gam.txt wrote: _MotorCognition_20060505_01-Leftepoched_psi.txt
- _MotorCognition_20060505_01-Rightepoched_a.txt wrote: _MotorCognition_20060505_01-Rightepoched_lam.txt wrote: _MotorCognition_20060505_01-Rightepoched_alp.txt wrote: _MotorCognition_20060505_01-Rightepoched_gam.txt wrote: _MotorCognition_20060505_01-Rightepoched_psi.txt

Lawrence Livermore National Laboratory

MEG SUBJECT BRAINWAVE DATA

MongoDB Warfighter Subject MEG BrainWave Database

MEG Biometric Database

BCI MEG SUBJECT

MEG SUBJECT On Brain Computer Interface

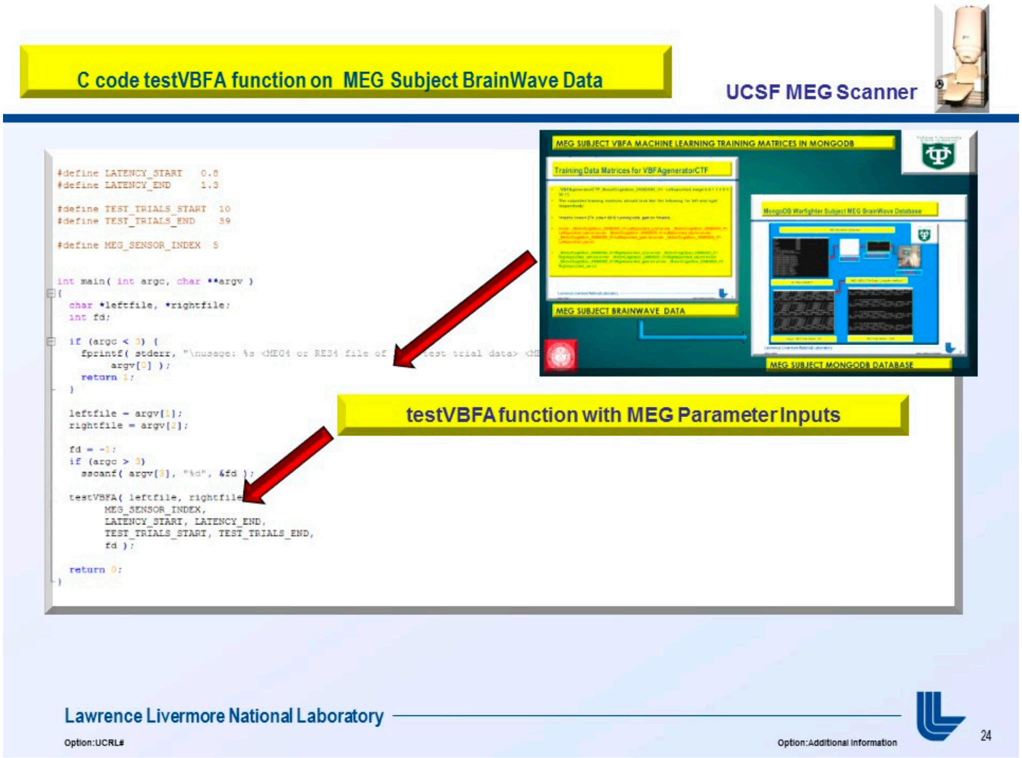
Subject MEG Train Matrix 'ALP'

MEG Train Matrix 'GAM'

Lawrence Livermore National Laboratory

MEG SUBJECT MONGODB DATABASE

(**c**)



C code testVBFA function on MEG Subject BrainWave Data

UCSF MEG Scanner

```
#define LATENCY_START   0.8
#define LATENCY_END     1.3

#define TEST_TRIALS_START  10
#define TEST_TRIALS_END    39

#define MEG_SENSOR_INDEX  5


int main( int argc, char **argv )
{
  char *leftfile, *rightfile;
  int fd;

  if (argc < 3) {
    fprintf( stderr, "\nusage: %s <MEG4 or RES4 file of [illegible] test trial data> <M
         argv[0] );
    return 1;
  }

  leftfile = argv[1];
  rightfile = argv[2];

  fd = -1;
  if (argc > 3)
    sscanf( argv[3], "%d", &fd );

  testVBFA( leftfile, rightfile,
        MEG_SENSOR_INDEX,
        LATENCY_START, LATENCY_END,
        TEST_TRIALS_START, TEST_TRIALS_END,
        fd );

  return 0;
}
```

testVBFA function with MEG Parameter Inputs

Lawrence Livermore National Laboratory

Option:UCRL#

Option:Additional Information

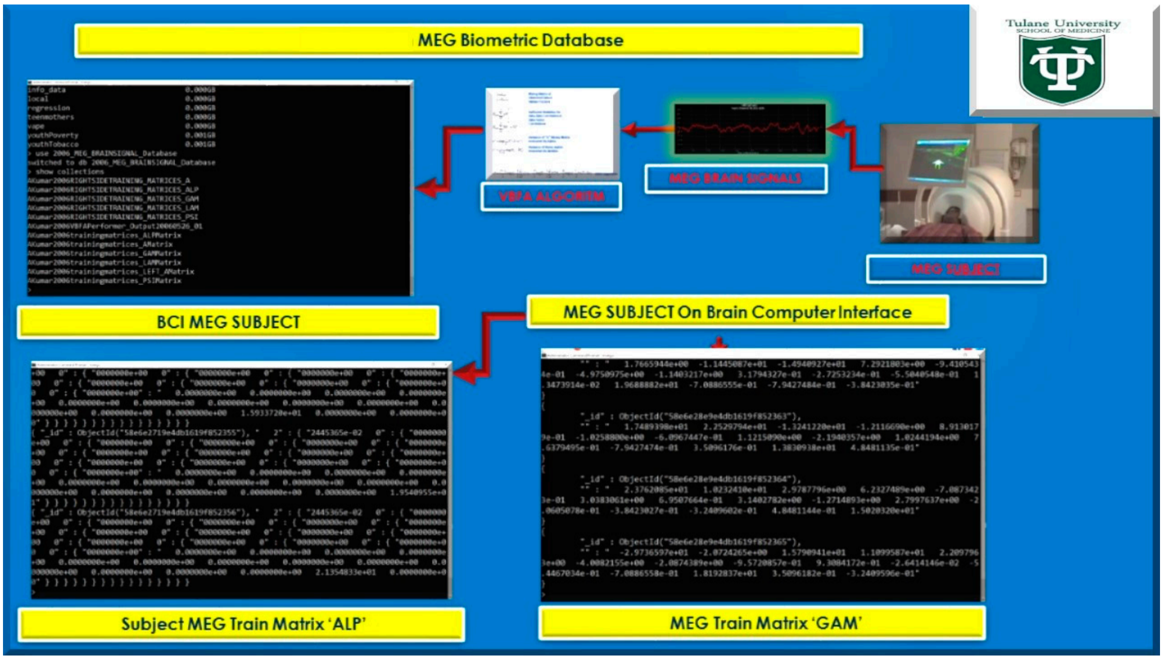
MEG SUBJECT VBFA MACHINE LEARNING TRAINING MATRICES IN MONGODB
Training Data Matrices for VBFAgeneratorCTF
MEG SUBJECT BRAINWAVE DATA
MongoDB Warfighter Subject MEG BrainWave Database
MEG SUBJECT MONGODB DATABASE
24

(**d**)

Figure 8. (**a**) Phase II, MongoDB Magnetoencephalography Brain-Computer Interface Database. (**b**) Phase II, Variational Bayesian Factor Analysis (VBFA) Machine Learning Algorithm. (**c**) Phase II, MEG Subject Brain Wave Data and VBFAgeneratorCTF training matrices in MongoDBdatabase(s). (**d**) Phase II, C code testVBFA function on MEG Subject Brainwave Data.

3. Phase II: Wireless EEG MongoDB & Cassandra Brain Computer Interface Databases and iOS Applications

The Phase II integrates with Phase I utilizing the same machine learning classifier as in Phase I project begins with the use of a wireless Emotiv Epoch EEG headset integrated to MongoDB and Cassandra NoSQL databases with iOS Mobile Applications, demonstrated in Figures 9–11. Figure 12, illustrates the overall and analogous architecture of the Phase II NAZZY with Frozen Video Game BCI process. The Emotiv Epoch EEG headset was designed by Emotiv Systems the predominant leader in commercial Brain Computer Interface technology. The Emotiv Epoch system was utilized to measure the electrical activity that is associated with the brain and muscles of the face and it converts brain signals and activity into control signals [1,10]. The Emotiv Epoch implements Artificial Neural Networks based learning and training techniques while using the McCulloch-Pitts model and Back-Propagation Neural Network algorithm [10].

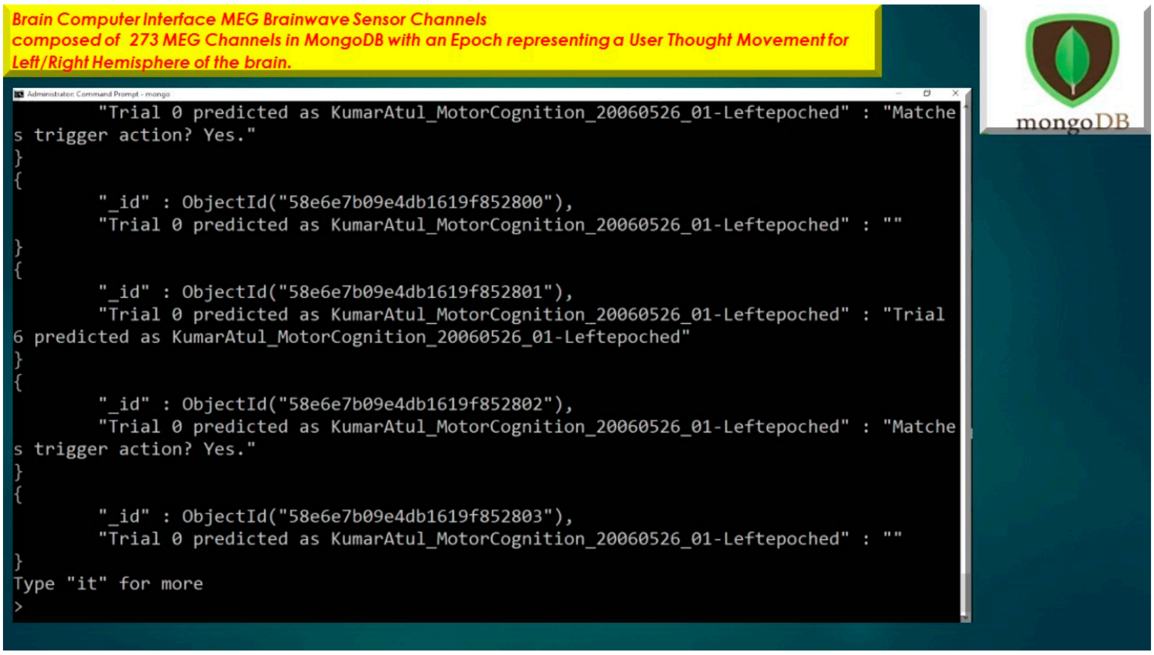

MEG Biometric Database
Tulane University School of Medicine
VBFA ALGORITHM
MEG BRAIN SIGNALS
MEG SUBJECT
BCI MEG SUBJECT
MEG SUBJECT On Brain Computer Interface
Subject MEG Train Matrix 'ALP'
MEG Train Matrix 'GAM'

```
info_data                        0.000GB
local                            0.000GB
regression                       0.000GB
teenmothers                      0.000GB
vape                             0.000GB
youthPoverty                     0.001GB
youthTobacco                     0.001GB
> use 2006_MEG_BRAINSIGNAL_Database
switched to db 2006_MEG_BRAINSIGNAL_Database
> show collections
AKumar2006RIGHTSIDETRAINING_MATRICES_A
AKumar2006RIGHTSIDETRAINING_MATRICES_ALP
AKumar2006RIGHTSIDETRAINING_MATRICES_GAM
AKumar2006RIGHTSIDETRAINING_MATRICES_LAM
AKumar2006RIGHTSIDETRAINING_MATRICES_PSI
AKumar2006VBFAPerformer_Output20060526_01
AKumar2006trainingmatrices_ALPMatrix
AKumar2006trainingmatrices_AMatrix
AKumar2006trainingmatrices_GAMMatrix
AKumar2006trainingmatrices_LAMMatrix
AKumar2006trainingmatrices_LEFT_AMatrix
AKumar2006trainingmatrices_PSIMatrix
>
```

```
+00   0" : { "0000000e+00   0" : { "0000000e+00   0" : { "0000000e+00   0" : { "0000000e+
00   0" : { "0000000e+00   0" : { "0000000e+00   0" : { "0000000e+00   0" : { "0000000e+0
0   0" : { "0000000e+00" : "   0.0000000e+00   0.0000000e+00   0.0000000e+00   0.0000000e
+00   0.0000000e+00   0.0000000e+00   0.0000000e+00   0.0000000e+00   0.0000000e+00   0.0
000000e+00   0.0000000e+00   0.0000000e+00   1.5933720e+01   0.0000000e+00   0.0000000e+0
0" } } } } } } } } } } } } } } } } } }
{ "_id" : ObjectId("58e6e2719e4db1619f852355"), "   2" : { "2445365e-02   0" : { "0000000
e+00   0" : { "0000000e+00   0" : { "0000000e+00   0" : { "0000000e+00   0" : { "0000000e
+00   0" : { "0000000e+00   0" : { "0000000e+00   0" : { "0000000e+00   0" : { "0000000e+
00   0" : { "0000000e+00   0" : { "0000000e+00   0" : { "0000000e+00   0" : { "0000000e+0
0   0" : { "0000000e+00" : "   0.0000000e+00   0.0000000e+00   0.0000000e+00   0.0000000e
+00   0.0000000e+00   0.0000000e+00   0.0000000e+00   0.0000000e+00   0.0000000e+00   0.0
000000e+00   0.0000000e+00   0.0000000e+00   0.0000000e+00   0.0000000e+00   1.9540955e+0
1" } } } } } } } } } } } } } } } } } }
{ "_id" : ObjectId("58e6e2719e4db1619f852356"), "   2" : { "2445365e-02   0" : { "0000000
e+00   0" : { "0000000e+00   0" : { "0000000e+00   0" : { "0000000e+00   0" : { "0000000e
+00   0" : { "0000000e+00   0" : { "0000000e+00   0" : { "0000000e+00   0" : { "0000000e+
00   0" : { "0000000e+00   0" : { "0000000e+00   0" : { "0000000e+00   0" : { "0000000e+0
0   0" : { "0000000e+00" : "   0.0000000e+00   0.0000000e+00   0.0000000e+00   0.0000000e
+00   0.0000000e+00   0.0000000e+00   0.0000000e+00   0.0000000e+00   0.0000000e+00   0.0
000000e+00   0.0000000e+00   0.0000000e+00   0.0000000e+00   2.1354833e+01   0.0000000e+0
0" } } } } } } } } } } } } } } } } } }
>
```

```
        "" : "   1.7665944e+00  -1.1445087e+01  -1.4940927e+01   7.2921803e+00  -9.410543
4e-01  -4.9750975e+00  -1.1403217e+00   3.1794327e-01  -2.7253234e-01  -5.5040548e-01   1
.34739l4e-02   1.9688882e+01  -7.0886555e-01  -7.9427484e-01  -3.8423035e-01"
}
{
        "_id" : ObjectId("58e6e28e9e4db1619f852363"),
        "" : "   1.7489398e+01   2.2529794e+01  -1.3241220e+01  -1.2116690e+00   8.913017
9e-01  -1.0258800e+00  -6.0967447e-01   1.1215090e+00  -2.1940357e+00   1.0244194e+00   7
.63794095e-01  -7.9427474e-01   3.5096176e-01   1.3830938e+01   4.8481135e-01"
}
{
        "_id" : ObjectId("58e6e28e9e4db1619f852364"),
        "" : "   2.3762085e+01   1.0232410e+01   2.9787796e+00   6.2327489e+00  -7.087342
3e-01   3.0383061e+00   6.9507664e-01   3.1402782e+00  -1.2714893e+00   2.7997637e+00  -2
.0605078e-01  -3.8423027e-01  -3.2409602e-01   4.8481144e-01   1.5020320e+01"
}
{
        "_id" : ObjectId("58e6e28e9e4db1619f852365"),
        "" : "  -2.9736597e+01  -2.0724265e+00   1.5790941e+01   1.1099587e+01   2.209796
3e+00  -4.0082155e+00  -2.0874389e+00  -9.5720857e-01   9.3084172e-01  -2.6414146e-02  -5
.44670034e-01  -7.0886558e-01   1.8192837e+01   3.5096182e-01  -3.2409596e-01"
}
>
```

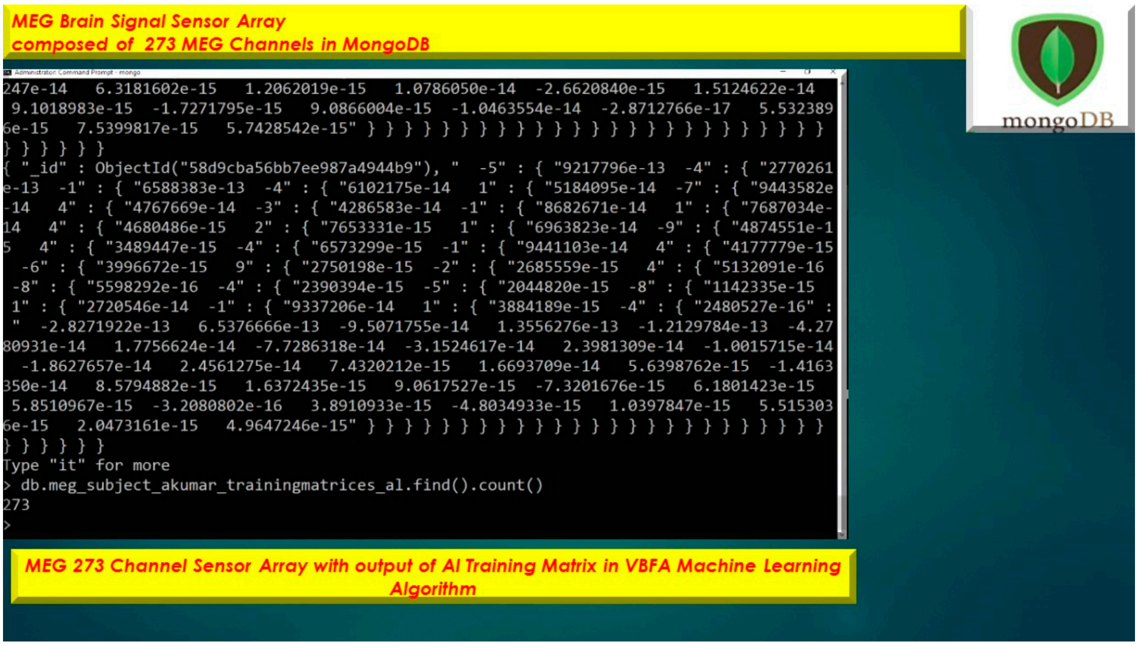
Figure 9. Phase II, MongoDB Magnetoencephalography Brain-Computer Interface Database storage of MEG Subject Variational Bayesian Factor Analysis training matrices and MEG Subject Performance and Metadata.



Brain Computer Interface MEG Brainwave Sensor Channels composed of 273 MEG Channels in MongoDB with an Epoch representing a User Thought Movement for Left/Right Hemisphere of the brain.

```
        "Trial 0 predicted as KumarAtul_MotorCognition_20060526_01-Leftepoched" : "Matche
s trigger action? Yes."
}
{
        "_id" : ObjectId("58e6e7b09e4db1619f852800"),
        "Trial 0 predicted as KumarAtul_MotorCognition_20060526_01-Leftepoched" : ""
}
{
        "_id" : ObjectId("58e6e7b09e4db1619f852801"),
        "Trial 0 predicted as KumarAtul_MotorCognition_20060526_01-Leftepoched" : "Trial
6 predicted as KumarAtul_MotorCognition_20060526_01-Leftepoched"
}
{
        "_id" : ObjectId("58e6e7b09e4db1619f852802"),
        "Trial 0 predicted as KumarAtul_MotorCognition_20060526_01-Leftepoched" : "Matche
s trigger action? Yes."
}
{
        "_id" : ObjectId("58e6e7b09e4db1619f852803"),
        "Trial 0 predicted as KumarAtul_MotorCognition_20060526_01-Leftepoched" : ""
}
Type "it" for more
>
```

Figure 10. MEG Brainwave data acquisition in MongoDB with 12-byte BSON timestamp representing ObjectID for Epoch Trial performance for MEG Subject.

