AMERICAN UNIVERSITY OF BEIRUT

TRANSFER ENTROPY CALCULATIONS USING GPUS FOR DETERMINING EPILEPSY FOCUS

by
HASSAN ALI NASSER

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to Department of Electrical and Computer Engineering
of the Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
May 2018

AMERICAN UNIVERSITY OF BEIRUT

TRANSFER ENTROPY CALCULATIONS USING GPUS FOR DETERMINING EPILEPSY FOCUS

by

HASSAN ALI NASSER

Approved by:

Dr. Fadi Karameh, Associate Professor Department of Electrical & Computer Engineering	Advisor
Dr. Ibrahim Abou Faycal, Professor Department of Electrical & Computer Engineering	Member of Committee
Dr. Sami Karaki, Professor Department of Electrical & Computer Engineering	Member of Committee
Dr. Louay Bazzi, Professor Department of Electrical & Computer Engineering	Member of Committee

Date of thesis defense: May 02, 2018

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: Nasser (Last) Hassan (First) Ali (Middle)

● Master's Thesis ○ Master's Project ○ Doctoral Dissertation

[x] I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

[ ] I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes

after : **One ---- year from the date of submission of my thesis, dissertation, or project.**
Two ---- years from the date of submission of my thesis, dissertation, or project.
Three ---- years from the date of submission of my thesis, dissertation, or project.

May 11, 2018

Signature Date

ACKNOWLEDGMENTS

Special thanks are for Prof. Fadi Karameh, Prof. Ibrahim Abou Faycal, Prof. Sami Karaki, and Prof. Louay Bazzi for their support. In addition, special thanks for my family for their special support.

AN ABSTRACT OF THE THESIS OF

Hassan Ali Nasser for Master of Engineering
Major: Electrical and Computer Engineering

Title: Transfer Entropy Calculations Using GPUs for Determining Epilepsy Focus

About one third of epilepsy patients do not respond well to drug treatment. For part of this population, surgical intervention is a promising solution whereby the brain tissue causing seizure initiation is removed (the seizure onset zone or SoZ). Here, several clinical tests are normally conducted to detect and highlight the SoZ, including scalp EEG, MRI, SPECT, PET. Pre-surgical intracranial EEG (IEEG) is collected from multi-electrode arrays placed on the cortical surface to improve SoZ detection. Among the various multivariate techniques used to study the collected IEEG, conditional transfer entropy is very effective in finding causal relationships between the signals in recorded channels due to its generality and exhaustiveness. It is, however computationally very expensive so that using traditional CPUs is impractical. In this thesis, we used GPUs where thousands of cores could run in parallel to implement multi-variate CTE studies. Moreover, we focused on code optimization where the same functions could be implemented in untraditional way so that faster execution is achieved. Part of the code optimization is memory management where different types of memories with different speeds are available on GPUs. We reduced time needed from few days to few hours, thereby rendering the ability of applying CTE to IEEG data more readily attainable for research in SoZ prediction as well as other studies of high-dimensional causal interaction.

CONTENTS

ILLUSTRATIONS

FIGURE PAGE

TABLES

Table Page

CHAPTER 1

INTRODUCTION

Epilepsy is a chronic disease where recurrent seizures occur. A seizure is a hallmark of excessive brain excitation whereby normal motor, sensory, or mental function are negatively impacted. Due to repeated seizures, epilepsy patients and their environment will be under stress and their emotional and social health will be affected. According to World Health Organization 50 million people worldwide have epilepsy [10]. Thirty percent of them don't respond to drug treatment. Part of those patients may subject to epilepsy surgery in which onset zone in the brain is removed. Determining onset zone or seizure focus is of high importance so that the removed brain part could be as small as possible so as to minimize dysfunction while being large enough to stop seizure recurrence. Intracranial Electroencephalography (iEEG) is an effective choice used to measure brain activity where electrodes are put directly on the surface of the brain. IEEG is recorded in a controlled environment to get measures during seizure and in the period just before its occurrence (also called pre-ictal period). Several seizure episodes are commonly recorded to attain sufficient data for statistical confidence.

IEEG is usually formed of hundred electrode array which makes the problem computationally expensive. Therefore, many approaches tried to provide SoZ prediction using methods that, unlike transfer entropy calculations, do not require heavy computations. For example, correlation and coherence [1] [2], direct transfer function (DTF) [3] methods were used in analyzing data. However, results didn't show good accuracy because of problem complexity. We preferred to use transfer entropy as the basis of determining information flow between every two electrodes of the iEEG. Transfer entropy showed precise measure of seizure focus [4] [7]. However, it needs a huge amount of computations to get results. Therefore, to alleviate the prohibitively expensive approach of classic programing, we herein propose to employ parallel programming where Nvidia GPUs formed by three thousand cores to apply transfer entropy methodology. This should give high speed up since transfer entropy computation is highly parallelizable. i.e. every transfer entropy computation is independent of the other one so that they will go all together. In fact, parallel programming using GPUs is an art in itself. There would be three main issues to consider while programming: load balance between cores, memory management, and code optimization. Load balance is attained upon avoiding the status where many cores

complete their mission and you keep waiting other ones to end up for a long period of time. Memory management involves maximizing the use of fast memory access, particularly in this case where millions of memory access are needed. In code optimization, number of instructions is reduced as much as possible and slow instructions are replaced by faster ones.