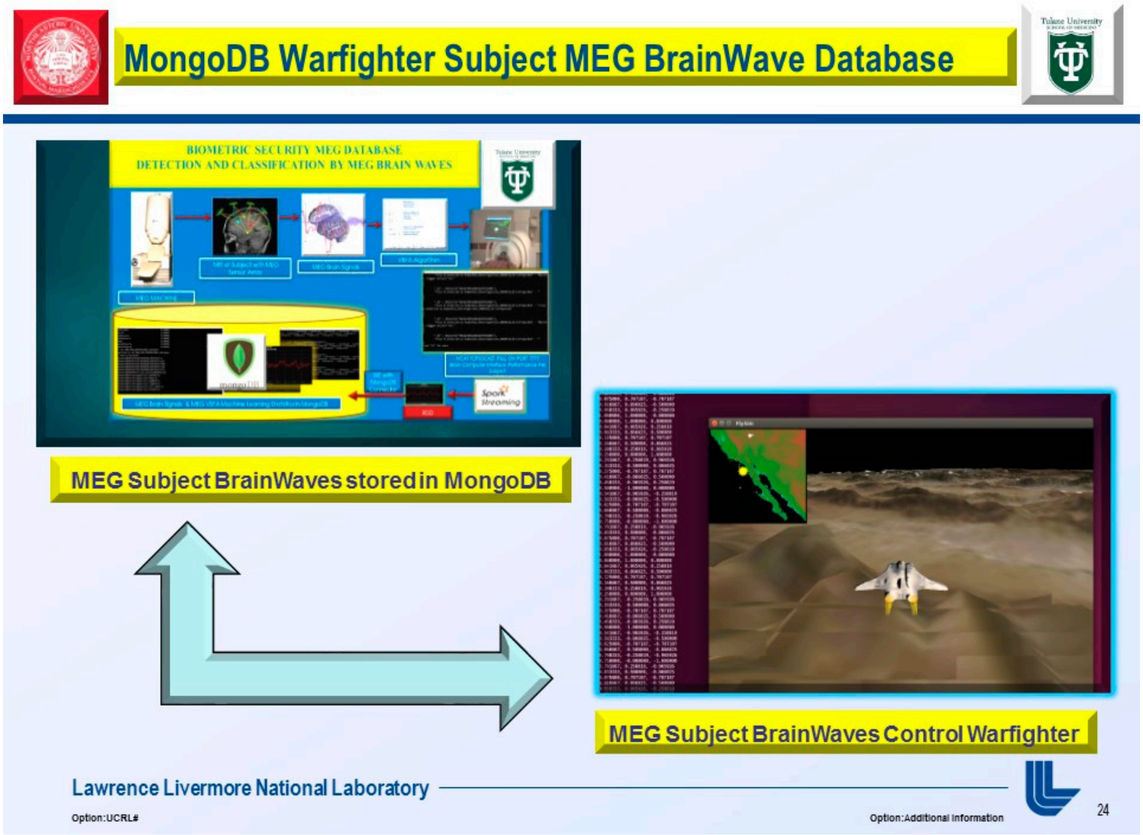


MEG Brain Signal Sensor Array
composed of 273 MEG Channels in MongoDB

```
247e-14   6.3181602e-15   1.2062019e-15   1.0786050e-14  -2.6620840e-15   1.5124622e-14
 9.1018983e-15  -1.7271795e-15   9.0866004e-15  -1.0463554e-14  -2.8712766e-17   5.532389
6e-15   7.5399817e-15   5.7428542e-15" } } } } } } } } } } } } } } } } } } } } } } } } } } }
} } } } } }
{ "_id" : ObjectId("58d9cba56bb7ee987a4944b9"), "  -5" : { "9217796e-13  -4" : { "2770261
e-13  -1" : { "6588383e-13  -4" : { "6102175e-14   1" : { "5184095e-14  -7" : { "9443582e
-14   4" : { "4767669e-14  -3" : { "4286583e-14  -1" : { "8682671e-14   1" : { "7687034e-
14   4" : { "4680486e-15   2" : { "7653331e-15   1" : { "6963823e-14  -9" : { "4874551e-1
5   4" : { "3489447e-15  -4" : { "6573299e-15  -1" : { "9441103e-14   4" : { "4177779e-15
  -6" : { "3996672e-15   9" : { "2750198e-15  -2" : { "2685559e-15   4" : { "5132091e-16
 -8" : { "5598292e-16  -4" : { "2390394e-15  -5" : { "2044820e-15  -8" : { "1142335e-15
 1" : { "2720546e-14  -1" : { "9337206e-14   1" : { "3884189e-15  -4" : { "2480527e-16" :
 "  -2.8271922e-13   6.5376666e-13  -9.5071755e-14   1.3556276e-13  -1.2129784e-13  -4.27
80931e-14   1.7756624e-14  -7.7286318e-14  -3.1524617e-14   2.3981309e-14  -1.0015715e-14
  -1.8627657e-14   2.4561275e-14   7.4320212e-15   1.6693709e-14   5.6398762e-15  -1.4163
350e-14   8.5794882e-15   1.6372435e-15   9.0617527e-15  -7.3201676e-15   6.1801423e-15
 5.8510967e-15  -3.2080802e-16   3.8910933e-15  -4.8034933e-15   1.0397847e-15   5.515303
6e-15   2.0473161e-15   4.9647246e-15" } } } } } } } } } } } } } } } } } } } } } } } } } } }
} } } } } }
Type "it" for more
> db.meg_subject_akumar_trainingmatrices_al.find().count()
273
>
```

MEG 273 Channel Sensor Array with output of AI Training Matrix in VBFA Machine Learning Algorithm

(**a**)



MongoDB Warfighter Subject MEG BrainWave Database

Tulane University
BIOMETRIC SECURITY MEG DATABASE
DETECTION AND CLASSIFICATION BY MEG BRAIN WAVES
MEG MACHINE
MRI of Subject with MEG Sensor Array
MEG Brain Signals
VBFA Algorithm
mongoDB
Spark Streaming
RDD

MEG Subject BrainWaves stored in MongoDB

MEG Subject BrainWaves Control Warfighter

Lawrence Livermore National Laboratory

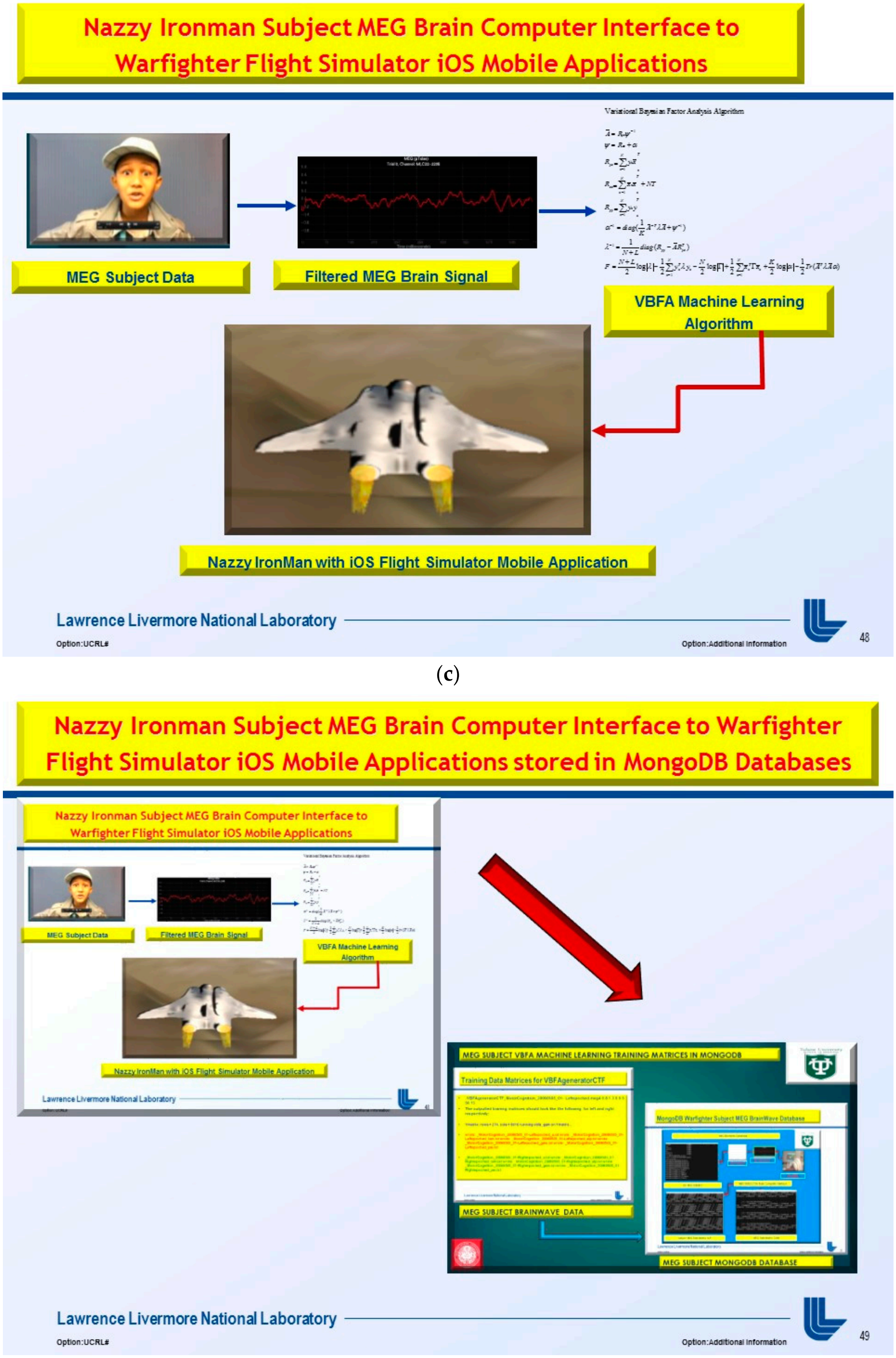
Option:UCRL#

Option:Additional Information

24

(**b**)

Figure 11. *Cont.*



Nazzy Ironman Subject MEG Brain Computer Interface to Warfighter Flight Simulator iOS Mobile Applications

Variational Bayesian Factor Analysis Algorithm

$$\bar{\lambda} = R_{yx}\psi^{-1}$$
$$\psi = R_{xx} + \alpha$$
$$R_{yx} = \sum_{n=1}^{N} y_n \bar{x}_n^T$$
$$R_{xx} = \sum_{n=1}^{N} \bar{x}_n \bar{x}_n^T + N\Gamma$$
$$R_{yy} = \sum_{n=1}^{N} y_n y_n^T$$
$$\alpha^{-1} = diag\left(\frac{1}{K}\bar{\lambda}^T\lambda\bar{\lambda} + \psi^{-1}\right)$$
$$\lambda^{-1} = \frac{1}{N+L} diag\left(R_{yy} - \bar{\lambda}R_{yx}^T\right)$$
$$F = \frac{N+L}{2}\log|\lambda| - \frac{1}{2}\sum_{n=1}^{N} y_n^T\lambda y_n - \frac{N}{2}\log|\Gamma| + \frac{1}{2}\sum_{n=1}^{N} \bar{x}_n^T\Gamma\bar{x}_n + \frac{K}{2}\log|\alpha| - \frac{1}{2}Tr(\bar{\lambda}^T\lambda\bar{\lambda}\alpha)$$

MEG Subject Data
Filtered MEG Brain Signal
VBFA Machine Learning Algorithm
Nazzy IronMan with iOS Flight Simulator Mobile Application

Lawrence Livermore National Laboratory

Option:UCRL#

Option:Additional Information

48

(**c**)

Nazzy Ironman Subject MEG Brain Computer Interface to Warfighter Flight Simulator iOS Mobile Applications stored in MongoDB Databases

Nazzy Ironman Subject MEG Brain Computer Interface to Warfighter Flight Simulator iOS Mobile Applications

MEG Subject Data
Filtered MEG Brain Signal
VBFA Machine Learning Algorithm
Nazzy IronMan with iOS Flight Simulator Mobile Application
Lawrence Livermore National Laboratory
MEG SUBJECT VBFA MACHINE LEARNING TRAINING MATRICES IN MONGODB
Training Data Matrices for VBFAgeneratorCTF
MEG SUBJECT BRAINWAVE DATA
MongoDB Warfighter Subject MEG BrainWave Database
MEG SUBJECT MONGODB DATABASE

- The outputted learning matrices should look like the the following for left and right respectively:

Lawrence Livermore National Laboratory

Option:UCRL#

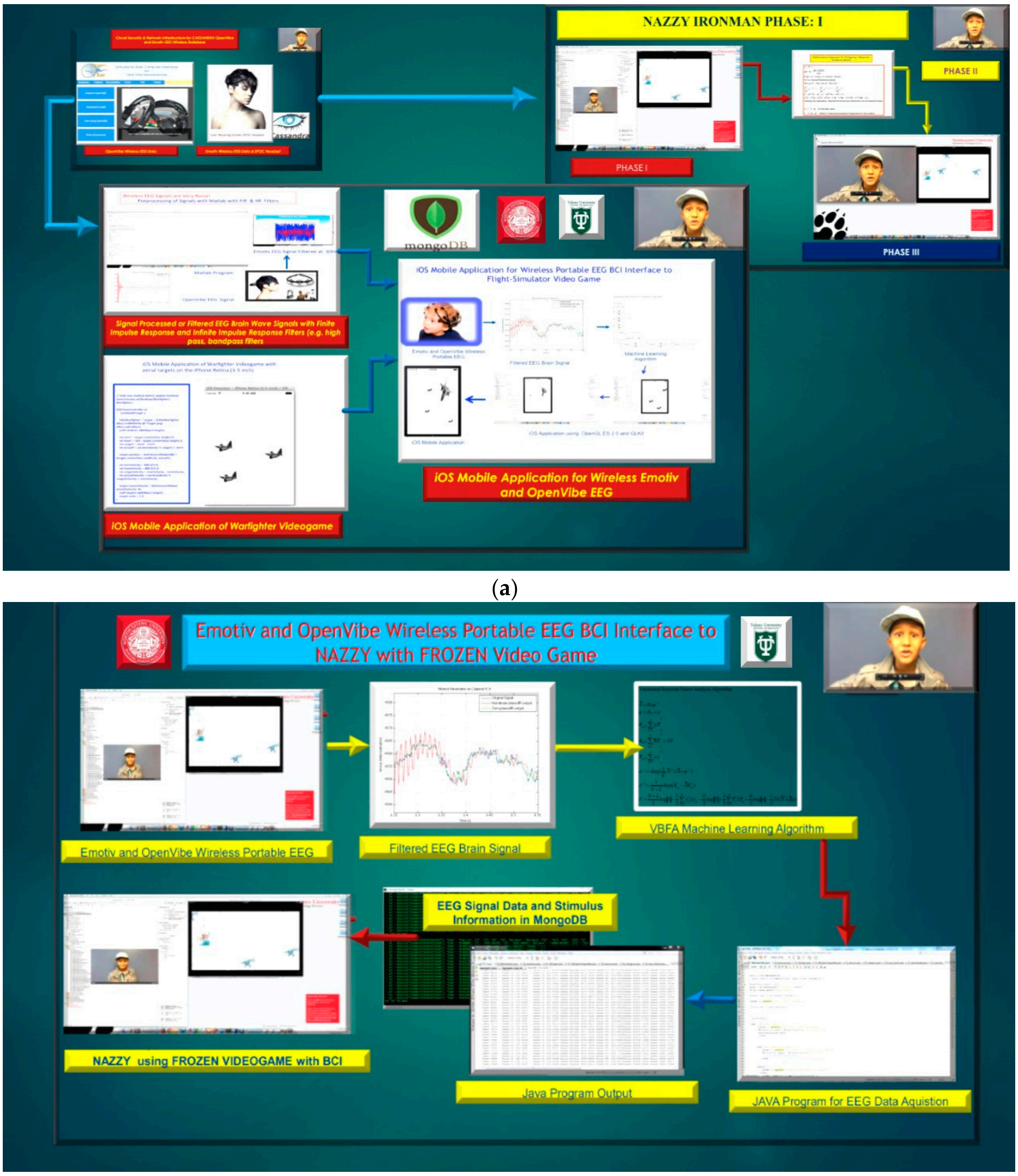
Option:Additional Information

49

(**d**)

Figure 11. (**a**) MEG Brainwave data acquisition in MongoDB with 12-byte BSON timestamp representing ObjectID representing Subject's Training Matrices acquired during VBFA Machine learning algorithm training on MEG brainwaves. (**b**) MEG Brainwave data acquisition in MongoDB with 12-byte

BSON timestamp representing ObjectID representing with Subject Brainwaves controlling flight of Warfighter simulation. (**c**) Nazzy Ironman Subject MEG Brain Computer Interface to Warfighter Flight Simulator iOS Mobile Applications yielding over 90% performance on MEG Subject brain signal data. (**d**) Nazzy Ironman Subject MEG Brain Computer Interface to Warfighter Flight Simulator iOS Mobile Applications stored in MongoDB databases yielding over 90% performance on Subject Data, demonstrated in Figures 9–11.



Cloud Security & Network Infrastructure for CASSANDRA OpenVibe and Emotiv EEG Wireless Database

OpenVibe Wireless EEG Data

Emotiv Wireless EEG Data & EPOC Headset

NAZZY IRONMAN PHASE: I

PHASE I

PHASE II

PHASE III

Wireless EEG Signals are Very Noisy!
Preprocessing of Signals with Matlab with FIR & IIR Filters

Emotiv EEG Signal Filtered at 30Hz

Matlab Program

OpenVibe EEG Signal

Signal Processed or Filtered EEG Brain Wave Signals with Finite Impulse Response and Infinite Impulse Response Filters (e.g. high pass, bandpass filters

iOS Mobile Application of Warfighter Videogame with aerial targets on the iPhone Retina (3.5 inch)

iOS Mobile Application of Warfighter Videogame

iOS Mobile Application for Wireless Portable EEG BCI Interface to Flight-Simulator Video Game

Emotiv and OpenVibe Wireless Portable EEG

Filtered EEG Brain Signal

Machine Learning Algorithm

iOS Mobile Application

iOS Application using OpenGL ES 2.0 and GLKit

iOS Mobile Application for Wireless Emotiv and OpenVibe EEG

(**a**)

Emotiv and OpenVibe Wireless Portable EEG BCI Interface to NAZZY with FROZEN Video Game

Emotiv and OpenVibe Wireless Portable EEG

Filtered EEG Brain Signal

VBFA Machine Learning Algorithm

EEG Signal Data and Stimulus Information in MongoDB

NAZZY using FROZEN VIDEOGAME with BCI

Java Program Output

JAVA Program for EEG Data Aquistion

(**b**)

Figure 12. (**a**) NAZZY IronMan with Frozen Videogame & iOS Warfighter Mobile Game for Brain Computer Interface Project with Emotiv/OpenVibe Wireless electroencephalography (EEG) brain signal(s) data while using machine learning algorithms to classify brain signals in iOS videogame applications utilizing EEG brain signal data storage in NoSQL database MongoDB. (**b**) NAZZY IronMan with Frozen Project with Emotiv Wireless EEG brain signal(s) data using machine learning algorithms to classify brain signals in iOS Frozen videogame utilizing EEG brain signal data storage in NoSQL database MongoDB.

3.1. EEG Data Acquisition and Signal Processing

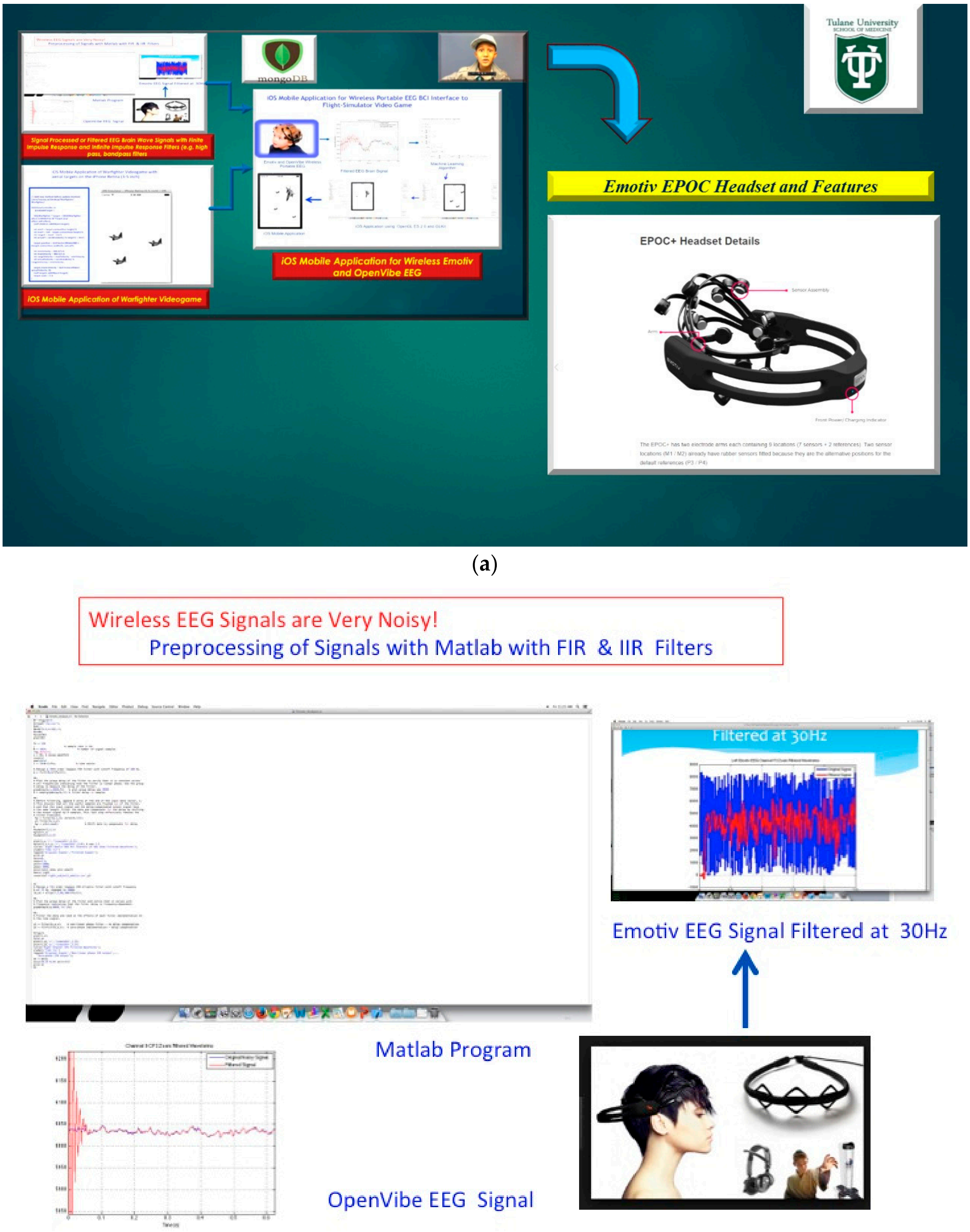
The aim of the wireless BCI research assumes that it is possible to assess thought movements of individuals that are induced by specific events in the virtual environment while using EEG and videogame analysis. In it is stated mental thoughts which are characteristics of user intent can be classified using machine learning algorithms, while the subject's brain waves are being acquired via wireless EEG, and are essential in this research for measuring multiple states from left-right thought movement. Electroencephalography (EEG) is the recording of electrical brain activity, which is caused by the firing of neurons within the brain. The EEG brain signals and machine learning/signal processing feature extraction are used to classify the EEG signal will be used as a control signal for the warfighter simulator or NAZZY Frozen Video game. The classification of EEG brainwave data illustrating multiple states of left-right thought movement are detected and analyzed with novel machine learning algorithms. The output of the subject's analytics while playing the flight simulator videogame or NAZZY Frozen Videogame post signal classification is recorded and warehoused in a NoSQL Cassandra and MongoDB database environment.

The dilemma with wireless EEG is that the sensors typically may have a very poor signal-to-noise ratio (e.g., lower monetary EEG sensor array cost may yield poor conductivity) and the utilization of proprietary software (e.g., Emotiv Back-Propagation Neural Networks) are often necessary to pre-process the wireless EEG brain-wave signals before the VBFA machine learning classifier can be applied, see in Figure 13 [15]. The lack of conductivity from EEG sensor array as opposed to MEG sensor arrays are based on a higher remuneration cost for design for brain signal acquisition. In the human brain, the current is generated mostly by pumping the positive ions of sodium, potassium, calcium, and the negative ion of chlorine, through the neuron membranes in the direction that is governed by the membrane potential [16]. Thus, an EEG signal is a measurement of currents during synaptic excitations of neurons in the cerebral cortex. The human head pertains to three different layers inclusive of the skull, scalp, and brain, and myriads of subsequent thin layers. The lack of conductivity from EEG sensors to acquisition brain signals is due to the skull attenuating the signals approximately one hundred times more than soft tissue [10,16].

EEG is the quintessential tool to study and diagnosis myriads of neurological disorders and other brain related abnormalities. The utilization of EEG Brain Computer Interface technology has myriads of applications to the following [1,17]:

Brain-Machine Interfaces

- Pilots and flight control
- Vigilance monitoring for air force, navy, or ground troop vehicles
- Clinical settings: Monitoring patient mental states and providing feedback
- Education: Improving vigilance, attention, learning, and memory
- Monitoring mental processes ("reading the mind")
- Detecting deception (FBI, CIA, other law enforcement agencies)
- Predicting behavior
- Detecting brain-based predispositions to certain mental tendencies (the brain version of Myers-Briggs)
- Likelihood of improving with one type of training versus another
- Likelihood of performing better under specific circumstances



Wireless EEG Signals are Very Noisy! Preprocessing of Signals with Matlab with FIR & IIR Filters

Signal Processed or Filtered EEG Brain Wave Signals with Finite Impulse Response and Infinite Impulse Response Filters (e.g. high pass, bandpass filters

iOS Mobile Application of Warfighter Videogame with aerial targets on the iPhone Retina (4-5 inch)

iOS Mobile Application of Warfighter Videogame

iOS Mobile Application for Wireless Portable EEG BCI Interface to Flight-Simulator Video Game

iOS Mobile Application for Wireless Emotiv and OpenVibe EEG

Emotiv EPOC Headset and Features

EPOC+ Headset Details

The EPOC+ has two electrode arms each containing 9 locations (7 sensors + 2 references). Two sensor locations (M1 / M2) already have rubber sensors fitted because they are the alternative positions for the default references (P3 / P4)

mongoDB
Tulane University
SCHOOL OF MEDICINE
Emotiv and OpenVibe Wireless Portable EEG
Filtered EEG Brain Signal
Machine Learning Algorithm
iOS Mobile Application
iOS Application using OpenGL ES 2.0 and GLKit
Sensor Assembly
Arms
Front Power/ Charging Indicator

(**a**)

Wireless EEG Signals are Very Noisy!
Preprocessing of Signals with Matlab with FIR & IIR Filters

(**b**)

Figure 13. (**a**) Emotiv EPOC Headset, Features, and Brain Computer Interface applications. (**b**) Utilization of Matlab FIR (Finite Impulse Response) & IIR (Infinite Impulse Response) Bandpass and Lowpass Filters on Wireless EEG Signals.

3.2. EEG Cassandra NoSQL Databases

Apache Cassandra is a proven high availability and scalable NoSQL database without diminishing in performance. The NoSQL database, Cassandra, is "suitable for applications that can't afford to lose data, even when an entire data center goes down [17]”. Therefore, with respect to Brain Computer Interface applications where brainwave signals are acquired in an EEG/MEG electrode array referencing each sensor channel often in real-time. The utilization of the Cassandra NoSQL column-oriented architecture database is a quintessential solution due to fault-tolerance, decentralization indicating no individual failure points and that every node is identical with data automatically replicated [18,19]. The usage of elasticity is a necessity with the Cassandra NoSQL database, since read and write output have linearity as new BCI machines could be added to the cloud network, as illustrated in Figures 14 and 15, below discussing beneficial cloud security constraints.

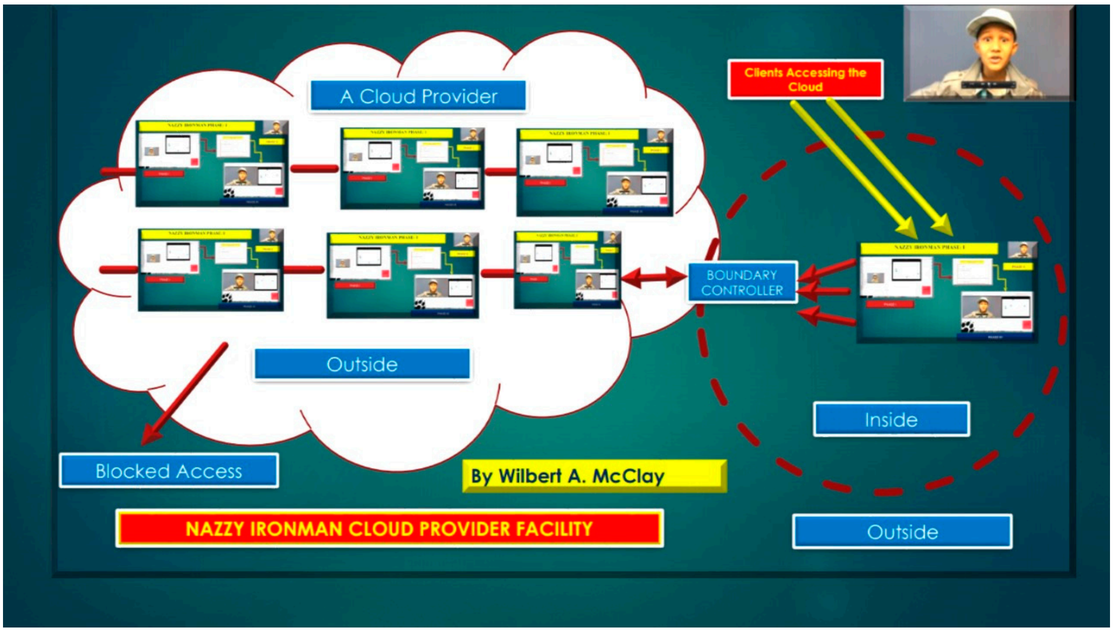
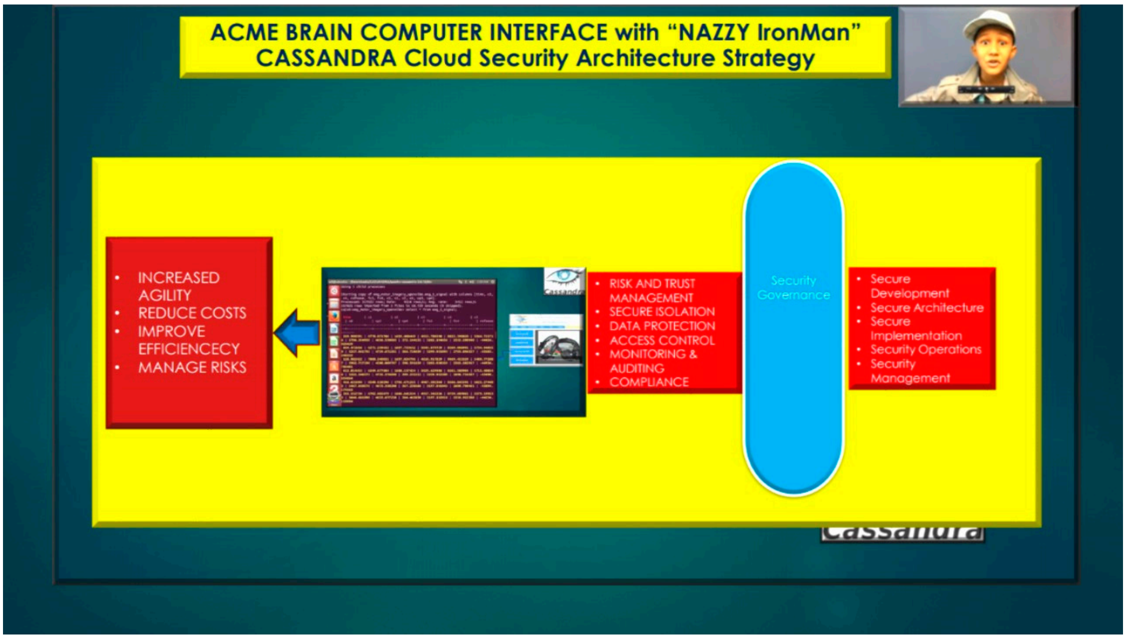


Figure 14. Nazzy IronMan Brain Computer Interface Cloud Provider Facility with Cassandra NoSQL database(s).


ACME BRAIN COMPUTER INTERFACE with “NAZZY IronMan” CASSANDRA Cloud Security Architecture Strategy

- INCREASED AGILITY
- REDUCE COSTS
- IMPROVE EFFICIENCECY
- MANAGE RISKS

- RISK AND TRUST MANAGEMENT
- SECURE ISOLATION
- DATA PROTECTION
- ACCESS CONTROL
- MONITORING & AUDITING
- COMPLIANCE

Security Governance

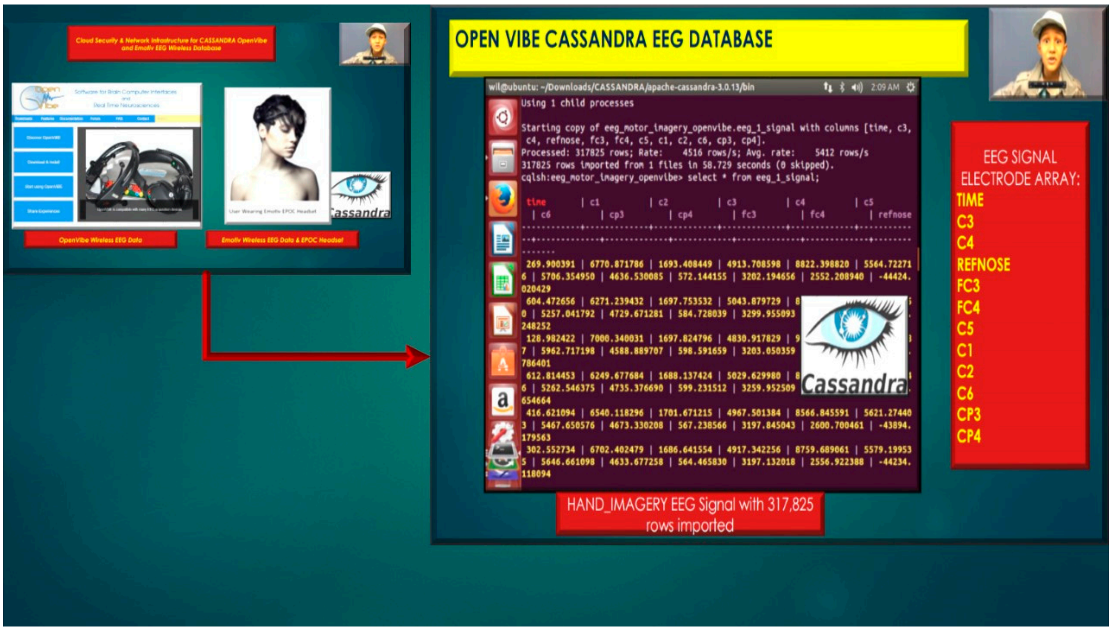
- Secure Development
- Secure Architecture
- Secure Implementation
- Security Operations
- Security Management

Figure 15. Nazzy IronMan Brain Computer Interface Cassandra Cloud Security Architecture Strategy.

3.3. Cassandra EEG Databases

Cassandra EEG Databases: KeySpaces and Column-Families

The Cassandra NoSQL database(s) utilizes KeySpaces (referenced as databases) for the BCI EEG project the referenced KeySpace, eeg_motor_imagery_openvibe illustrated in Figure 16, utilizes a Simple Strategy which is referred to as "Rack Unware Strategy", the strategy utilizes by default in the org.apache.cassandra.locator.RackUnwareStrategy configuration file. The Simple Strategy places replica sets in a single data center in a topology, which is "not aware" of their placement on a data center rack, thus denoted "Rack Unware Strategy". Theoretically, the usage of Rack Unware Strategy is computationally faster in theory due to implementation, but this is not the case if the next data node has the given keys as opposed to other data nodes.


Cloud Security & Network Infrastructure for CASSANDRA OpenVibe and Emotiv EEG Wireless Database
Software for Brain Computer Interfaces and Real Time Neurosciences
User Wearing Emotiv EPOC Headset
Cassandra
OpenVibe Wireless EEG Data
Emotiv Wireless EEG Data & EPOC Headset
OPEN VIBE CASSANDRA EEG DATABASE
Cassandra
EEG SIGNAL ELECTRODE ARRAY:
TIME
C3
C4
REFNOSE
FC3
FC4
C5
C1
C2
C6
CP3
CP4
HAND_IMAGERY EEG Signal with 317,825 rows imported

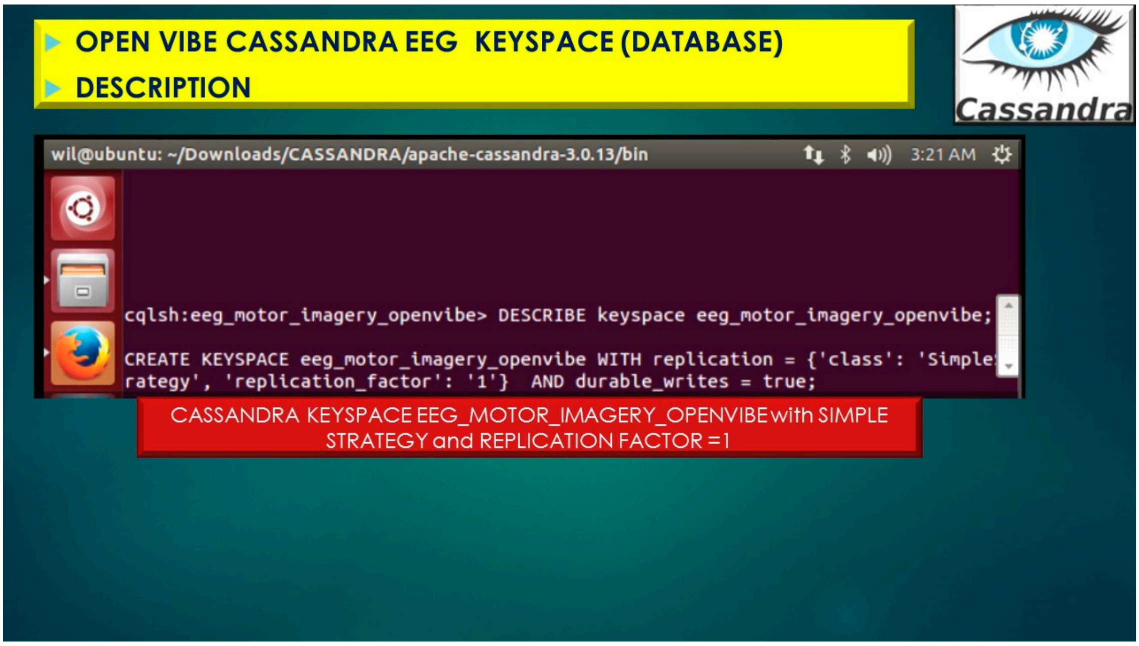
```
wil@ubuntu: ~/Downloads/CASSANDRA/apache-cassandra-3.0.13/bin                2:09 AM
Using 1 child processes

Starting copy of eeg_motor_imagery_openvibe.eeg_1_signal with columns [time, c3,
 c4, refnose, fc3, fc4, c5, c1, c2, c6, cp3, cp4].
Processed: 317825 rows; Rate:    4516 rows/s; Avg. rate:    5412 rows/s
317825 rows imported from 1 files in 58.729 seconds (0 skipped).
cqlsh:eeg_motor_imagery_openvibe> select * from eeg_1_signal;

 time        | c1           | c2           | c3           | c4           | c5
  | c6           | cp3          | cp4          | fc3          | fc4          | refnose
------------+--------------+--------------+--------------+--------------+-----------
--+--------------+--------------+--------------+--------------+--------------+--------
-------
 269.900391 | 6770.871786 | 1693.408449 | 4913.708598 | 8822.398820 | 5564.72271
6 | 5706.354950 | 4636.530085 | 572.144155 | 3202.194656 | 2552.208940 | -44424.
020429
 604.472656 | 6271.239432 | 1697.753532 | 5043.879729 | 8                        5
0 | 5257.041792 | 4729.671281 | 584.728039 | 3299.955093                         .
248252
 128.982422 | 7000.340031 | 1697.824796 | 4830.917829 | 9                        3
7 | 5962.717198 | 4588.889707 | 598.591659 | 3203.050359                         .
786401
 612.814453 | 6249.677684 | 1688.137424 | 5029.629980 | 8                        4
6 | 5262.546375 | 4735.376690 | 599.231512 | 3259.952509                         .
654664
 416.621094 | 6540.118296 | 1701.671215 | 4967.501384 | 8566.845591 | 5621.27440
3 | 5467.650576 | 4673.330208 | 567.238566 | 3197.845043 | 2600.700461 | -43894.
179563
 302.552734 | 6702.402479 | 1686.641554 | 4917.342256 | 8759.689061 | 5579.19953
5 | 5646.661098 | 4633.677258 | 564.465830 | 3197.132018 | 2556.922388 | -44234.
118094
```

Figure 16. Emotiv and OpenVibe EEG Sensor Array stored in Cassandra NoSQL database.