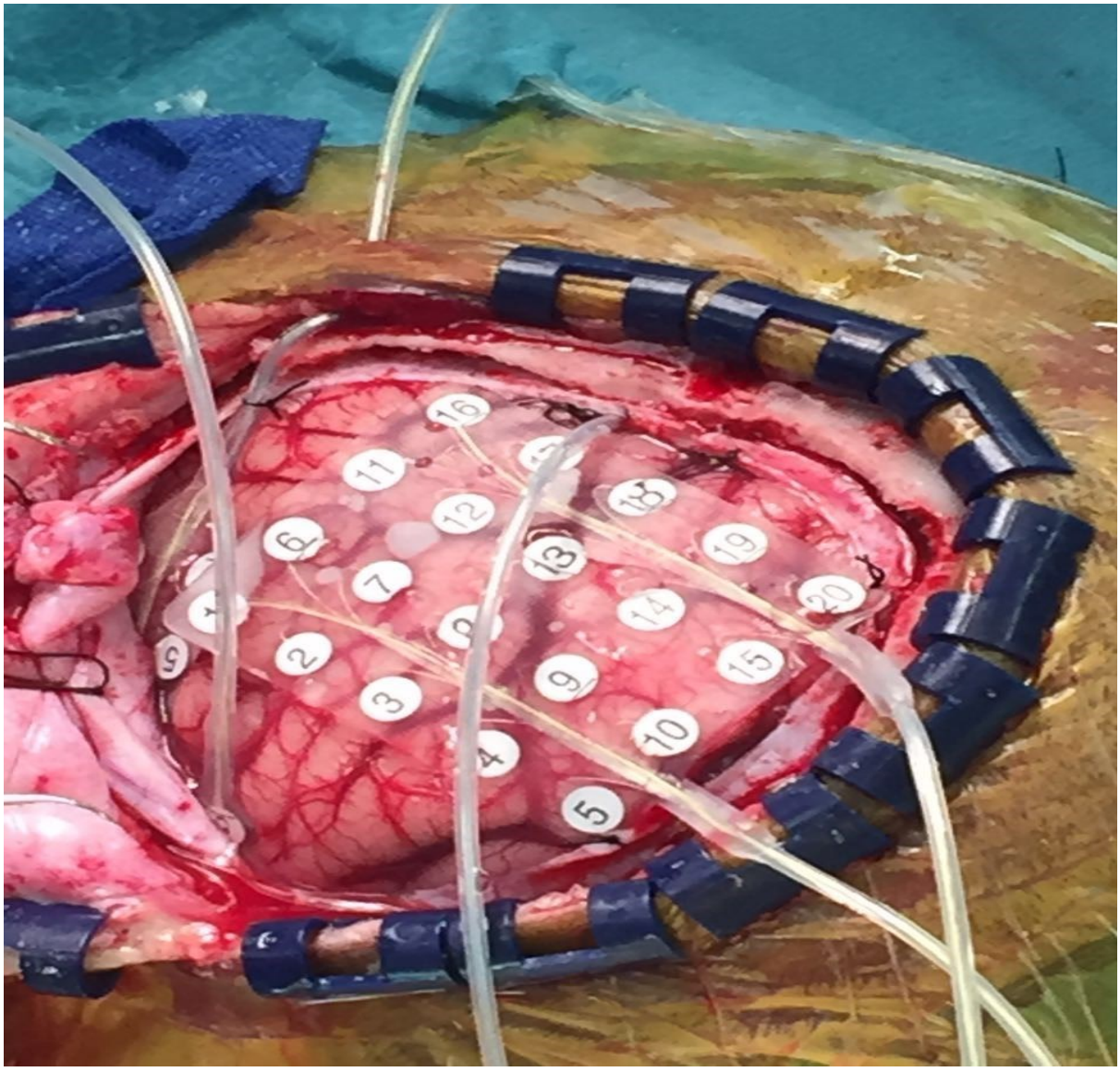

Figure 1

CHAPTER 2

LITERATURE REVIEW

To detect seizure onset zones using IEEG, several computational technique have been utilized that mainly study the directional connectivity or the causal relationship between different recordings. That is, of channel A is found to have a causal effect of channel B, then A is functionally connected to B and has a driver effect on B. In terms of seizures, channel A is preceding Channel B in the seizure. SoZ, therefore is finding the primary region or channel that drives the seizure activity in other areas. The functional connections have been traditionally studied using model-based measures [1][2][3] which use the concept of frequency and hence are inherently linear measures. Information theoretic measures, such as transfer entropy has also been used [4]. [4] introduced good representation of transfer entropy. However, it didn't mention amount of time needed to get results although it needs complex computations. Complex parallelizable computations would always give better time when using GPUs as in this case. [4] experimental results showed a 100 % sensitivity and a false positive rate of 1.79% for SOZ localization. Moreover, [4] introduced an automatic way for SOZ

localization. In [1], determining epileptic focus is done by studying the connectivity between different regions of the brain. They are expecting that epileptic focus has maximum inflow power. However, this methodology gave around 60 percent accuracy only while transfer entropy gave better accuracy as in [4]. [2] uses Generalized Partial Directed Coherence of data collected from intracranial electroencephalographic IEEG. This method also showed a 60 percent accuracy without a mention of time needed for computations. Meanwhile our proposed method has better accuracy with fast results due to using transfer entropy and GPUs. [3] uses directed transfer function (DTF) to detect the directional connectivity using intracranial electrodes. However the study need to be completed. No accuracy is presented to be compared to directional transfer entropy. [5] presented the theory of transfer entropy which is one of the main basis of our algorithm. [6] and [7] presented two tools for computing transfer entropy and conditional transfer entropy respectively. They were used in verifying some results of our research. Moreover, [6] and [7] would be used to determine how much speed up will be achieved. [8] and [9] are good references for programming GPUs using Cuda. We get benefit of them in developing the research software.