The Replication Factor is specifically important because it indicates how many copies of the portioned data will be stored and then distributed and dispersed throughout the Cassandra Cluster. The Replication Factor Setting yields this information indicated as the following, Simple_Strategy and Replication Factor = 1, and demonstrated in Figures 17 and 18, with displaying primary key and all attributes for keyspace, eeg_motor_imagery_openvibe and column-family (table), eeg_1.

Furthermore, if the replication factor is set to 1, then the writes are written only to a single node, as in Figures 17 and 18, below. If the nodes goes down, the values are no longer accessible. However, if the replication factor is set to 2 or greater, then the nodes in the cluster will get the value written to the nodes on every write and therefore replicas of each other.

The eeg_1 represents the Cassandra Column-Family illustrated in Figures 19 and 20, which is analogous to a table in a traditional relational database system.



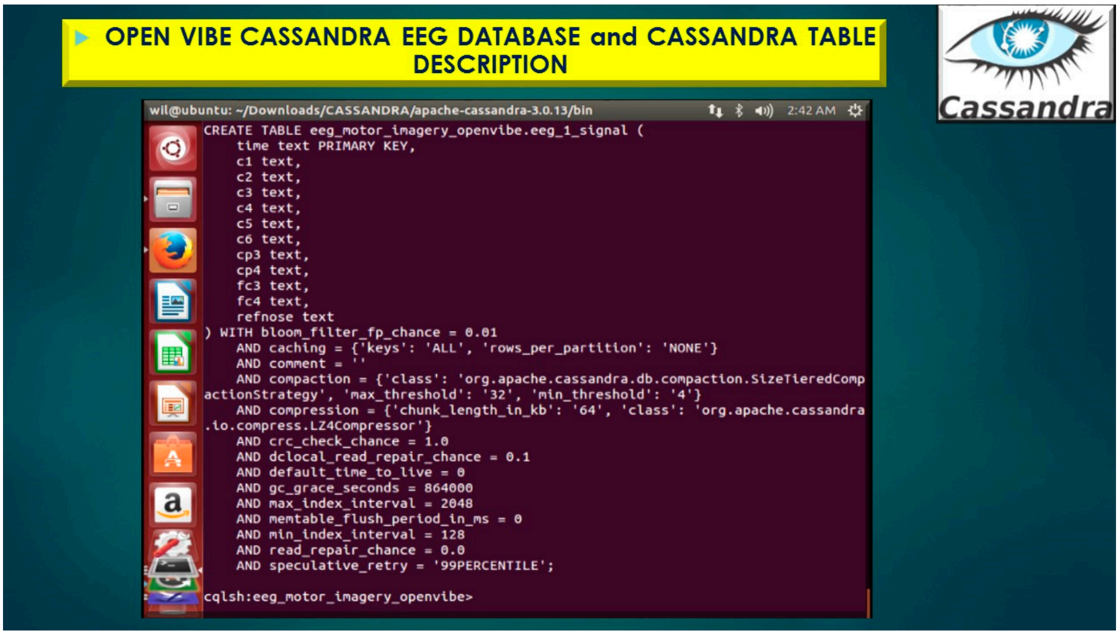
OPEN VIBE CASSANDRA EEG KEYSPACE (DATABASE)

DESCRIPTION

```
wil@ubuntu: ~/Downloads/CASSANDRA/apache-cassandra-3.0.13/bin            3:21 AM

cqlsh:eeg_motor_imagery_openvibe> DESCRIBE keyspace eeg_motor_imagery_openvibe;

CREATE KEYSPACE eeg_motor_imagery_openvibe WITH replication = {'class': 'Simple
rategy', 'replication_factor': '1'}  AND durable_writes = true;
```

CASSANDRA KEYSPACE EEG_MOTOR_IMAGERY_OPENVIBE with SIMPLE STRATEGY and REPLICATION FACTOR =1

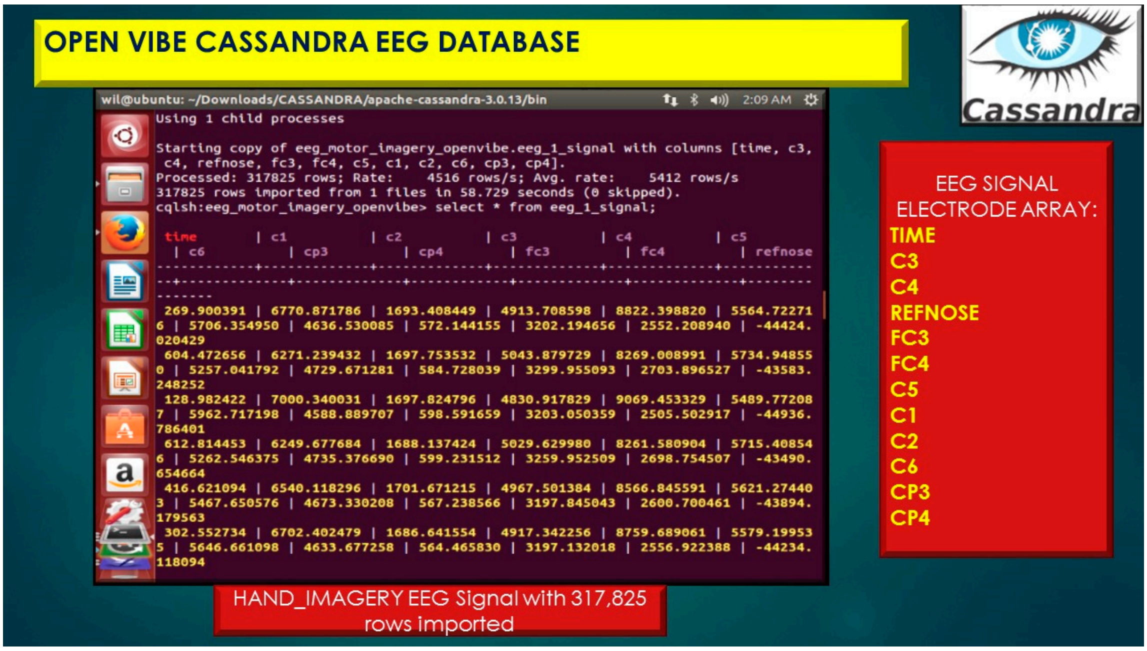
Figure 17. OpenVibe EEG Sensor Array stored in Cassandra NoSQL KEYSPACE (database) with Simple_Strategy and Replication Factor = 1.



OPEN VIBE CASSANDRA EEG DATABASE and CASSANDRA TABLE DESCRIPTION

```
wil@ubuntu: ~/Downloads/CASSANDRA/apache-cassandra-3.0.13/bin            2:42 AM
CREATE TABLE eeg_motor_imagery_openvibe.eeg_1_signal (
    time text PRIMARY KEY,
    c1 text,
    c2 text,
    c3 text,
    c4 text,
    c5 text,
    c6 text,
    cp3 text,
    cp4 text,
    fc3 text,
    fc4 text,
    refnose text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredComp
actionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra
.io.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';

cqlsh:eeg_motor_imagery_openvibe>
```

Figure 18. OpenVibe EEG Sensor Array stored in Cassandra NoSQL KEYSPACE (database) with Simple_Strategy and Replication Factor = 1 displaying primary key and all attributes for keyspace, eeg_motor_imagery_openvibe and table, eeg_1_signal Cassandra statistics.



OPEN VIBE CASSANDRA EEG DATABASE

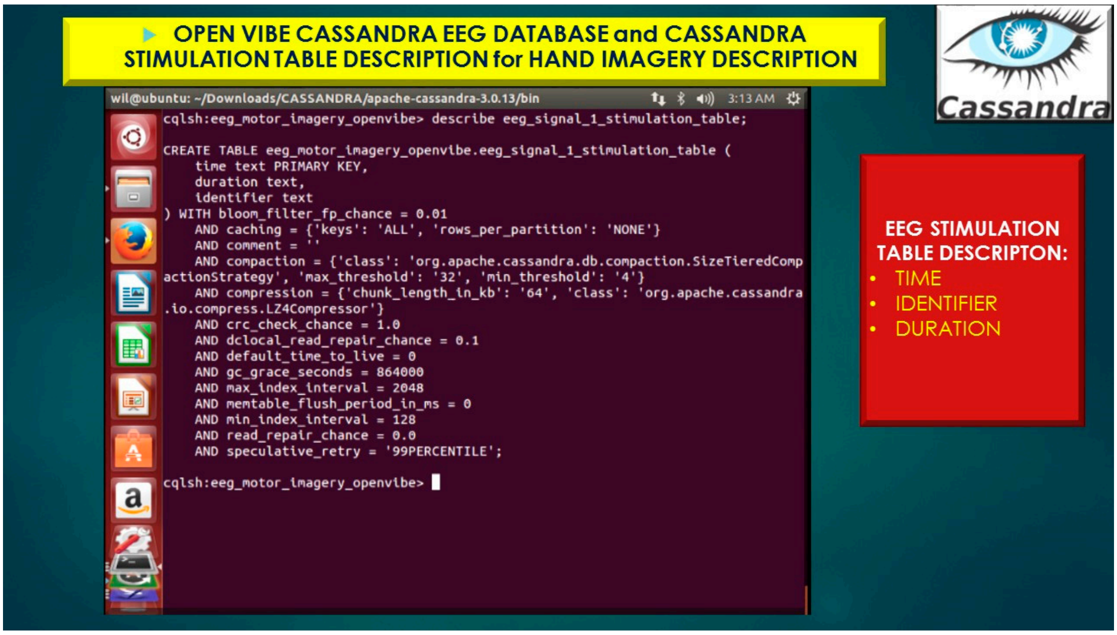
wil@ubuntu: ~/Downloads/CASSANDRA/apache-cassandra-3.0.13/bin
2:09 AM
Cassandra

```
Using 1 child processes

Starting copy of eeg_motor_imagery_openvibe.eeg_1_signal with columns [time, c3,
 c4, refnose, fc3, fc4, c5, c1, c2, c6, cp3, cp4].
Processed: 317825 rows; Rate:    4516 rows/s; Avg. rate:    5412 rows/s
317825 rows imported from 1 files in 58.729 seconds (0 skipped).
cqlsh:eeg_motor_imagery_openvibe> select * from eeg_1_signal;

 time        | c1          | c2          | c3          | c4          | c5
  | c6          | cp3         | cp4         | fc3         | fc4         | refnose
------------+-------------+-------------+-------------+-------------+----------
--+-------------+-------------+------------+-------------+-------------+--------
-------
 269.900391 | 6770.871786 | 1693.408449 | 4913.708598 | 8822.398820 | 5564.72271
6 | 5706.354950 | 4636.530085 | 572.144155 | 3202.194656 | 2552.208940 | -44424.
020429
 604.472656 | 6271.239432 | 1697.753532 | 5043.879729 | 8269.008991 | 5734.94855
0 | 5257.041792 | 4729.671281 | 584.728039 | 3299.955093 | 2703.896527 | -43583.
248252
 128.982422 | 7000.340031 | 1697.824796 | 4830.917829 | 9069.453329 | 5489.77208
7 | 5962.717198 | 4588.889707 | 598.591659 | 3203.050359 | 2505.502917 | -44936.
786401
 612.814453 | 6249.677684 | 1688.137424 | 5029.629980 | 8261.580904 | 5715.40854
6 | 5262.546375 | 4735.376690 | 599.231512 | 3259.952509 | 2698.754507 | -43490.
654664
 416.621094 | 6540.118296 | 1701.671215 | 4967.501384 | 8566.845591 | 5621.27440
3 | 5467.650576 | 4673.330208 | 567.238566 | 3197.845043 | 2600.700461 | -43894.
179563
 302.552734 | 6702.402479 | 1686.641554 | 4917.342256 | 8759.689061 | 5579.19953
5 | 5646.661098 | 4633.677258 | 564.465830 | 3197.132018 | 2556.922388 | -44234.
118094
```

EEG SIGNAL ELECTRODE ARRAY:
TIME
C3
C4
REFNOSE
FC3
FC4
C5
C1
C2
C6
CP3
CP4

HAND_IMAGERY EEG Signal with 317,825 rows imported

Figure 19. OpenVibe EEG Sensor Array stored in Cassandra NoSQL KEYSPACE (database) with Simple_Strategy, table, eeg_1_signal importing 317,825 rows of EEG brain signal data.



OPEN VIBE CASSANDRA EEG DATABASE and CASSANDRA STIMULATION TABLE DESCRIPTION for HAND IMAGERY DESCRIPTION

wil@ubuntu: ~/Downloads/CASSANDRA/apache-cassandra-3.0.13/bin
3:13 AM
Cassandra

```
cqlsh:eeg_motor_imagery_openvibe> describe eeg_signal_1_stimulation_table;

CREATE TABLE eeg_motor_imagery_openvibe.eeg_signal_1_stimulation_table (
    time text PRIMARY KEY,
    duration text,
    identifier text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredComp
actionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra
.io.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';

cqlsh:eeg_motor_imagery_openvibe> 
```

EEG STIMULATION TABLE DESCRIPTON:

- TIME
- IDENTIFIER
- DURATION

Figure 20. OpenVibe EEG Sensor Array stored in Cassandra NoSQL KEYSPACE (database) with Simple_Strategy, Stimulation table, eeg_signal_1_stimulation_table importing eeg brain signal data (*e.g., time, identifier, duration*).

3.4. EEG MongoDB NoSQL Databases

The usage of a NoSQL database such as MongoDB are excellent for fast queries and indexing without the usage of a schema-less architecture and is perfect for sensor inputs, particularly if the sensor fails are does not acquisition the signal properly it will yield a null value which for large number of users could be devasting to a typical relational database management system. The novelty of the MongoDB fast indexing and querying shown in Figures 21–27, below illustrates the power of MongoDB to match the user's intent or Stimulation Code against the acquired brain-signal for fast indexing. As in subsequent sections, the usage of elasticity is a necessity with the MongoDB NoSQL database, since read and write output have linearity as new BCI machines are added to the cloud network, as illustrated in Figure 21 below, discussing beneficial cloud security constraints.

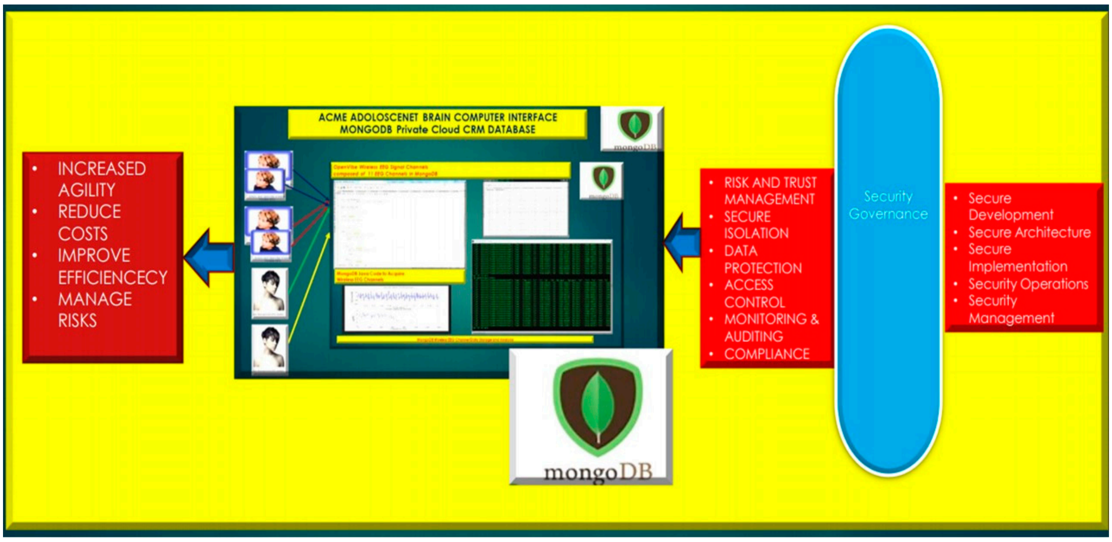
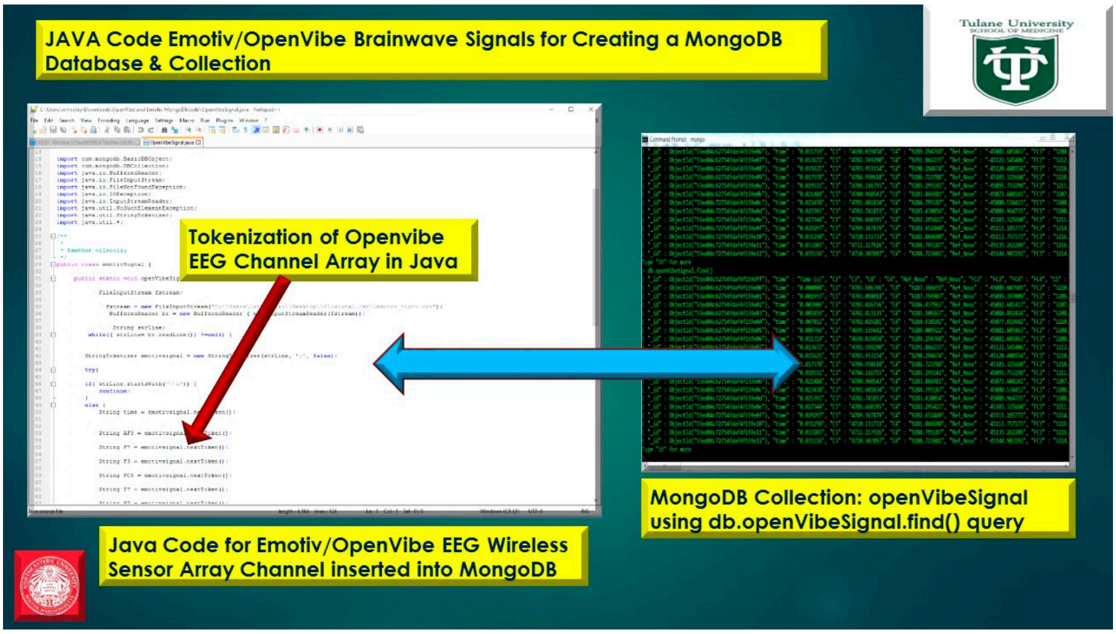
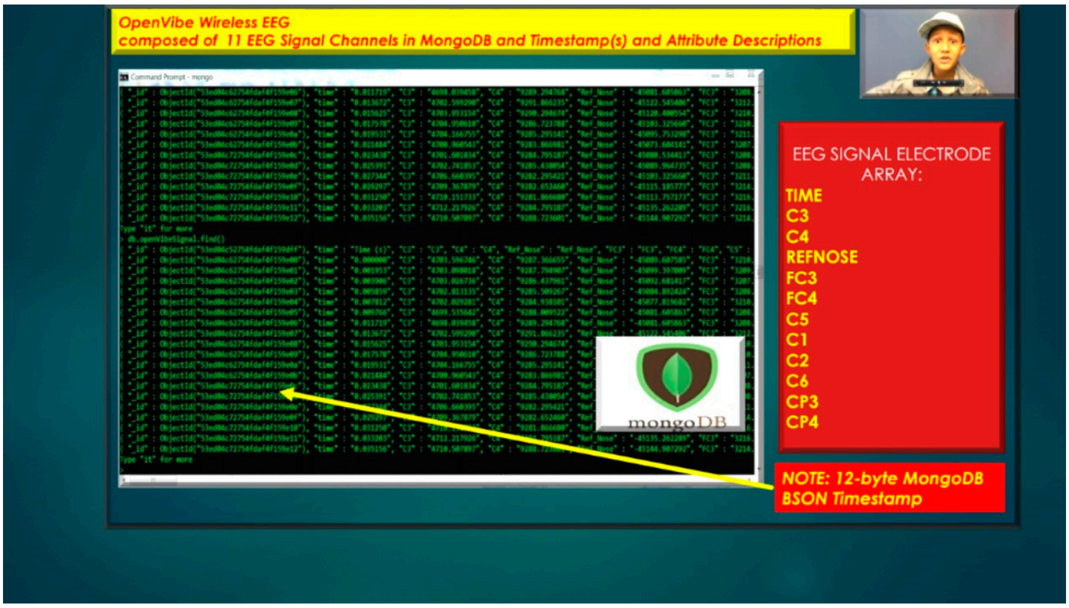


Figure 21. MongoDB Brain Computer Interface Cloud Security Restraints.


JAVA Code Emotiv/OpenVibe Brainwave Signals for Creating a MongoDB Database & Collection
Tulane University
Tokenization of Openvibe EEG Channel Array in Java
MongoDB Collection: openVibeSignal using db.openVibeSignal.find() query
Java Code for Emotiv/OpenVibe EEG Wireless Sensor Array Channel inserted into MongoDB
OpenVibeSignal.java

```
import com.mongodb.BasicDBObject;
import com.mongodb.DBCollection;
import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.NoSuchElementException;
import java.util.StringTokenizer;
import java.util.*;

/**
 *
 * @author wilmcclay
 */
public class emotivSignal {

    public static void openVibeSig[illegible]

            FileInputStream fstream;

             fstream = new FileInputStream("C:\\Users\\w[illegible]ay\\Desktop\\01-signal.csv\\emotiv_right.csv");
              BufferedReader br = new BufferedReader ( [illegible]putStreamReader(fstream));

               String strLine;
          while(( strLine= br.readLine()) !=null) {

        StringTokenizer emotivsignal = new StringT[illegible]zer(strLine, ";", false);

        try{

        if( strLine.startsWith("\\x")) {
            continue;
        }
        else {
            String time = emotivsignal.n[illegible]ken();

            String AF3 = emotivsigna[illegible]Token();
            String F7 = emotivsignal.nextToken();
            String F3 = emotivsignal.nextToken();
            String FC5 = emotivsignal.nextToken();
            String T7 = emotivsignal.nextToken();
            String P7 = emotivsignal.nextToken();
```

Figure 22. Java Tokenization of OpenVibe EEG Sensor Array inputted into MongoDB Collection utilizing db.openVibeSignal.find() queries.



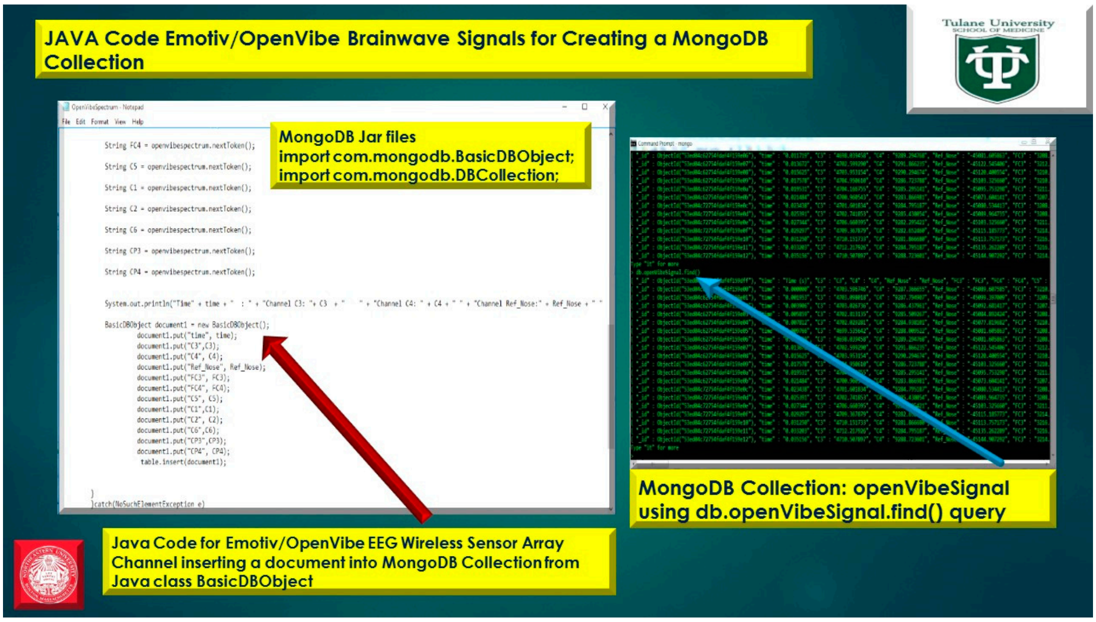
OpenVibe Wireless EEG
composed of 11 EEG Signal Channels in MongoDB and Timestamp(s) and Attribute Descriptions

```
{ "_id" : ObjectId("53ed04c62754fdaf4f159e06"), "time" : "0.011719", "C3" : "4698.039458", "C4" : "9289.294768", "Ref_Nose" : "-45081.605863", "FC3" : "3208.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e07"), "time" : "0.013672", "C3" : "4702.599290", "C4" : "9291.866235", "Ref_Nose" : "-45122.545406", "FC3" : "3212.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e08"), "time" : "0.015625", "C3" : "4703.953154", "C4" : "9290.294674", "Ref_Nose" : "-45120.400554", "FC3" : "3210.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e09"), "time" : "0.017578", "C3" : "4704.950610", "C4" : "9286.723788", "Ref_Nose" : "-45103.325660", "FC3" : "3210.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e0a"), "time" : "0.019531", "C3" : "4704.166755", "C4" : "9285.295141", "Ref_Nose" : "-45095.753298", "FC3" : "3211.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e0b"), "time" : "0.021484", "C3" : "4700.960543", "C4" : "9283.866981", "Ref_Nose" : "-45073.604141", "FC3" : "3207.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e0c"), "time" : "0.023438", "C3" : "4701.601834", "C4" : "9284.795187", "Ref_Nose" : "-45080.534413", "FC3" : "3208.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e0d"), "time" : "0.025391", "C3" : "4702.741853", "C4" : "9285.438054", "Ref_Nose" : "-45089.964735", "FC3" : "3208.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e0e"), "time" : "0.027344", "C3" : "4706.660395", "C4" : "9282.295421", "Ref_Nose" : "-45103.325660", "FC3" : "3211.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e0f"), "time" : "0.029297", "C3" : "4709.367879", "C4" : "9282.652460", "Ref_Nose" : "-45115.185773", "FC3" : "3214.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e10"), "time" : "0.031250", "C3" : "4710.151733", "C4" : "9281.866680", "Ref_Nose" : "-45113.757173", "FC3" : "3216.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e11"), "time" : "0.033203", "C3" : "4712.217926", "C4" : "9284.795187", "Ref_Nose" : "-45135.262289", "FC3" : "3216.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e12"), "time" : "0.035156", "C3" : "4710.507897", "C4" : "9288.723601", "Ref_Nose" : "-45144.907292", "FC3" : "3214.
Type "it" for more
> db.openVibeSignal.find()
{ "_id" : ObjectId("53ed04c52754fdaf4f159dff"), "time" : "Time (s)", "C3" : "C3", "C4" : "C4", "Ref_Nose" : "Ref_Nose", "FC3" : "FC3", "FC4" : "FC4", "C5" :
{ "_id" : ObjectId("53ed04c62754fdaf4f159e00"), "time" : "0.000000", "C3" : "4703.596746", "C4" : "9287.366655", "Ref_Nose" : "-45089.607585", "FC3" : "3210.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e01"), "time" : "0.001953", "C3" : "4703.098018", "C4" : "9287.794907", "Ref_Nose" : "-45099.397009", "FC3" : "3209.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e02"), "time" : "0.003906", "C3" : "4703.026736", "C4" : "9286.437961", "Ref_Nose" : "-45092.681417", "FC3" : "3207.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e03"), "time" : "0.005859", "C3" : "4702.813135", "C4" : "9285.509267", "Ref_Nose" : "-45084.892424", "FC3" : "3208.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e04"), "time" : "0.007812", "C3" : "4702.029281", "C4" : "9284.938101", "Ref_Nose" : "-45077.819682", "FC3" : "3210.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e05"), "time" : "0.009766", "C3" : "4699.535642", "C4" : "9288.009522", "Ref_Nose" : "-45081.605863", "FC3" : "3208.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e06"), "time" : "0.011719", "C3" : "4698.039458", "C4" : "9289.294768", "Ref_Nose" : "-45081.605863", "FC3" : "3208.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e07"), "time" : "0.013672", "C3" : "4702.599290", "C4" : "9291.866235", "Ref_Nose" : "-45122.545406", "FC3" : "3212.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e08"), "time" : "0.015625", "C3" : "4703.953154", "C4" : "9290.294674", "Ref
{ "_id" : ObjectId("53ed04c62754fdaf4f159e09"), "time" : "0.017578", "C3" : "4704.950610", "C4" : "9286.723788", "Ref
{ "_id" : ObjectId("53ed04c62754fdaf4f159e0a"), "time" : "0.019531", "C3" : "4704.166755", "C4" : "9285.295141", "Ref
{ "_id" : ObjectId("53ed04c62754fdaf4f159e0b"), "time" : "0.021484", "C3" : "4700.960543", "C4" : "9283.866981", "Ref
{ "_id" : ObjectId("53ed04c72754fdaf4f159e0c"), "time" : "0.023438", "C3" : "4701.601834", "C4" : "9284.795187", "Ref
{ "_id" : ObjectId("53ed04c72754fdaf4f15  "time" : "0.025391", "C3" : "4702.741853", "C4" : "9285.438054", "Ref
{ "_id" : ObjectId("53ed04c72754fdaf4f159e0e"), "time" : "0.027344", "C3" : "4706.660395", "C4" : "9282.295421", "Ref
{ "_id" : ObjectId("53ed04c72754fdaf4f159e0f"), "time" : "0.029297", "C3" : "4709.367879", "C4" : "9282.652460", "Ref
{ "_id" : ObjectId("53ed04c72754fdaf4f159e10"), "time" : "0.031250", "C3" : "4710.151733", "C4" : "9281.866680", "Ref
{ "_id" : ObjectId("53ed04c72754fdaf4f159e11"), "time" : "0.033203", "C3" : "4712.217926", "C4" : "9284.795187", "Ref_Nose" : "-45135.262289", "FC3" : "3216.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e12"), "time" : "0.035156", "C3" : "4710.507897", "C4" : "9288.723601", "Ref_Nose" : "-45144.907292", "FC3" : "3214.
Type "it" for more
>
```

EEG SIGNAL ELECTRODE ARRAY:
TIME
C3
C4
REFNOSE
FC3
FC4
C5
C1
C2
C6
CP3
CP4

NOTE: 12-byte MongoDB BSON Timestamp

Figure 23. Usage of NoSQL database MongoDB for Wireless EEG Signal Storage and Retrieval with MongoDB BSON Timestamp with EEG Signal Electrode Array.



JAVA Code Emotiv/OpenVibe Brainwave Signals for Creating a MongoDB Collection

```
        String FC4 = openvibespectrum.nextToken();

        String C5 = openvibespectrum.nextToken();

        String C1 = openvibespectrum.nextToken();

        String C2 = openvibespectrum.nextToken();

        String C6 = openvibespectrum.nextToken();

        String CP3 = openvibespectrum.nextToken();

        String CP4 = openvibespectrum.nextToken();


        System.out.println("Time" + time + "  : " + "Channel C3: "+ C3  + "    " + "Channel C4: " + C4 + " " + "Channel Ref_Nose:" + Ref_Nose + " "

        BasicDBObject document1 = new BasicDBObject();
                document1.put("time", time);
                document1.put("C3",C3);
                document1.put("C4", C4);
                document1.put("Ref_Nose", Ref_Nose);
                document1.put("FC3", FC3);
                document1.put("FC4", FC4);
                document1.put("C5", C5);
                document1.put("C1",C1);
                document1.put("C2", C2);
                document1.put("C6",C6);
                document1.put("CP3",CP3);
                document1.put("CP4", CP4);
                 table.insert(document1);


    }
    }catch(NoSuchElementException e)
```

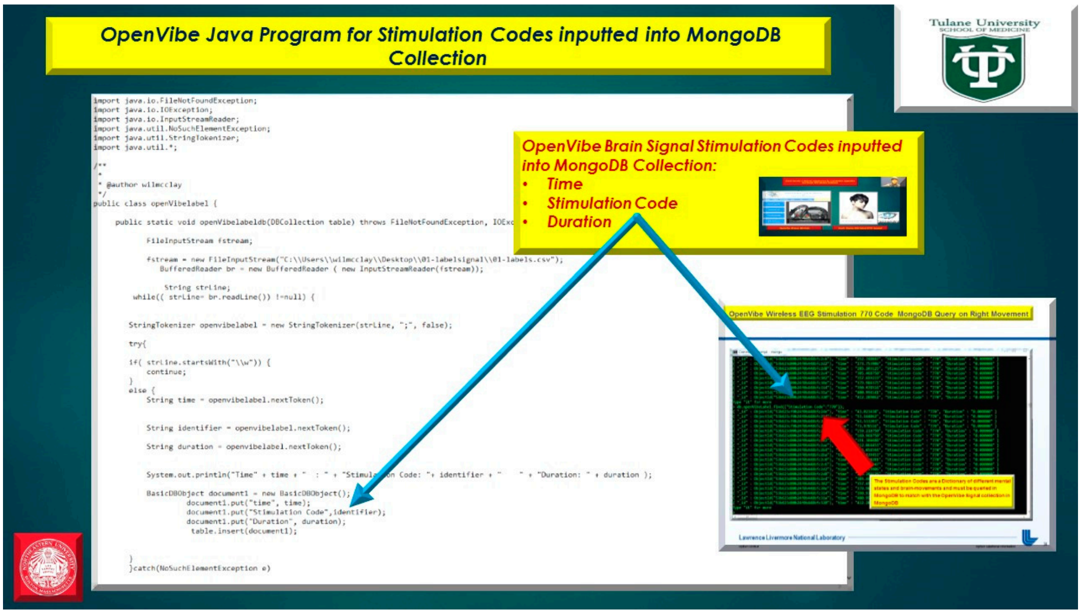
MongoDB Jar files
import com.mongodb.BasicDBObject;
import com.mongodb.DBCollection;

Java Code for Emotiv/OpenVibe EEG Wireless Sensor Array Channel inserting a document into MongoDB Collection from Java class BasicDBObject

```
{ "_id" : ObjectId("53ed04c62754fdaf4f159e06"), "time" : "0.011719", "C3" : "4698.039458", "C4" : "9289.294768", "Ref_Nose" : "-45081.605863", "FC3" : "3208.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e07"), "time" : "0.013672", "C3" : "4702.599290", "C4" : "9291.866235", "Ref_Nose" : "-45122.545406", "FC3" : "3212.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e08"), "time" : "0.015625", "C3" : "4703.953154", "C4" : "9290.294674", "Ref_Nose" : "-45120.400554", "FC3" : "3210.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e09"), "time" : "0.017578", "C3" : "4704.950610", "C4" : "9286.723788", "Ref_Nose" : "-45103.325660", "FC3" : "3210.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e0a"), "time" : "0.019531", "C3" : "4704.166755", "C4" : "9285.295141", "Ref_Nose" : "-45095.753298", "FC3" : "3211.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e0b"), "time" : "0.021484", "C3" : "4700.960543", "C4" : "9283.866981", "Ref_Nose" : "-45073.604141", "FC3" : "3207.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e0c"), "time" : "0.023438", "C3" : "4701.601834", "C4" : "9284.795187", "Ref_Nose" : "-45080.534413", "FC3" : "3208.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e0d"), "time" : "0.025391", "C3" : "4702.741853", "C4" : "9285.438054", "Ref_Nose" : "-45089.964735", "FC3" : "3208.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e0e"), "time" : "0.027344", "C3" : "4706.660395", "C4" : "9282.295421", "Ref_Nose" : "-45103.325660", "FC3" : "3211.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e0f"), "time" : "0.029297", "C3" : "4709.367879", "C4" : "9282.652460", "Ref_Nose" : "-45115.185773", "FC3" : "3214.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e10"), "time" : "0.031250", "C3" : "4710.151733", "C4" : "9281.866680", "Ref_Nose" : "-45113.757173", "FC3" : "3216.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e11"), "time" : "0.033203", "C3" : "4712.217926", "C4" : "9284.795187", "Ref_Nose" : "-45135.262289", "FC3" : "3216.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e12"), "time" : "0.035156", "C3" : "4710.507897", "C4" : "9288.723601", "Ref_Nose" : "-45144.907292", "FC3" : "3214.
Type "it" for more
> db.openVibeSignal.find()
{ "_id" : ObjectId("53ed04c52754fdaf4f159dff"), "time" : "Time (s)", "C3" : "C3", "C4" : "C4", "Ref_Nose" : "Ref_Nose", "FC3" : "FC3", "FC4" : "FC4", "C5" :
{ "_id" : ObjectId("53ed04c62754fdaf4f159e00"), "time" : "0.000000", "C3" : "4703.596746", "C4" : "9287.366655", "Ref_Nose" : "-45089.607585", "FC3" : "3210.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e01"), "time" : "0.001953", "C3" : "4703.098018", "C4" : "9287.794907", "Ref_Nose" : "-45099.397009", "FC3" : "3209.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e02"), "time" : "0.003906", "C3" : "4703.026736", "C4" : "9286.437961", "Ref_Nose" : "-45092.681417", "FC3" : "3207.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e03"), "time" : "0.005859", "C3" : "4702.813135", "C4" : "9285.509267", "Ref_Nose" : "-45084.892424", "FC3" : "3208.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e04"), "time" : "0.007812", "C3" : "4702.029281", "C4" : "9284.938101", "Ref_Nose" : "-45077.819682", "FC3" : "3210.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e05"), "time" : "0.009766", "C3" : "4699.535642", "C4" : "9288.009522", "Ref_Nose" : "-45081.605863", "FC3" : "3208.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e06"), "time" : "0.011719", "C3" : "4698.039458", "C4" : "9289.294768", "Ref_Nose" : "-45081.605863", "FC3" : "3208.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e07"), "time" : "0.013672", "C3" : "4702.599290", "C4" : "9291.866235", "Ref_Nose" : "-45122.545406", "FC3" : "3212.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e08"), "time" : "0.015625", "C3" : "4703.953154", "C4" : "9290.294674", "Ref_Nose" : "-45120.400554", "FC3" : "3210.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e09"), "time" : "0.017578", "C3" : "4704.950610", "C4" : "9286.723788", "Ref_Nose" : "-45103.325660", "FC3" : "3210.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e0a"), "time" : "0.019531", "C3" : "4704.166755", "C4" : "9285.295141", "Ref_Nose" : "-45095.753298", "FC3" : "3211.
{ "_id" : ObjectId("53ed04c62754fdaf4f159e0b"), "time" : "0.021484", "C3" : "4700.960543", "C4" : "9283.866981", "Ref_Nose" : "-45073.604141", "FC3" : "3207.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e0c"), "time" : "0.023438", "C3" : "4701.601834", "C4" : "9284.795187", "Ref_Nose" : "-45080.534413", "FC3" : "3208.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e0d"), "time" : "0.025391", "C3" : "4702.741853", "C4" : "9285.438054", "Ref_Nose" : "-45089.964735", "FC3" : "3208.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e0e"), "time" : "0.027344", "C3" : "4706.660395", "C4" : "9282.295421", "Ref_Nose" : "-45103.325660", "FC3" : "3211.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e0f"), "time" : "0.029297", "C3" : "4709.367879", "C4" : "9282.652460", "Ref_Nose" : "-45115.185773", "FC3" : "3214.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e10"), "time" : "0.031250", "C3" : "4710.151733", "C4" : "9281.866680", "Ref_Nose" : "-45113.757173", "FC3" : "3216.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e11"), "time" : "0.033203", "C3" : "4712.217926", "C4" : "9284.795187", "Ref_Nose" : "-45135.262289", "FC3" : "3216.
{ "_id" : ObjectId("53ed04c72754fdaf4f159e12"), "time" : "0.035156", "C3" : "4710.507897", "C4" : "9288.723601", "Ref_Nose" : "-45144.907292", "FC3" : "3214.
Type "it" for more
```

MongoDB Collection: openVibeSignal using db.openVibeSignal.find() query

Figure 24. Java Program for Emotiv and OpenVibe EEG Sensor Array Channel inserting a document into MongoDB Collection using Java class ***BasicDBObject***.