CHAPTER 3

BACKGROUND

A. Medical Background

As defined by Meriam-Webster dictionary: epilepsy is a disorder of the nervous system that can cause people to suddenly become unconscious and to have violent, uncontrolled movements of the body. Such recurrent seizure will put epilepsy patient and people surrounding him under stress. The source of this disorder is the brain as a result of excessive electrical discharges in its cells. Around one third of epileptic patients don't respond to drugs [10]. Surgery is a choice in such situation where a part of the brain is removed. Determining epilepsy focus is very important to minimize the size of brain part to be removed. IEEG (intracranial electroencephalogram) collection is used to record brain activity so that signals immediately before and during seizure are analyzed to determine epilepsy focus.

B. Information Transfer within Dynamical Networks

To study dynamical networks there are many tools. One of them is entropy which is defined as:

$$H(X) \triangleq \sum p_i \log_2 p_i$$

Entropy gives an idea about how much of information is there in the signal X. As much as the signal is more stochastic its entropy value will increase. You may use it to get amount of information of two signals X and Y by considering the entropy of their joint or the entropy of one conditioned on the other. However, this has nothing to do with our search about information flow from X to Y.

Another suggested approach could be the use Mutual Information represented by the following equation:

$$M(X,Y) \triangleq \sum p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}$$

Mutual information is an indication of how much the information between X and Y are dependent. When X and Y are independent, p(x,y) will equal p(x)*p(y) and M(X,Y) will turn to zero. However, although the Mutual Information is a good indication about information dependency, it leaks directionality. That is M(X,Y)=M(Y,X) so it is not good indication of information flow.

So if we have an array of random variable and we have information that is flowing in this array, mutual information is not enough to determine the source of information. We need to use transfer entropy TE(J→I) defined as by [5]:

$$TE(J \rightarrow I) \triangleq \sum \mathrm{p}(\mathrm{i}_{\mathrm{n}+1}, i_{\mathrm{n}}^{(k)}, \mathrm{j}_{\mathrm{n}}^{(l)}) \log \left(\frac{\mathrm{p}\left(\mathrm{i}_{\mathrm{n}+1} \left| \mathrm{i}_{\mathrm{n}}^{(\mathrm{k})}, \mathrm{j}_{\mathrm{n}}^{(\mathrm{l})} \right.\right)}{\mathrm{p}(\mathrm{i}_{\mathrm{n}+1} | \mathrm{i}_{\mathrm{n}}^{(\mathrm{k})})}\right)$$

J: represents driver signal or source of information signal.

I: represents target signal or information sink signal.

i_{n+1}: is the sample n+1 of I signal

$i_n^{(k)}$: is vector formed by k successive samples of I ending at sample n.

$j_n^{(l)}$: is vector formed by l successive samples of J ending at sample n.

It's clear from the equation that TE(I→J) is completely different than TE(J→I). As stated by [5], the absence of information flow makes the ratio inside the log to be one and minimizes TE. Oppositely, when there is information flow, terms of ratio inside log diverge will go far from one so that TE increases. An important modification introduced by [5] also is to include a conditional term Z in the equation:

$$TE(J \to I|Z) \triangleq \sum \mathrm{p}(\mathrm{i}_{\mathrm{n}+1}, i_{\mathrm{n}}^{(k)}, \mathrm{j}_{\mathrm{n}}^{(l)}, z_n^{(l)}) \log \left(\frac{\mathrm{p}\left(\mathrm{i}_{\mathrm{n}+1} \left| \mathrm{i}_{\mathrm{n}}^{(\mathrm{k})}, \mathrm{j}_{\mathrm{n}}^{(\mathrm{l})}, z_n^{(l)} \right.\right)}{\mathrm{p}(\mathrm{i}_{\mathrm{n}+1} | \mathrm{i}_{\mathrm{n}}^{(\mathrm{k})}, \mathrm{z}_{\mathrm{n}}^{(\mathrm{l})})}\right)$$

Notice that when Z and J are completely correlated or there is one to one relation between them, their joint probability is the probability of each one: P(J,Z)=P(J)=P(Z). So the fraction inside the log will be equal to one and TE will decrease. In other words, when Z is a similar source of information like J the TE decreases to minimum. So theoretically speaking when we have an array of signals and we are studying TE from channel X to channel Y. We should put the TE under all possible conditions. All possible conditions include taking every individual channel, every pair of channels, every triple of channels, and so on. However, generally this is not practical because it includes a tremendous amount of computations. However, in this paper we will consider every channel individually which is computationally not much more expensive and it's more efficient than non-conditional transfer entropy and conditional transfer entropy with the condition term Z considered one block of all remaining channels. We consider this is adequate enough for our situation where we have single focus of epilepsy.

In conclusion, transfer entropy is a good mathematical basis to locate epileptic source. Moreover, conditional transfer entropy TE(X→Y/Z) is better indication of information flow from X to Y on the condition that source of information is not Z.

C. GPUs Background

GPUs are considered a very good solution for problems where speedup is needed and parallelization is possible. They are used in artificial intelligence, machine learning, bioinformatics, and simulation of physical phenomena. GPUs are multi-core parallel programming hardware designed originally for graphic cards targeted especially for games. CUDA, a C like language support for GPUs, was put in market since 2007 which enabled GPUs to be used as general purpose parallel programming units. Number of cores in GPUs was in the order of hundreds at the beginning. Nowadays, it's in the order of thousands. This gives an idea how much speed up is increasing. Moreover, the speed up is not only due to number of cores but also due to its memory management. Many current GPUs use DDR5 as global memory with data bus width more than 300 bits which enables high memory transfer rate compared to traditional CPUs. Moreover, in GPUs there is what we call shared memory which is much faster than DDR. Shared memory could be used to allocate variables that are commonly used by the program. In

Nvidia, every 32 cores are grouped in a Warp. Warps are SIMT (Single Instruction Multiple Thread).i.e. a warp can execute one instruction at a time for all its cores. However, single instruction doesn't mean single parameter. Each thread of a warp could have its own parameter while fetching and decoding stages are common for all threads of a warp. This causes some limitations when programming GPUs but allows hardware designers to increase number of cores till few thousands which is a big advantage. When within the same warp cores have different instruction to execute, we have what we call divergence. Divergence kills parallelism and should be avoided when possible.