OpenVibe Java Program for Stimulation Codes inputted into MongoDB Collection

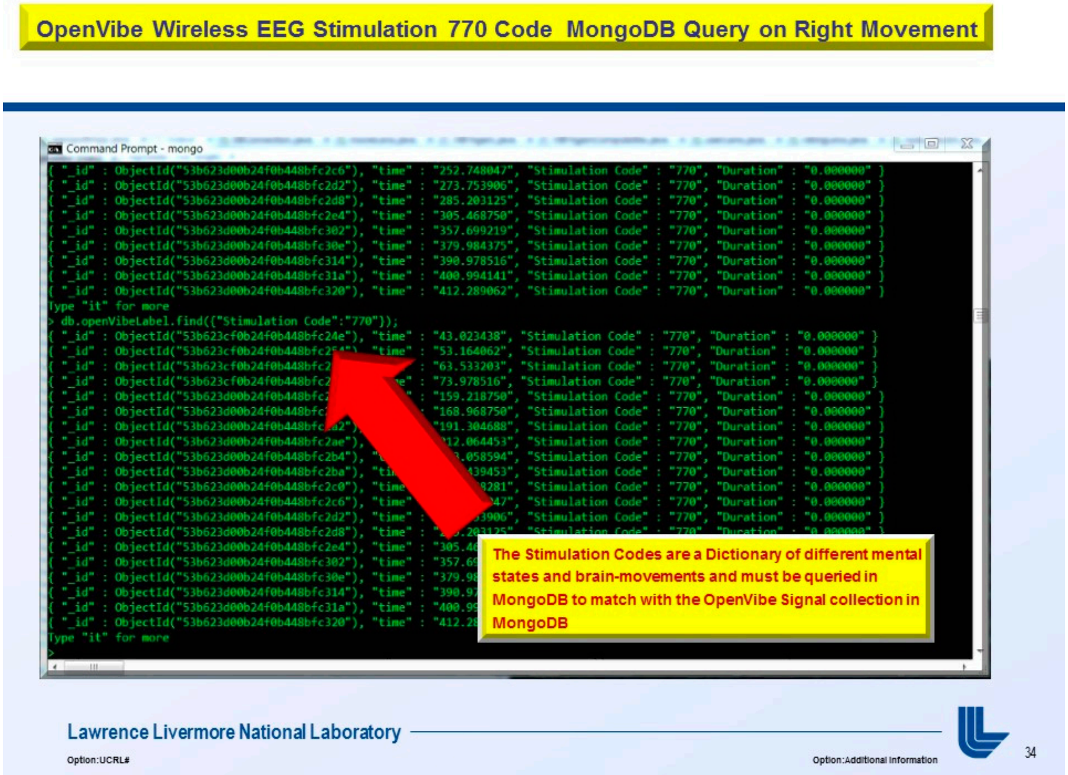
```
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.NoSuchElementException;
import java.util.StringTokenizer;
import java.util.*;

/**
 *
 * @author wilmcclay
 */
public class openVibelabel {

    public static void openVibelabeldb(DBCollection table) throws FileNotFoundException, IOExc

          FileInputStream fstream;

          fstream = new FileInputStream("C:\\Users\\wilmcclay\\Desktop\\01-labelsignal\\01-labels.csv");
             BufferedReader br = new BufferedReader ( new InputStreamReader(fstream));

              String strLine;
      while(( strLine= br.readLine()) !=null) {


      StringTokenizer openvibelabel = new StringTokenizer(strLine, ";", false);

      try{

      if( strLine.startsWith("\\w")) {
          continue;
      }
      else {
          String time = openvibelabel.nextToken();


          String identifier = openvibelabel.nextToken();

          String duration = openvibelabel.nextToken();


          System.out.println("Time" + time + "  : " + "Stimula   on Code: "+ identifier + "    " + "Duration: " + duration );

          BasicDBObject document1 = new BasicDBObject();
                  document1.put("time", time);
                  document1.put("Stimulation Code",identifier);
                  document1.put("Duration", duration);
                   table.insert(document1);


      }
      }catch(NoSuchElementException e)
```

OpenVibe Brain Signal Stimulation Codes inputted into MongoDB Collection:

- Time
- Stimulation Code
- Duration

OpenVibe Wireless EEG Stimulation 770 Code MongoDB Query on Right Movement
Tulane University
SCHOOL OF MEDICINE
Lawrence Livermore National Laboratory

Figure 25. OpenVibe EEG Sensor Array Java Program for Brainwave Signal Stimulation Codes for time, stimulation code, and duration.



OpenVibe Wireless EEG Stimulation 770 Code MongoDB Query on Right Movement

```
{ "_id" : ObjectId("53b623d00b24f0b448bfc2c6"), "time" : "252.748047", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc2d2"), "time" : "273.753906", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc2d8"), "time" : "285.203125", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc2e4"), "time" : "305.468750", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc302"), "time" : "357.699219", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc30e"), "time" : "379.984375", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc314"), "time" : "390.978516", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc31a"), "time" : "400.994141", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc320"), "time" : "412.289062", "Stimulation Code" : "770", "Duration" : "0.000000" }
Type "it" for more
> db.openVibeLabel.find({"Stimulation Code":"770"});
{ "_id" : ObjectId("53b623cf0b24f0b448bfc24e"), "time" : "43.023438", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623cf0b24f0b448bfc254"), "time" : "53.164062", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623cf0b24f0b448bfc2            : "63.533203", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623cf0b24f0b448bfc2        e" : "73.978516", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc2            "159.218750", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc             "168.968750", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc a2")        "191.304688", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc2ae"),        12.064453", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc2b4"), "       .058594", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc2ba"), "ti      439453", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc2c0"), "time     8281", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc2c6"), "time"     47", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc2d2"), "time" :   3906", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc2d8"), "time" :   .203125", "Stimulation Code" : "770", "Duration" : "0.000000" }
{ "_id" : ObjectId("53b623d00b24f0b448bfc2e4"), "time" : "305.46
{ "_id" : ObjectId("53b623d00b24f0b448bfc302"), "time" : "357.69
{ "_id" : ObjectId("53b623d00b24f0b448bfc30e"), "time" : "379.98
{ "_id" : ObjectId("53b623d00b24f0b448bfc314"), "time" : "390.97
{ "_id" : ObjectId("53b623d00b24f0b448bfc31a"), "time" : "400.99
{ "_id" : ObjectId("53b623d00b24f0b448bfc320"), "time" : "412.28
Type "it" for more
>
```

The Stimulation Codes are a Dictionary of different mental states and brain-movements and must be queried in MongoDB to match with the OpenVibe Signal collection in MongoDB

Command Prompt - mongo
Lawrence Livermore National Laboratory
Option:UCRL#
Option:Additional Information
34

Figure 26. Wireless EEG Java Stimulation Code Dictionary to input EEG signal patterns in MongoDB.

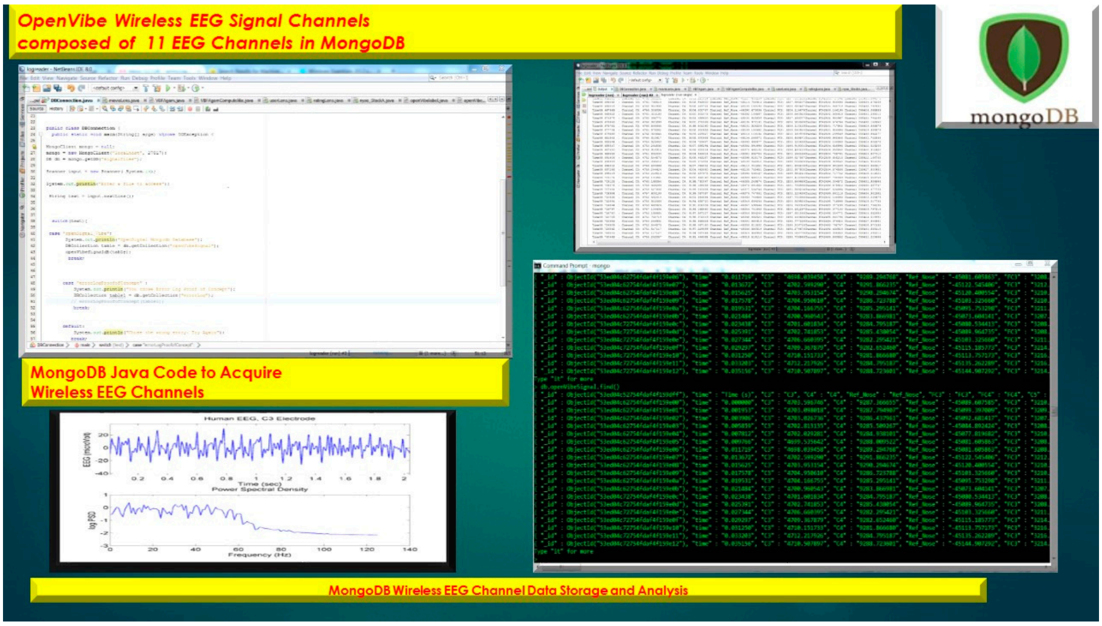
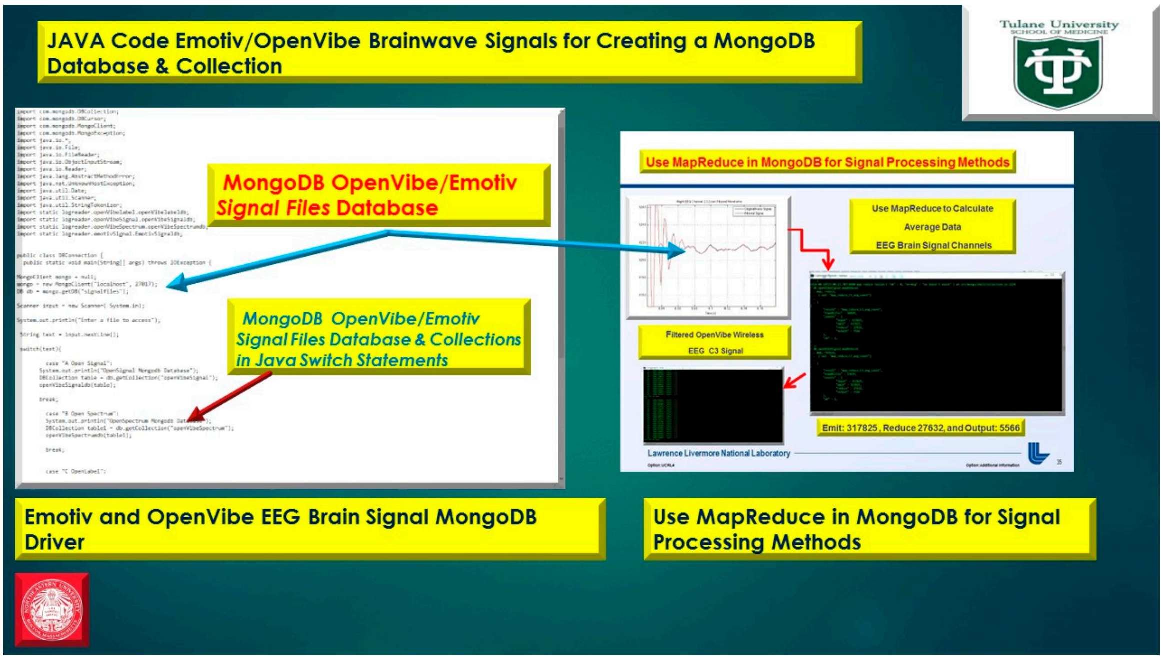


Figure 27. Stimulation Codes have to match the acquired EEG signal patterns in MongoDB.

The other asset to utilizing MongoDB as a document-oriented data store is the use of MapReduce as a computational paradigm for key-value pairs to do basic signal processing methods for each user's intent in a condensed and formulated manner, as shown below in Figures 23–28. MongoDB also utilizes an ObjectID with a 12-byte timestamp in BSON notation composed of the machine id, process, and other features. The usage of ObjectID to identify a document in a MongoDB collection (e.g., table) is useful for MEG/EEG brainwave signal channel arrays, which can be easily parsed in a Java MongoDB driver file to Tokenize the channel array before ingestion into the MongoDB, "Signal Files" database, as demonstrated in Figure 23 [20,21], below. For instance, in Figure 23, a MongoDB connection was implemented with a series of Java Case Switch Statements illustrating either an Emotiv or OpenVibe collection for ingestion into the MongoDB Signal Files database, also illustrated in Figure 27. In addition in Figure 23, once the OpenVibe EEG sensor channel is Tokenized and ingested in MongoDB, the usage of signal processing techniques on EEG channels can be implemented in MongoDB also utilizing the MapReduce computational paradigm algorithm for key-value pair analysis with signal processing techniques illustrated in Figure 28, below. Thus, this is beneficial for sensor-based signal acquisition, analysis, and fault-tolerance. For instance, if one of the MEG or EEG sensor channels fails to emit brain signal activity, the database will not yield large sequences of "NULL" values that could easily compromise or crash a traditional relational database during signal acquisition and real-time queries. Therefore, the utilization of NoSQL databases such as MongoDB are the quintessential tool for real-time signal acquisition and analysis for data storage [2], and additionally with the MongoDB BSON timestamp notation a replica data set on a node can be easily referenced and detected if a system failure took place during the Brain Computer Interface brainwave data acquisition.



JAVA Code Emotiv/OpenVibe Brainwave Signals for Creating a MongoDB Database & Collection

```
import com.mongodb.DBCollection;
import com.mongodb.DBCursor;
import com.mongodb.MongoClient;
import com.mongodb.MongoException;
import java.io.*;
import java.io.File;
import java.io.FileReader;
import java.io.ObjectInputStream;
import java.io.Reader;
import java.lang.AbstractMethodError;
import java.net.UnknownHostException;
import java.util.Date;
import java.util.Scanner;
import java.util.StringTokenizer;
import static logreader.openVibelabel.openVibelabeldb;
import static logreader.openVibeSignal.openVibeSignaldb;
import static logreader.openVibeSpectrum.openVibeSpectrumdb;
import static logreader.emotivSignal.EmotivSignaldb;

public class DBConnection {
  public static void main(String[] args) throws IOException {

MongoClient mongo = null;
mongo = new MongoClient("localhost", 27017);
DB db = mongo.getDB("signalfiles");

Scanner input = new Scanner( System.in);

System.out.println("Enter a file to access");

 String text = input.nextLine();

 switch(text){

        case "A Open Signal":
      System.out.println("OpenSignal Mongodb Database");
      DBCollection table = db.getCollection("openVibeSignal");
      openVibeSignaldb(table);

      break;

        case "B Open Spectrum":
        System.out.println("OpenSpectrum Mongodb Database");
        DBCollection table1 = db.getCollection("openVibeSpectrum");
        openVibeSpectrumdb(table1);

        break;


        case "C OpenLabel":
```

MongoDB OpenVibe/Emotiv *Signal Files* Database

MongoDB OpenVibe/Emotiv Signal Files Database & Collections in Java Switch Statements

Use MapReduce in MongoDB for Signal Processing Methods

Use MapReduce to Calculate Average Data EEG Brain Signal Channels

Filtered OpenVibe Wireless EEG C3 Signal

Emit: 317825 , Reduce 27632, and Output: 5566

Lawrence Livermore National Laboratory

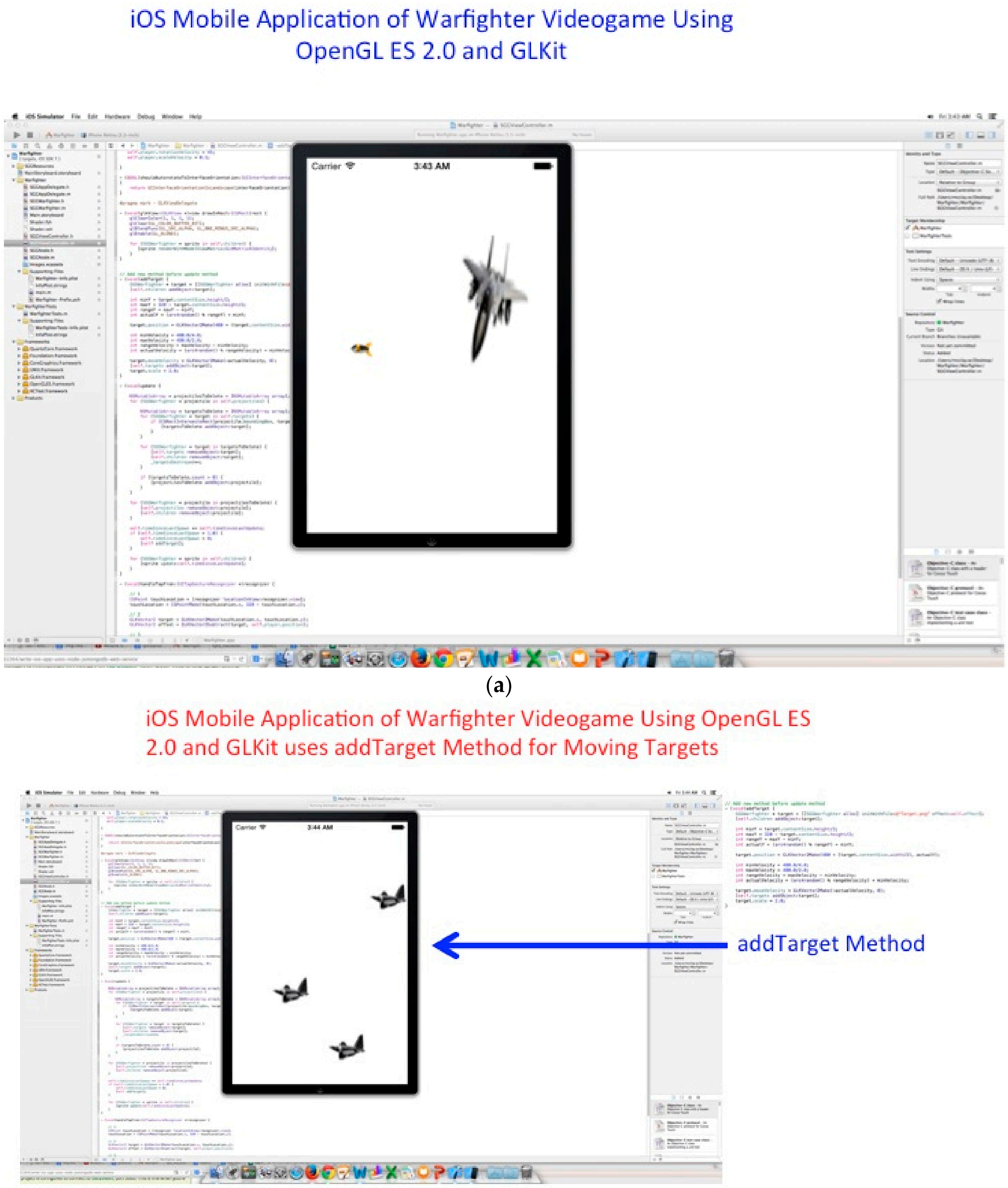
Emotiv and OpenVibe EEG Brain Signal MongoDB Driver

Use MapReduce in MongoDB for Signal Processing Methods

Figure 28. MapReduce in MongoDB for Signal Processing and EEG data analytics.

3.5. EEG and MEG BCI Objective and iPhone Integration

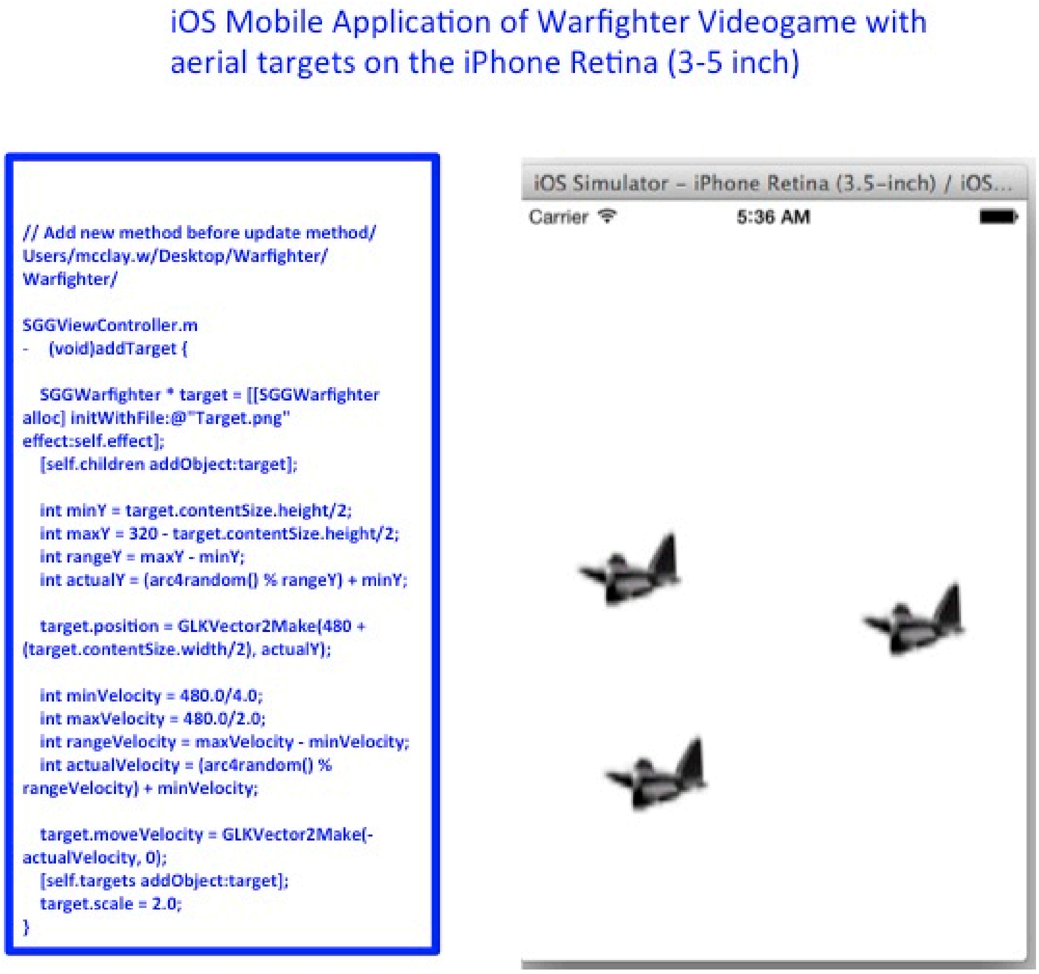
The Wireless EEG project describes the measurements done with a wireless EEG neuro-helmet while a user is involved in a warfighter simulation while using brainwaves that translate the user's intentions into actions controlling the warfighter simulator or manually using the iOS UITapGestureRecognizer Class in the OpenGL ES 2.0 and GLKit environment to fire projectiles or control movement which can be done by button-press or from acquired offline Emotiv and OpenVibe EEG brain-wave files while using the following mongodb export function, for example "mongoexport –db brainwaveusers –collection brainsignals –csv –fieldFile fields.txt –out /opt/backups/contacts.csv", as shown in Figures 29, 30 and 31a below.


iOS Mobile Application of Warfighter Videogame Using OpenGL ES 2.0 and GLKit
Carrier
3:43 AM
iOS Mobile Application of Warfighter Videogame Using OpenGL ES 2.0 and GLKit uses addTarget Method for Moving Targets
Carrier
3:44 AM
addTarget Method

(**a**)

(**b**)

Figure 29. *Cont.*



iOS Mobile Application of Warfighter Videogame with aerial targets on the iPhone Retina (3-5 inch)

```
// Add new method before update method/
Users/mcclay.w/Desktop/Warfighter/
Warfighter/

SGGViewController.m
-   (void)addTarget {

   SGGWarfighter * target = [[SGGWarfighter
alloc] initWithFile:@"Target.png"
effect:self.effect];
   [self.children addObject:target];

   int minY = target.contentSize.height/2;
   int maxY = 320 - target.contentSize.height/2;
   int rangeY = maxY - minY;
   int actualY = (arc4random() % rangeY) + minY;

   target.position = GLKVector2Make(480 +
(target.contentSize.width/2), actualY);

   int minVelocity = 480.0/4.0;
   int maxVelocity = 480.0/2.0;
   int rangeVelocity = maxVelocity - minVelocity;
   int actualVelocity = (arc4random() %
rangeVelocity) + minVelocity;

   target.moveVelocity = GLKVector2Make(-
actualVelocity, 0);
   [self.targets addObject:target];
   target.scale = 2.0;
}
```

(**c**)

Figure 29. (**a**) iOS Mobile Application of Warfighter Videogame using OpenGL ES 2.0 (Khronos Group, Beaverton, Oregon if USA, country, https://www.khronos.org/about/) and GLKit with the UITapGestureRecognizer class to fire a projectile. (**b**) iOS Mobile Application of Warfighter Videogame using OpenGL ES 2.0 and GLKit with aerial targets using the addTarget Method. (**c**) Display of iOS Mobile Application of Warfighter Videogame using OpenGL ES 2.0 and GLKit with aerial targets using the addTarget Method (close-up).