CHAPTER 4

METHODOLOGY

A. General Methodology

We have N channels forming the set C, one channel per electrode. To determine the SOZ, we propose to compute a simplified version of the general transfer entropy by considering triplets of electrodes at a time. Specifically, we define a Triangular Transfer Entropy for every possible two channels C_i and C_j representing electrode i and electrode j respectively, as indicated in the following equation:

$$TTE(C_i \rightarrow C_j) = \max_{\forall\, d1 \in D1} \left(\min_{\substack{\forall\, k \in M-\{i,j\} \\ \forall\, d2 \in D2}} \left(TE\left(C_{i,0} \rightarrow C_{j,d1} \middle| C_{k,-d2}\right)\right)\right) \qquad \text{Eq. I}$$

M = {1,2,…N} where N ∈ ℕ is the number of all channels(electrodes).

C_i = {x_{i1}, x_{i2}, x_{i3}, x_{i4},…, x_{iS} where S is the total number of samples per channel and x_{in} is the sample n value of channel i}

C = {C_i where i ∈ M} is the set of all channels.

$C_{i,d}=\{x_{i(1+d)}, x_{i(2+d)}, x_{i(3+d)}, x_{i(4+d)},\ldots, x_{i(S+d)}\}$ where S is the total number of samples per channel and x_{in} is the sample n value of channel i. It is the shifted version of C_i . $C_{i0}=C_i$.

D1 = {0, 1, 2…, D1max } where D1max $\in \mathbb{N}$ is the maximum delay of sink channel relative to source channel.

D2 = {1, 2, 3…,D2max }where D2max $\in \mathbb{N}$ is the maximum shift of conditional channel relative to source channel.

$$TE(X \to Y\,|Z) \triangleq \sum \mathrm{p}(\mathrm{Y_n}, \mathrm{Y_n^{(d)-}}, \mathrm{X_n^{(d)-}}, \mathrm{Z_n^{(d)-}}) \log \left(\frac{\mathrm{p}\left(\mathrm{Yn}\,\middle|\mathrm{Y_n^{(d)-}}, \mathrm{X_n^{(d)-}}, \mathrm{Z_n^{(d)-}}\right)}{\mathrm{p}(\mathrm{Yn}\,|\mathrm{Y_n^{(d)-}}, \mathrm{Z_n^{(d)-}})}\right) \quad \text{Eq. II}$$

Y_n is the sample n of channel Y.

$X_n^{(d)-}$ is a vector of order d which is a subpart of length d of channel X ending at n-1 .

$Y_n^{(d)-}$ and $Z_n^{(d)-}$ are defined similarly as $X_n^{(d)-}$ so they are defined as follows:

$$X_n^{(d)-} \triangleq (x_{n-d}, x_{n-d+1}, x_{n-d+2}, \ldots, x_{n-1})$$

$$Y_n^{(d)-} \triangleq (y_{n-d}, y_{n-d+1}, y_{n-d+2}, \ldots, y_{n-1})$$

$$Z_n^{(d)-} \triangleq (z_{n-d}, z_{n-d+1}, z_{n-d+2}, \ldots, z_{n-1})$$

High transfer entropy (TE) indicate that there is a flow of information from electrode X to electrode Y on the condition that there is no same information flow from electrode Z

to electrode Y where Z represents all other possible electrodes other than X and Y. Practically, there could be a delay in the transfer of information between X^{-}_{n} and Y^{-}_{n}. Therefore a search for the maximal transfer entropy entails the repetition of the computation of TE from zero delay (instantaneous) to maximum possible delay between X^{-}_{n} and Y^{-}_{n} (as would be inferred from the physical phenomenon). We need to consider only the maximum TE of these delayed versions. As will be demonstrated later, it is generally more effective to consider the effect of each electrode Z other than X and Y individually. That is, the quantity Z in (Eq. II) should represent only one electrode at a time, rather than the total remaining N-2 electrodes as one block. In effect, therefore, TE computation should be repeated for every possible Z, which is number of all electrodes minus two. From all possible Z we consider only the minimum TE whose Z electrode could be source of information flowing to Y similar to the flow from X to Y. As a result, the situation of information flow from X to Y is neglected when there is at least one similar flow from one Z to Y where Z precedes X. Moreover, there could be flow of information from X to Y that should be neglected due to similar flow from Z to Y where Z could be preceding X by a given time but not less than one step. Therefore, computation of TE should be repeated for all possible practical precedence of every Z to

X. Cases where Z is after X in time should not be considered because in such situation X is considered the source of information for both Z and Y. Cases where X and Z are zero shift should be counted as flow from X to Y and from Z to Y. In other words, they will not cancel each other. All these computations are needed for computing the optimum TE between one pair of electrodes naming it TTE (Triangular Transfer Entropy). These computations need to be repeated for all possible electrode pairs. As a result, the computation complexity would be as follows:

$(D_z * (N-2) * D_y * C) * N * (N-1)$

Where:

D_z: Number of all possible forward shift of Z signal.

N: Number of electrodes.

D_y: Number of all possible backward shift of Y signal.

C: The computation complexity of individual TE computation.

$D_z * (N-2) * D_y * C$: is the computation complexity of every single pair of electrodes. For an array of 76 electrodes (N=76), 6 steps possible precedence of Z (D_z =6), 7 steps possible delay between X and Y (D_y =7) computation complexity is: 17,715,600*C.