We developed a driver that is able to substitute the actions of the BCI as mouse button presses for real-time or non-real-time use in visual simulations. The process was added into warfighter simulator visualization demonstration. By thinking left or right, the user experiences the aircraft turning in the chosen direction. The driver components of the BCI can be compiled into any software and substitute a user's intent for specific keyboard strikes or mouse button presses to evade or chase aerial targets, as shown in Figure 29 referenced above and Figure 30 below.

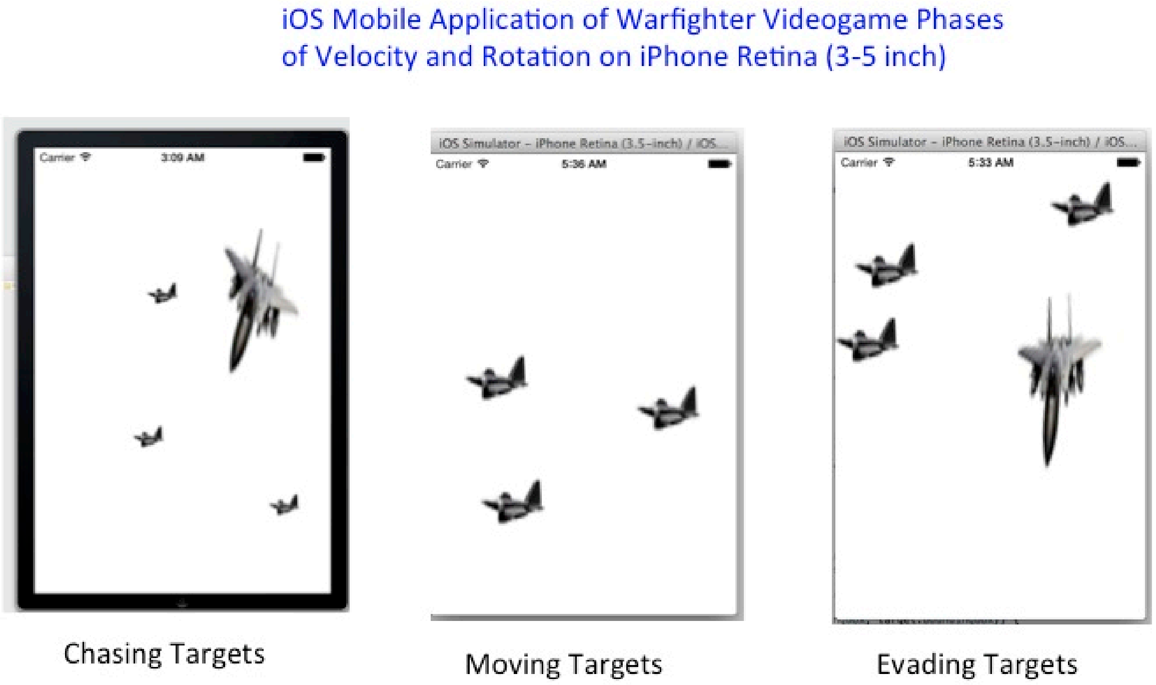
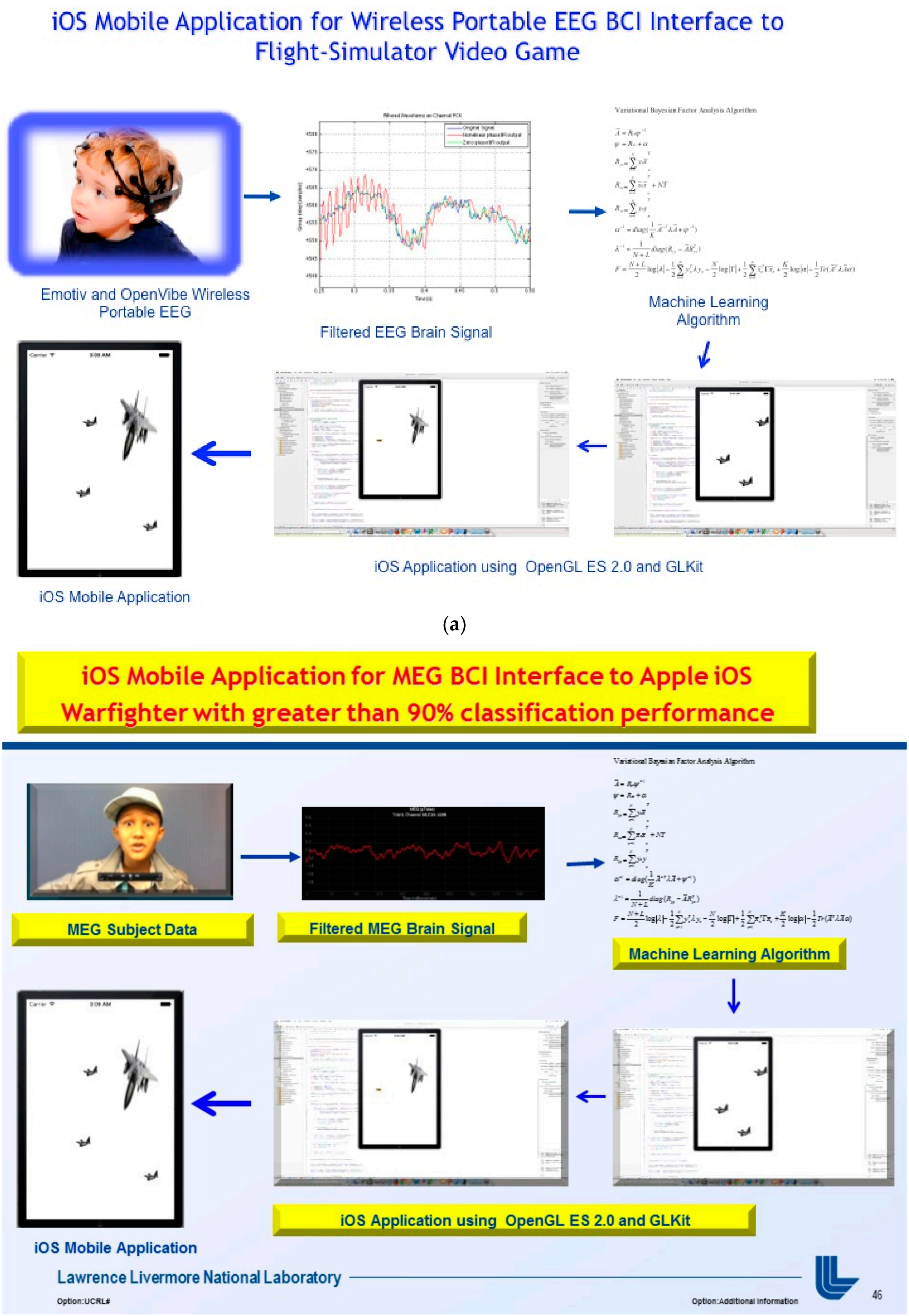

iOS Mobile Application of Warfighter Videogame Phases
of Velocity and Rotation on iPhone Retina (3-5 inch)
Carrier
3:09 AM
iOS Simulator – iPhone Retina (3.5-inch) / iOS...
Carrier
5:36 AM
iOS Simulator – iPhone Retina (3.5-inch) / iOS...
Carrier
5:33 AM
Chasing Targets
Moving Targets
Evading Targets

Figure 30. iOS Mobile Application of Warfighter Videogame using OpenGL ES 2.0 and GLKit to evade or chase aerial targets.

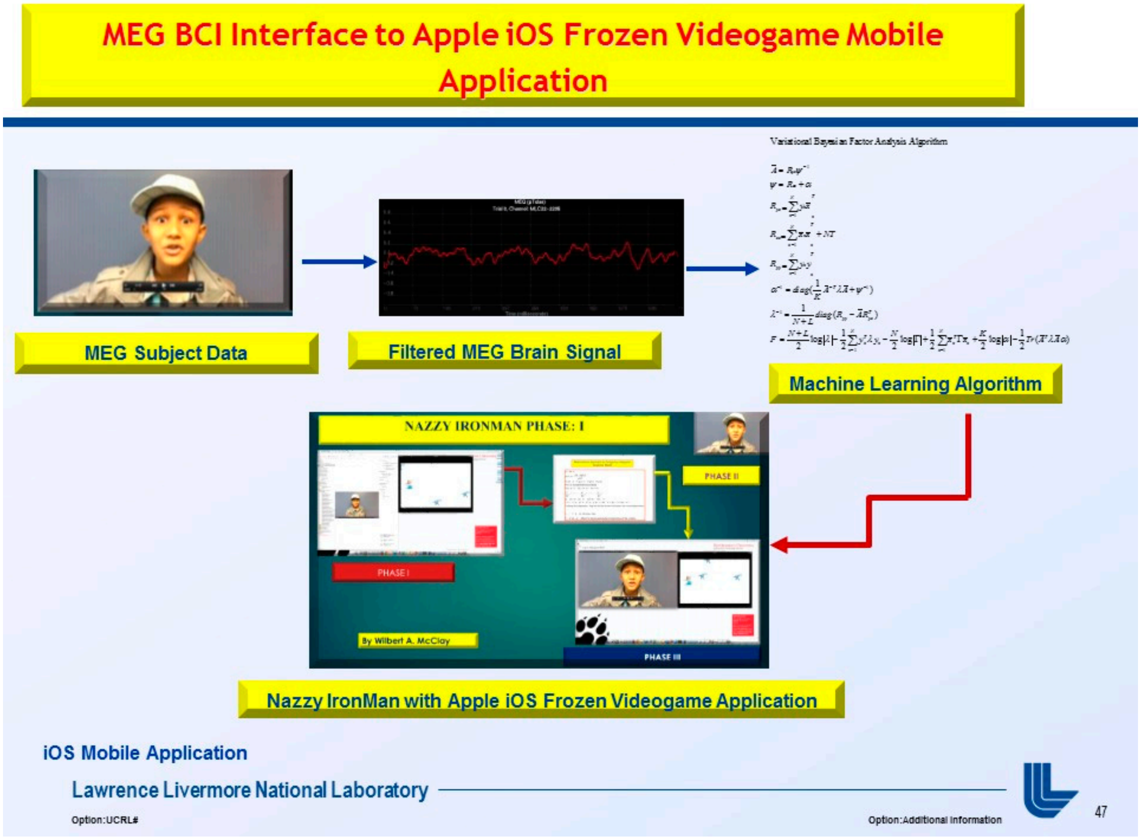
The innovation of the Wireless EEG BCI builds upon the BCI technology developed at LLNL, for user intent, which induces certain actions and has myriads of applications. The Wireless EEG BCI allows for people to use their brainwaves and psychological state as control possibilities. In Figure 30 we illustrate this below using the BCI with moving aerial targets in the flow diagram. However, the VBFA machine algorithm performance on EEG data classification is drastically below performance (e.g. approximately less than 60% EEG Subject performance) when compared to greater than 90% performance from MEG Subject data, as illustrated in Figure 31b. This is primarily due to better signal acquisition from the UCSF CTF MEG Scanner with 275 sensor array utilizing Superconducting Quantum Interference Device (SQUIDs) technology in a magnetically shielded room during subject brain signal acquisition and testing [1,22]. In addition, the monetary value of MEG sensors are orders of magnitude greater than EEG, which complements a lower signal-to-noise ratio and lucid signal acquisition for MEG Subject classification. We illustrate the MEG Subject Brainwave classification and implementation on iOS Mobile applications in the subsequent section(s) of this manuscript.


iOS Mobile Application for Wireless Portable EEG BCI Interface to Flight-Simulator Video Game
Emotiv and OpenVibe Wireless Portable EEG
Filtered Waveforms on Channel FC4
Original Signal
Nonlinear phase IIR output
Zero-phase IIR output
Time (s)
Filtered EEG Brain Signal
Variational Bayesian Factor Analysis Algorithm
Machine Learning Algorithm
iOS Application using OpenGL ES 2.0 and GLKit
iOS Mobile Application
Carrier
3:09 AM
iOS Mobile Application for MEG BCI Interface to Apple iOS Warfighter with greater than 90% classification performance
MEG Subject Data
Filtered MEG Brain Signal
Machine Learning Algorithm
iOS Application using OpenGL ES 2.0 and GLKit
iOS Mobile Application
Lawrence Livermore National Laboratory
Option:UCRL#
Option:Additional Information
46

(**a**)

(**b**)

Figure 31. *Cont.*



MEG BCI Interface to Apple iOS Frozen Videogame Mobile Application

MEG Subject Data

Filtered MEG Brain Signal

Variational Bayesian Factor Analysis Algorithm

$\bar{\Lambda} = R_{yx}\psi^{-1}$

$\psi = R_{xx} + \alpha$

$R_{yx} = \sum_{n=1}^{N} y_n \bar{x}_n^T$

$R_{xx} = \sum_{n=1}^{N} \bar{x}_n \bar{x}_n^T + N\Gamma$

$R_{yy} = \sum_{n=1}^{N} y_n y_n^T$

$\alpha^{-1} = diag(\frac{1}{K}\bar{\Lambda}^{-T}\lambda\bar{\Lambda} + \psi^{-1})$

$\lambda^{-1} = \frac{1}{N+L} diag(R_{yy} - \bar{\Lambda}R_{yx}^T)$

$F = \frac{N+L}{2}\log|\lambda| - \frac{1}{2}\sum_{n=1}^{N} y_n^T\lambda y_n - \frac{N}{2}\log|\Gamma| + \frac{1}{2}\sum_{n=1}^{N} \bar{x}_n^T\Gamma\bar{x}_n + \frac{K}{2}\log|\alpha| - \frac{1}{2}Tr(\bar{\Lambda}^T\lambda\bar{\Lambda}\alpha)$

Machine Learning Algorithm

Nazzy IronMan with Apple iOS Frozen Videogame Application

iOS Mobile Application

Lawrence Livermore National Laboratory

Option:UCRL#

Option:Additional Information

47

(**c**)

Figure 31. (**a**) iOS Mobile Application of Warfighter Videogame using OpenGL ES 2.0 and GLKit to evade or chase aerial targets. (**b**) iOS Mobile Application of Warfighter Videogame using OpenGL ES 2.0 and GLKit to evade or chase aerial targets can be interfaced to MEG Subject Brain Signal Data with over 90% classification performance. (**c**) Nazzy IronMan with Apple iOS Frozen Videogram Application can be interfaced to with MEG Subject Brain Signal Data with over 90% classification performance.

3.6. MEG Subject Data BCI iOS Mobile Applications Integration

The MEG Subject Brainwave data demonstrated performance greater than 92% for mid and post- movement thought activity. Thus, the interfacing of iOS Mobile Applications using classified brainwaves as control signals for a user's intent was a simple implementation, as in Phase I. The interfacing to iOS Mobile Applications was the quintessential usage of the BCI technology since the iOS applications developed in this project were written in Objective-C and the BCI technology in Phase 1 retrieving and utilizing machine learning algorithms [23–25] to classify brainwaves was written in C/C++, demonstrated in Figure 31b,c, below. The other aspects of Phase I & Phase II of integrating the MEG Subject's brainwave performance and videogame analytics into the Hadoop Ecosystem, MongoDB, and Cassandra were written in the Java programming language.

The final and near future phase of the iOS Mobile Application involves the collection of user statistics displayed on an iOS mobile application with output yielded for each user′s videogame analytics' and dynamic biometric features, such as a user's UITapGestureRecognizer tap speed or thought movement processes, an illustration of this premise is displayed in Figure 32, below, can be stored and analyzed in a MongoDB database with multiple MEG Subject′s collection or a Cassandra MEG Subject keyspace.

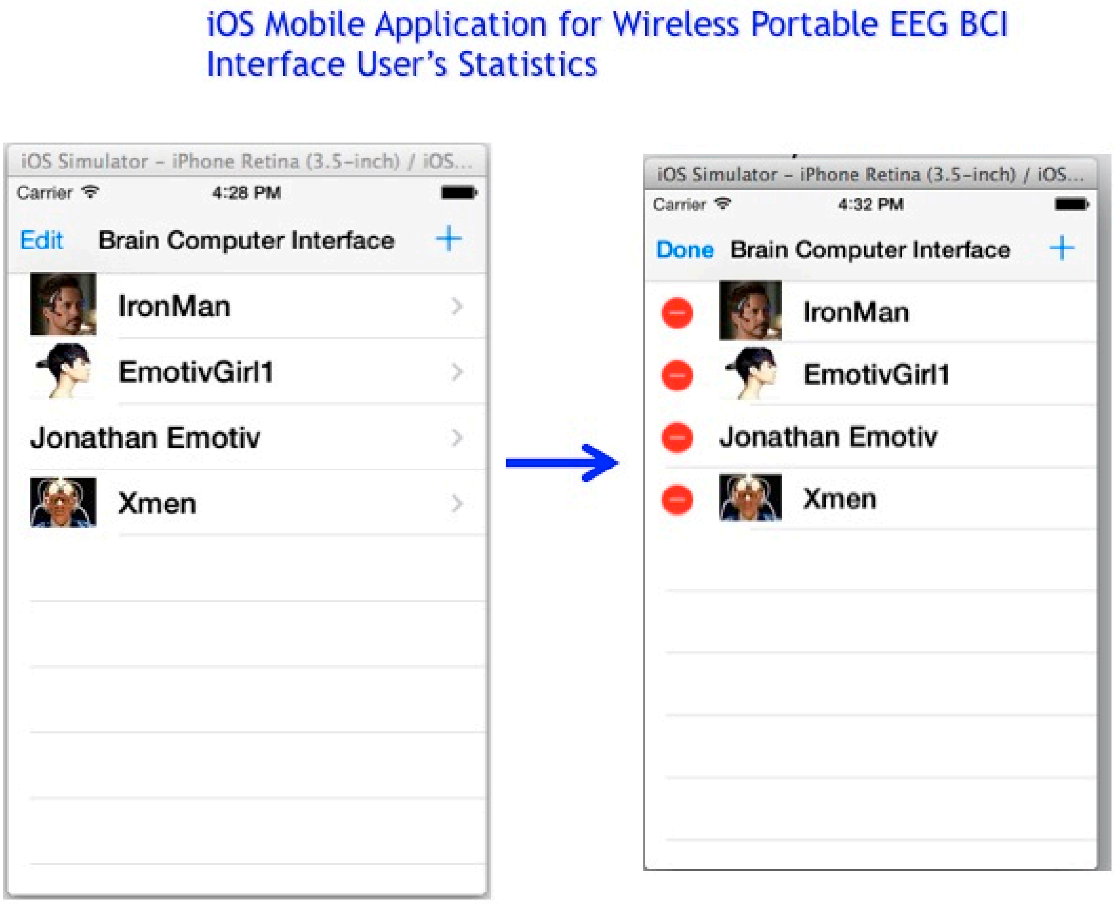

iOS Mobile Application for Wireless Portable EEG BCI Interface User's Statistics
iOS Simulator – iPhone Retina (3.5-inch) / iOS...
Carrier
4:28 PM
Edit
Brain Computer Interface
+
IronMan
EmotivGirl1
Jonathan Emotiv
Xmen
iOS Simulator – iPhone Retina (3.5-inch) / iOS...
Carrier
4:32 PM
Done
Brain Computer Interface
+
IronMan
EmotivGirl1
Jonathan Emotiv
Xmen

Figure 32. iOS Mobile Application of Warfighter Videogame using OpenGL ES 2.0 and GLKit for online user's game analytics and dynamic biometrics.