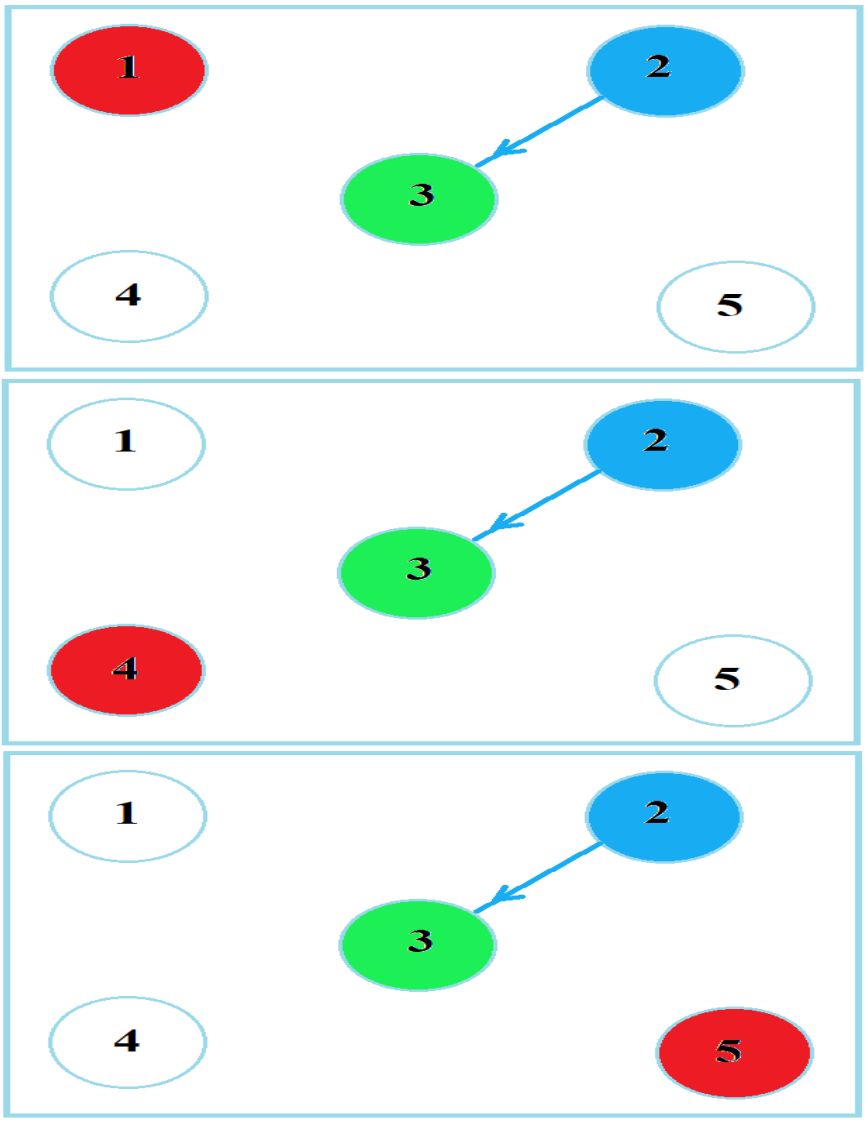

Figure 2

Consider an example of five channels and we need to compute triangular transfer entropy TTE(C_2 -> C_3) as illustrated in figure 2 where source channel is blue (C_2), sink channel is green (C_3), and condition channel is red (C_1,C_4, and C_5 one at a time). So you need to compute every TE under the condition of every red channel separately for all its possible shift and consider only the minimum of all of them. Then you should repeat the process for every possible shift between C_2 and C_3 and consider the maximum of them called TTE(C_2-> C_3).

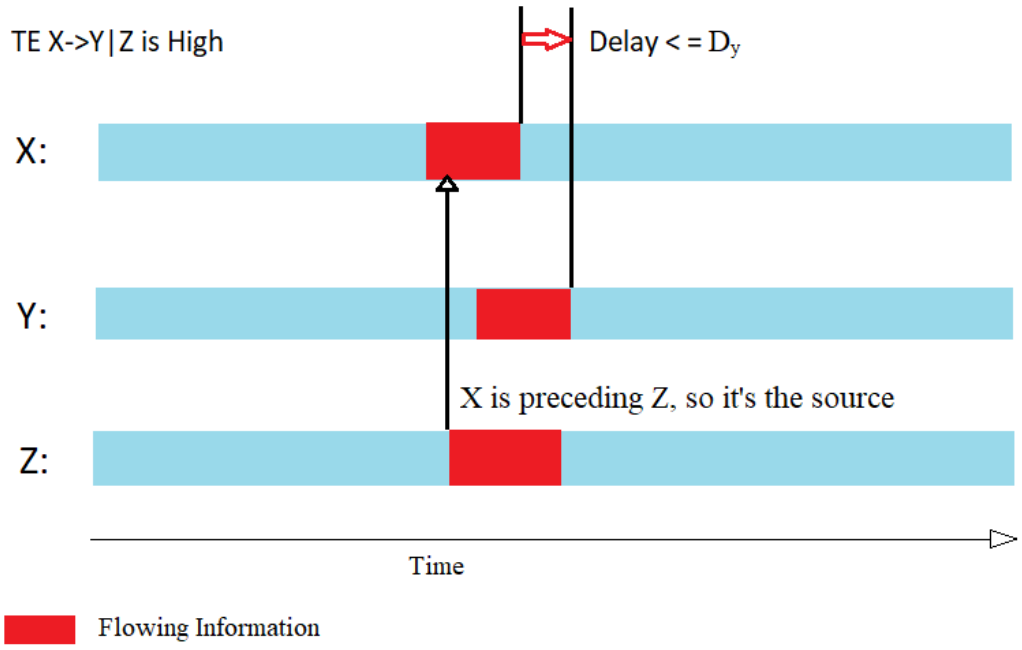

Figure 3

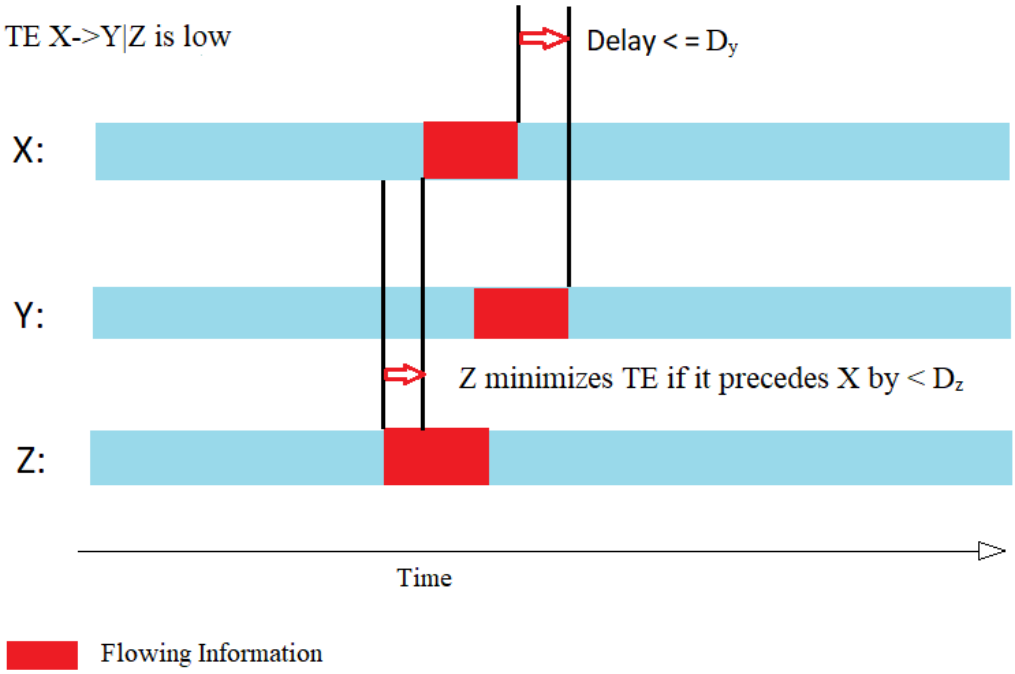

Figure 4

Fig. 3 and Fig. 4 clarify where we have high and low TE value. In few words, when we have multiple similar information flow to electrode Y the earliest source of information within a predefined range is considered the real source of flow and all remaining sources are neglected since their optimum TE will be low although they have flow of information to Y.

In Eq. II, all vectors X^{-}_{n}, Y^{-}_{n}, and Z^{-}_{n} have the same length (d). Moreover they are normalized to the same number of quantization levels (Q). As a result, the number of possible outcomes (PO) in our methodology where Z^{-}_{n} represents one electrode is as follows at a time:

$PO=Q^{3d}$

For Q=5 and d=2: PO= 15625.

Whereas when you consider other methodology where Z^{-}_{n} is a block of all other remaining electrodes. The number of possible outcomes is as follows:

$PO=Q^{dN}$ where N is the total number of electrodes.

For Q=5 and d=2, N=76: PO= 1.75 e+106

We may notice how the big difference between two POs. This will make the first methodology much better approximation of the probabilities introduced in the TE computation. When considering any probability computation number of trials should much greater than number of possible outcomes. Therefore, in our case where the number of outcomes (number of samples) is in the order of thousands we choose d=2 and Q=5 so that better approximation is achieved.

[7] proposed many methods of computing TE: Linear Estimator, Nearest Neighbor Estimator, and Binning Estimator with Uniform Embedding and Non-Uniform Embedding which shows better results for determining SOZ of epilepsy. In our research, we are going to parallelize Binning Estimator Method with Uniform Embedding with some modifications on how to consider the conditional Z. Binning Estimator is a simple good approximation method when you have enough samples to compute probabilities. It's based on the formula:

$$Probability(x) = \frac{Count\ of\ x\ occurence}{Total\ number\ of\ samples}$$

Using Bayes rule which states that P(a,b)=P(a|b)*P(b) the log term in Eq. II could be expressed as follows:

$$\log_2\left(\frac{p(Y_n|Y_n^-, X_n^-, Z_n^-)}{p(Y_n|Y_n^-, Z_n^-)}\right) = \log_2\left(p(Y_n, Y_n^-, X_n^-, Z_n^-)\right) - \log_2(Y_n^-, X_n^-, Z_n^-) +$$

$$\log_2\left(p(Y_n^-, Z_n^-)\right) - \log_2(p(Y_n, Y_n^-, Z_n^-))$$

And TE turned to be :

$$TE(X \to Y|Z) = \sum p(Y_n, Y_n^-, X_n^-, Z_n^-) * \left(\log_2\left(p(Y_n, Y_n^-, X_n^-, Z_n^-)\right) - \right.$$

$$\left.\log_2(Y_n^-, X_n^-, Z_n^-) + \log_2\left(p(Y_n^-, Z_n^-)\right) - \log_2(p(Y_n, Y_n^-, Z_n^-))\right) \qquad \text{Eq. III}$$

We will present below a pseudocode for the three main stages involved in computing TTE (X->Y/Z):

1. Pseudocode1:

To compute the $TE_{X-->Y/Z}$ for a given X, a given shifted version of Y, and a given shifted version of Z we need to:

- Count the number of occurrence of every (Y_n,Y^-_n, X^-_n, Z^-_n)
- Count the number of occurrence of every (Y^-_n, X^-_n, Z^-_n)
- Count the number of occurrence of every (Y^-_n, Z^-_n)
- Count the number of occurrence of every (Y_n ,Y^-_n, Z^-_n)
- Compute the probabilities p(Y_n,Y^-_n, X^-_n, Z^-_n), p(Y_n, X^-_n, Z^-_n), p(Y^-_n, Z^-_n), p(Y_n,Y^-_n,Z^-_n) by dividing the number of corresponding occurrences by the total number of samples.
- Compute TE1=$TE_{X-->Y/Z}$ as in Eq. III

2. Pseudocode2:

- For a given X and Y repeat the computation of TE (as in pseudocode1) for every possible shift of every possible electrode Z.
- Compute TE2= minimum value of all TE computed in step 1 of pseudocode2

3. Pseudocode3:

- For a given X and Y repeat the computation of TE (as in pseudocode2) for every possible shift of Y.
- Compute TE3= maximum value of all TE computed in step1 of pseudocode3. TE3 is considered the optimum approximation of TTE(X->Y/Z):

4. Pseudocode4:

- Repeat the above stated pseudocode parts for every possible combination of X and Y.

- Store the results in a two dimensional array so that they are ready for later comparison and decision. The row index represents the source electrode (driver) and the column index represents the sink electrode (target).

After getting TTE(X->Y/Z) for every X and Y, a criteria is needed to determine if there is a flow from X electrode to Y electrode. A good way is to compute surrogate of TE(X->Y/Z) for every (X,Y) couple and consider it as a reference to compare with TTE(X->Y/Z).

Surrogate pseudocode is as follows:

5. Pseudocode5:

- For a given X and Y repeat the computation of TE (as in pseudocode1) for N far enough delay of Y for every possible electrode Z.
- Compute the average of computed TEs in previous step and consider it as surrogateTE(X,Y)
- Repeat steps 1 and 2 of pseudocode5 for all possible (X,Y) combination and store the results in a two dimensional surrogateTE array.

Then you build a ratio two dimensional array where every element of array is computed by dividing corresponding TTE array element by corresponding surrogateTE array.

$Ratio_{ij} = TTE_{ij} / surrogateTE_{ij}$.

Every $Ratio_{ij}$ is compared to given threshold. If it's greater than this threshold an outflow from electrode i to electrode j is considered positive. For every electrode i, count its number of outflow. Electrode with highest outflow is considered the SOZ of epilepsy.

B. GPU Methodology

In GPUs we may have more than three thousands cores (the basic unit of processing in a computer). Every core would be responsible for the computation of every optimum transfer entropy TTE (pseudocode1 -> peudocode4) including basic computations, different Y shifts, different Z and their different shifts. For N electrodes we have: (N-1)*(N-1) possible combinations of X and Y. For N=76 we have 5625 possibilities. Therefore, all GPU cores will be fully loaded which implies very good speedup. Making every core responsible of computing every TTE makes it independent of all other cores so that it has many advantages:

- No need for synchronization between different cores which may slow down the speed.
- Most if not all needed variables are stored in register memory which is much faster than shared memory and global memory.
- Less divergence per kernel. It's known that divergence kills parallelism. There are few "if" statements in kernels.

All possible Comparison within the same electrode are done before the execution of the main kernel (main function executed by the GPU) and stored in global memory as texture Memory with basic element of one byte length. Texture memory is part of global memory whose access is much faster than traditional global memory due to cashing. Cashing is the moving of memory blocks to a very fast memory (called cash) which is usually near the core so that it has the ability to be very fast. This methodology acquires the algorithm more speedup due to main reasons:

- Texture memory is faster due to cashing.
- For given X^{-}_{n}, Y^{-}_{n}, or Z^{-}_{n} with dimension d, when counting we don't need to fetch all d components of every term. All what we need is to read one cashed Boolean byte instead of d bytes whatever is the value of d.

After counting each term Eq. III expression need to be computed. So we need to compute four logs for all different terms of (Y_n ,$Y^{-}_{n,}$ $X^{-}_{n,}$ Z^{-}_{n}), ($Y^{-}_{n,}$ $X^{-}_{n,}$ Z^{-}_{n}), ($Y^{-}_{n,}$ Z^{-}_{n}), and (Y_n ,$Y^{-}_{n,}$ Z^{-}_{n}). Moreover log is a complex mathematical function (not simple as addition and multiplication) that need much more cycles than simple mathematical functions. Therefore, we computed all possibly needed values of log and store them in a

lookup table in shared memory. Thus to get result of log (p) all what we need is to fetch a value from shared memory which is much faster than invoking the log function. Notice that in some GPUs shared memory is hundred times faster than global memory. Log lookup table is computed once per block of threads which is around 1024 thread (one thread per core). So as overall, in situation with 76 electrode we need to compute the lookup table for six times only. Using lookup table accuracy is preserved since by using binning probability, computation is in the form of:

P=n/N or n/(N-1)

Where n: number of sample occurrence

N: total number of samples

As a result it's enough to partition the probability variable p whose log should be computed to N segments. So extracting probability value from lookup table would be precise since there is no need for interpolation.

CHAPTER 5

RESULTS

A. Test 1

The objective of this test is to show the importance of TTE where every conditional channel is considered individually compared to other algorithm where all conditional Z channels are considered as one block.

Considering synthetic data composed of eight electrodes where there is flow of information from electrode one to electrode two, to electrode three, and to all remaining electrodes till electrode eight as shown in Fig. 5. So practically speaking if unconditional transfer entropy is considered we have high transfer entropy from electrode one to all following, from electrode two to all following but not to electrode one, from electrode three to all following but not to electrode one or two and so on. We build this data by assigning the following signals to every electrode:

Electrode 1= random data.