4. Conclusions

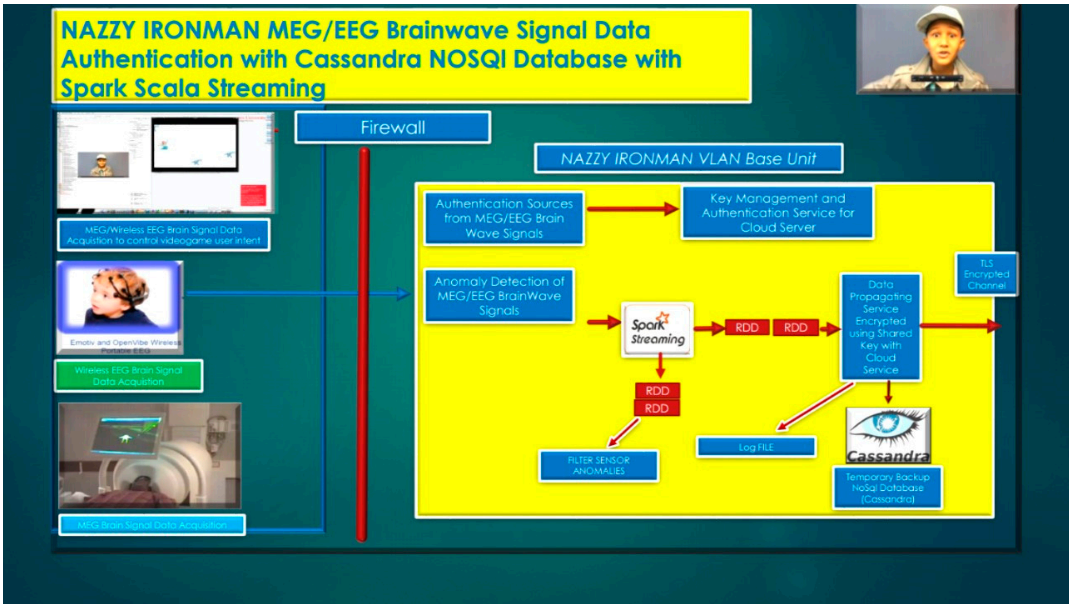
We are currently extending in several directions the machine-learning module that infers user intent from data. We delineate two of these directions here.

First, the classification algorithm that is described in this paper is based on modeling the data as i.i.d. Gaussian (conditioned on the mixing matrix). However, real MEG data are non-Gaussian and exhibit strong temporal correlations. A model that accounts for those features would describe the data more accurately and could therefore lead to improved classification and performance. We are exploring several extensions of our model, including formulating a time-frequency version to handle temporal correlation and replacing the factor model with a mixture of Gaussian distribution to handle non-Gaussianity.

Second, the present algorithm is designed for binary classification tasks. However, in the majority of BCI applications, the user has several separate and distinct, specific intents [26]. For example, in a flight simulator application, in addition to moving the plane left and right, the user may wish to move it up and down, to rotate it at different angles, and to fly it at different speeds. We are therefore extending our model to handle tasks involving more than two classes.

In Phase II of the BCI project, a non-real-time MEG brainwave CTF files and EEG Emotiv & OpenVibe brainwave files of thought movements [27,28] were analyzed while using signal processing and machine learning algorithms [26] for feature extraction and classification for an iOS mobile application using MongoDB and Cassandra as the MEG/EEG brain signal processing data warehouse. At specific places along the warfighter simulation, the user's intention to control the warfighter via button presses or hard-wired brainwaves becomes more challenging to avoid aerial targets and fire projectiles.

In the near future, we are developing better machine learning classifier and signal processing algorithms to ameliorate the signal-to-noise ratio, anomaly detection of MEG/EEG brainwave signals, and to perform brain-wave security [29], authentication classification in real-time online internet applications (e.g. NAZZY IronMan MEG/EEG VLAN Base Unit), as illustrated in Figure 33 below; with future outcomes investigating the usage of behavioral monitoring of cognitive state for "thought-actions", based on historical analysis of brainwave signature data imposed by a given stimulus.

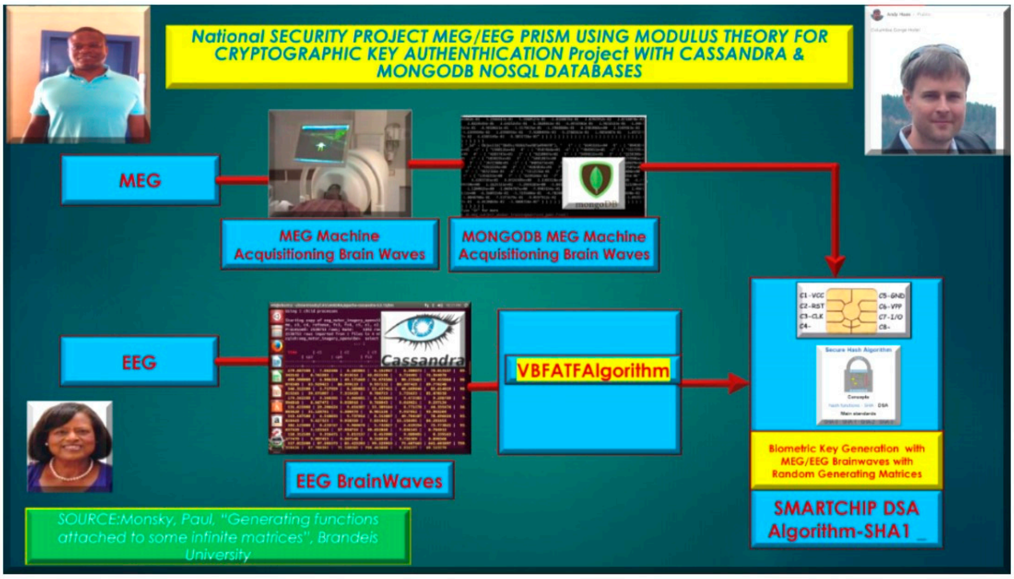


NAZZY IRONMAN MEG/EEG Brainwave Signal Data Authentication with Cassandra NOSQl Database with Spark Scala Streaming

Firewall
NAZZY IRONMAN VLAN Base Unit
MEG/Wireless EEG Brain Signal Data Acquistion to control videogame user intent
Emotiv and OpenVibe Wireless Portable EEG
Wireless EEG Brain Signal Data Acquistion
MEG Brain Signal Data Acquisition
Authentication Sources from MEG/EEG Brain Wave Signals
Key Management and Authentication Service for Cloud Server
Anomaly Detection of MEG/EEG BrainWave Signals
Spark Streaming
RDD
RDD
RDD
RDD
Data Propagating Service Encrypted using Shared Key with Cloud Service
TLS Encrypted Channel
FILTER SENSOR ANOMALIES
Log FILE
Cassandra
Temporary Backup NoSql Database (Cassandra)

Figure 33. Nazzy Ironman MEG/EEG (Virtual LAN) VLAN Base Unit for Security Authentication.

Secondly future directions, involve MEG/EEG brainwave signals for cryptographic key authentication using modulus theory based on generating functions utilizing NoSQL databases, such as Cassandra, MongoDB, and components of the Hadoop Ecosystem, such as Hive and Spark Directed Acyclic Graph (DAG) [30–33]. The usage of Spark DAG and the Spark Machine Learning Library and GraphX to develop biometric key generation based on the utilization of MEG/EEG brain waves for subject authentication. The premise of biometric key generation based on the subject MEG/EEG brainwaves [34] for authentication is based on abstract algebra and combinatoric techniques developed by, Monsky, Paul, "Generating Functions attached to some infinite matrices", at Brandeis University published in The Electronic Journal of Combinatorics, as illustrated in Figure 34, below.



National SECURITY PROJECT MEG/EEG PRISM USING MODULUS THEORY FOR CRYPTOGRAPHIC KEY AUTHENTHICATION Project WITH CASSANDRA & MONGODB NOSQL DATABASES

MEG

MEG Machine Acquisitioning Brain Waves

MONGODB MEG Machine Acquisitioning Brain Waves

EEG

EEG BrainWaves

VBFATFAlgorithm

Biometric Key Generation with MEG/EEG Brainwaves with Random Generating Matrices

SMARTCHIP DSA Algorithm-SHA1

SOURCE:Monsky, Paul, "Generating functions attached to some infinite matrices", Brandeis University

Figure 34. MEG/EEG Cryptographic Key Authentication utilizing MEG/EEG brainwaves with Cassandra and MongoDB NoSQL databases.

Our long-term research goal is to develop an MEG/EEG VLAN [35–37], Base Unit for Security Authentication of MEG/EEG neuromagnetic brainwave signals [38] to detect and classify authenticated MEG/EEG brainwaves and to reject anomalies and outliers [39] that are demonstrated in Figure 33 [40–43]. Furthermore, the usage of cryptographic key generation [44,45], based on the biometric authentication of a MEG/EEG subject's brainwaves will be researched and developed and coupled with DSA SHA-1 algorithm encryption techniques for Cassandra and MongoDB cryptographic biometric databases, as illustrated in Figure 34, above.

Funding: This research was originally funded by 2007 Lawrence Livermore National Laboratory DOE TechBase grant and no other external funding was received.

Acknowledgments: The Lawrence Livermore National Laboratory, DOE TechBase Grants, Tulane University School of Medicine & School of Social Work, and Northeastern University Department of Information Systems Management Program & Information Assurance Program under the College of Computer Science and Information Sciences, Brandeis University Department of Information Assurance, and Brandeis University Department of Mathematics support this paper.

Conflicts of Interest: The author declares no conflict of interest.

References

1. McClay, W.A.; Yadav, N.; Ozbek, Y.; Haas, A.; Attias, H.T.; Nagarajan, S.S. A Real-Time Magnetoencephalography Brain-Computer Interface Using Interactive 3D-Visualization and the Hadoop Ecosystem. *J. Brain Sci.* **2015**, *5*, 419–440. [CrossRef] [PubMed]
2. Sekihara, K.; Sahani, M.; Nagarajan, S.S. A simple nonparametric statistical thresholding for MEG spatial-filter source reconstruction images. *Neuroimage* **2005**, *27*, 368–376. [CrossRef] [PubMed]
3. MongoDB. Available online: https://www.mongodb.com (accessed on 17 August 2018).
4. Cruz-Hernandez, J.M. Systems and Methods for Haptically-Enabled Neural Interfaces. U.S. Patent 20170199569 A1, 24 July 2018.
5. Yongwook, C. Eye-Brain Interface (EBI) System and Method for Controlling Same. U.S. Patent 2018/0196511 A1, 12 July 2018.
6. Wijman, T. Mobile Revenues Accountfor More Than 50% of the Global Games Marketas It Reaches $137.9 Billionin 2018. Available online: https://newzoo.com/insights/articles/global-games-market-reaches-137-9-billion-in-2018-mobile-games-take-half/ (accessed on 17 August 2018).
7. Chulis, K. *Big Data Analytics for Video, Mobile, and Social Game Monetization: Understand and Influence Profitable Consumer Behavior*; IBM Corporation: Armonk, NY, USA, 2012.

8. Georgopoulos, A.P.; Langheim, F.J.; Leuthold, A.C.; Merkle, A.N. Magnetoencephalographic signals predict movement trajectory in space. *Exp. Brain Res.* **2005**, *167*, 132–135. [CrossRef] [PubMed]
9. Kensuke, S.; Maneesh, S.; Nagarajan, S.S. Localization bias and spatial resolution of adaptive and non-adaptive spatial filters for MEG source reconstruction. *Neuroimage* **2005**, *4*, 1056–1067.
10. Suhail, K. Technical Seminar on "Emotiv Epoc/EEG/BCI". Available online: http://www.slideshare.net/psycllone/emotiv-epoceegbci (accessed on 29 March 2011).
11. Martin, S.; Wolfgang, R.; Martin, B. *Adaptive SVM-Based Classification Increases Performance of a MEG-Based Brain-Computer Interface (BCI)*; Springer: Berlin, Germany, 2012.
12. Sekihara, K.; Hild, K.; Nagarajan, S.S. Influence of high-rank background interference on adaptive beamformer source reconstruction. In Proceedings of the International Conference for Bioelectromagnetism and Brain Electromagnetic Tomography and Non-invasive Functional Source Imaging, Minneapolis, MN, USA, 12–15 May 2005.
13. Attias, H. ICA, graphical models, and variational methods. In *Independent Component Analysis: Principles and Practice*; Roberts, S., Everson, R., Eds.; Cambridge University Press: Cambridge, UK, 2001; pp. 95–112.
14. Long, C.J.; Purdon, P.L.; Temereanca, S.; Desal, N.U.; Hamalainen, M.S.; Brown, E.N. State-space solutions to the dynamic magnetoencephalography inverse problem using high performance computing. *Ann. Appl. Stat.* **2011**, *5*, 1207–1228. [CrossRef] [PubMed]
15. Emotiv Systems. Available online: www.emotiv.com (accessed on 17 August 2018).
16. Sanei, S.; Chambers, J.A. EEG Signal Processing. In *Fundamentals of EEG Signal Processing Centre of Digital Signal Processing*; Cardiff University: Cardiff, UK; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2007.
17. Hämäläinen, M.; Hari, R.; Ilmoniemi, R.J.; Knuutila, J.; Lounasmaa, O.V. Magnetoencephalography: Theory, instrumentation, and applications to noninvasive studies of the working human brain. *Rev. Mod. Phys.* **1993**, *65*, 413–497. [CrossRef]
18. Apache Cassandra. Available online: http://cassandra.apache.org/ (accessed on 17 August 2018).
19. Eben, H. *Cassandra: The Definitive Guide*; O'Reilly Media: Sevvan, CA, USA, 2011.
20. Wang, L.; Chen, D.; Ranjan, R.; Khan, S.U.; Kolodziej, J.; Wang, J. Parallel processing of massive EEG Data with MapReduce. In Proceedings of the IEEE 18th International Conference on Parallel and Distributed Systems, Singapore, 17–19 December 2012.
21. Wang, Y.; Goh, W.; Wong, L.; Montana, G. Random forests on hadoop for genome-wide association studies of multivariate neuroimaging phenotypes. *BMC Bioinform.* **2013**, *14* (Suppl. S16). [CrossRef] [PubMed]
22. Mellinger, J.; Schalk, G.; Braun, C.; Preissi, H.; Rosenstiel, W.; Birbaumer, N.; Kubler, A. An MEG-based brain-computer interface (BCI). *Neuroimage* **2007**, *36*, 581–593. [CrossRef] [PubMed]
23. Attias, H. Planning by probabilistic inference. In Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, Key West, FL, USA, 3–6 January 2003.
24. Gross, J.; Ioannides, A.A. Linear transformations of data space in MEG. *Phys. Med. Biol.* **1999**, *44*, 2081–2097. [CrossRef] [PubMed]
25. Lal, T.N.; Schröder, M.; Hill, N.J.; Preissl, H.; Hinterberger, T.; Mellinger, J.; Bogdan, M.; Rosenstiel, W.; Hofmann, T.; Birbaumer, N.; et al. A brain computer interface with online feedback based on magnetoencephalography. In Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 7–11 August 2005; pp. 465–472.
26. Attias, H. A variational bayesian framework for graphical models. *Adv. Neural Inform. Process. Syst.* **2000**, *12*, 209–215.
27. Wolpaw, J.; McFarland, D. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *Proc. Natl. Acad. Sci. USA* **2004**, *101*, 17849–17854. [CrossRef] [PubMed]
28. Millán, J.R.; Rupp, R.; Müller-Putz, G.R.; Murray-Smith, R.; Giugliemma, C.; Tangermann, M.; Vidaurre, C.; Cincotti, F.; Kübler, A.; Leeb, R.; et al. Combining brain-computer interfaces and assistive technologies: State-of-the-art and challenges. *Front. Neurosci.* **2010**, *4*, 1–15. [CrossRef] [PubMed]
29. Smith, K.T. Big Data Security: The Evolution of Hadoop's Security Model. Available online: http://www.infoq.com/articles/HadoopSecurityModel/ (accessed on 14 August 2013).
30. Rodriguez, M. Big Graph Data on Hortonworks Data Platform. Available online: http://hortonworks.com/blog/big-graph-data-on-hortonworks-data-platform/ (accessed on 13 December 2012).
31. Miner, D.; Shook, A. *MapReduce Design Patterns*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2012.

32. Yu, H.; Wang, D. Research and implementation of massive health care data management and analysis based on hadoop. In Proceedings of the Fourth International Conference on Computational and Information Science, Chongqing, China, 17–19 August 2012.
33. The Apache HBase Reference Guide, 2014 Apache Software Foundation. Available online: http://hbase.apache.org/book/client.filter.html (accessed on 31 August 2015).
34. Guger Technologies—g.tec Medical Engineering—g.MOBIlab Mobile Laboratories at Sierningstrasse 14, Schiedlberg, Österreich (Austria).g.tec Developed the First Commercially Available BCI System in 1999 and Now Sells This System in More Than 60 Countries Worldwide. Our Products Work with All Major BCI Approaches (Motor Imagery, P300, SSVEP and Slow Cortical Potentials). Available online: http://www.gtec.at/ (accessed on 17 August 2018).
35. Nagarajan, S.S.; Attias, H.; Hild, K.; Sekihara, K. A graphical model for estimating stimulus-evoked brain responses from magnetoencephalography data with large background brain activity. *Neuroimage* **2006**, *30*, 400–416. [CrossRef] [PubMed]
36. Yuan, P.; Wang, Y.; Wu, W.; Xu, H.; Gao, X.; Gao, S. Study on an online collaborative BCI to accelerate response to visual targets. In Proceedings of the 34th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), San Diego, CA, USA, 28 August–1 September 2012; pp. 1736–1739.
37. Wolpaw, J.; Birbaumer, N.; McFarland, D.; Pfurtscheller, G.; Vaughan, T. Brain-computer interfaces for communication and control. *Electroencephalogr. Clin. Neurophysiol.* **2002**, *113*, 767–791.
38. OpenVibe Datasets Acquisition. Available online: https://www.mindmedia.com/en/products/nexus-32/ (accessed on 17 August 2018).
39. Kaelber, D.; Pan, E.C. The value of personal health record (PHR) systems. *AMIA Annu. Symp. Proc.* **2008**, *2008*, 343–347.
40. Climenser, A.; Awni, H.; Irving, C.; Frank, H.; Stefanie, A. User Input Validation and Verification for Augmented and Mixed Reality Experiences. U.S. Patent 2018/0188807 A1, 5 July 2018.
41. Muller, K.-R.; Anderson, C.; Birch, G. Linear and nonlinear methods for brain-computer interfaces. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2003**, *11*, 162–165. [CrossRef] [PubMed]
42. Attias, H. Independent factor analysis with temporally structured factors. *Adv. Neural Inform. Process. Syst.* **2000**, *12*, 386–392.
43. Garrett, D.; Peterson, D.A.; Anderson, C.W.; Thaut, M.H. Comparison of linear, nonlinear, and feature selection methods for EEG signal classification. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2003**, *11*, 141–144. [CrossRef] [PubMed]
44. Monsky, P. Generating Functions Attached to Some Infinite Matrices. *Electron. J. Comb.* **2011**, *18*, 1–12.
45. Attias, H. Learning in high dimensions: Modular mixture models. In Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics, Key West, FL, USA, 4–7 January 2001; pp. 144–148.