Electrode n = electrode 1 delayed by (n-1) step for: n not equal to1

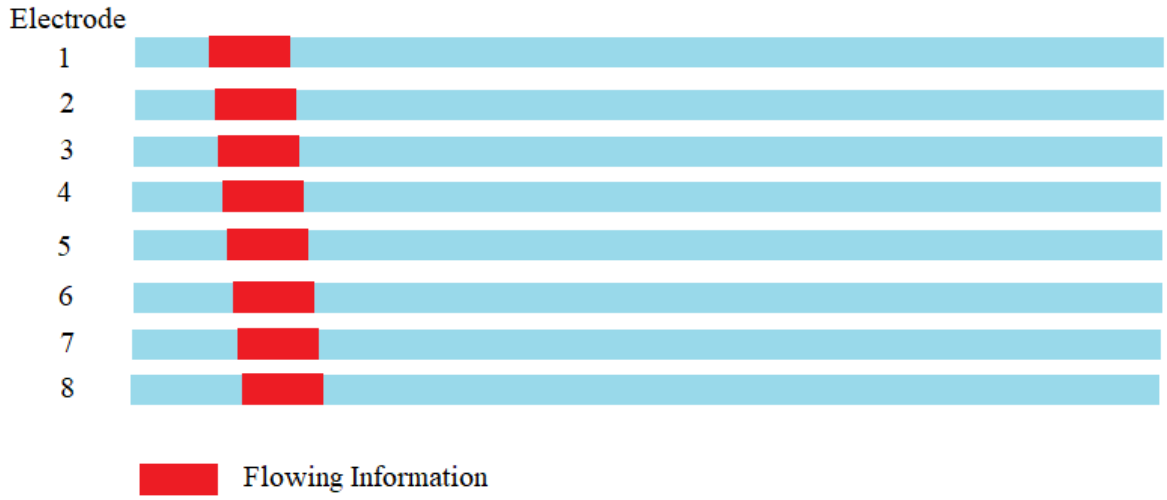

Figure 5

Tables 1 and 2 show the results of applying the TTE algorithm and one bock Z algorithm to the synthetic data. Table 1 shows TE from row to column where all remaining Z electrodes are considered as one block in computing the TE. Table 2 shows TTE from row to column electrodes where all remaining electrodes are considered each one individually.

Electrode	1	2	3	4	5	6	7	8
1	-	0.047	0.047	0.047	0.040	0.045	0.045	0.044
2	0.047	-	0.047	0.047	0.047	0.050	0.050	0.050
3	0.047	0.047	-	0.047	0.047	0.050	0.050	0.050
4	0.048	0.048	0.048	-	0.048	0.048	0.047	0.047
5	0.050	0.050	0.050	0.050	-	0.050	0.044	0.044
6	0.046	0.046	0.046	0.046	0.046	-	0.038	0.038
7	0.050	0.050	0.050	0.050	0.050	0.050	-	0.043
8	0.050	0.050	0.050	0.050	0.050	0.050	0.047	-

Table – 1

Electrode	1	2	3	4	5	6	7	8
1	-	0.871	0.881	0.881	0.881	0.881	0.881	0.881
2	0.838	-	0.026	0.026	0.024	0.007	0.007	0.007
3	0.027	0.027	-	0.027	0.027	0.026	0.007	0.007
4	0.025	0.025	0.025	-	0.023	0.023	0.021	0.006
5	0.026	0.026	0.026	0.026	-	0.024	0.022	0.006
6	0.026	0.026	0.026	0.026	0.026	-	0.023	0.007
7	0.026	0.026	0.026	0.026	0.026	0.026	-	0.007
8	0.025	0.026	0.025	0.025	0.024	0.024	0.024	-

Table – 2

Fig.6 is a plot of table 2 results:

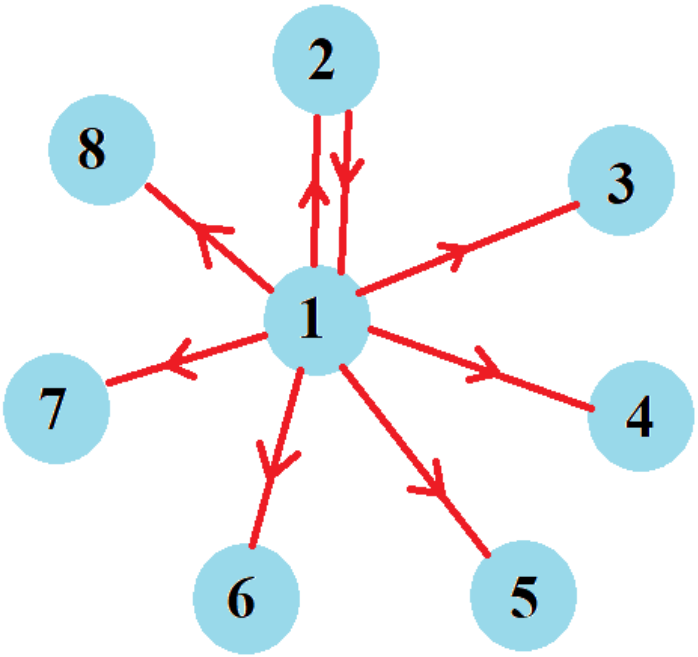

Figure 6

B. Test 2

The objective of this test is to show our TTE algorithm on real data that was previously tested with other algorithms and whose result is known.

TTE was applied on real data taken from [7] for ictal data. We essentially test the data using GPUs by applying our proposed algorithm under the following criteria:

- Fs (sampling rate) = 100 Hz.
- Number of samples is 1000.
- Maximum shift is 60 msec as backward and forward shift for the sink electrode(Y) and for the conditional electrode (Z) respectively. This shift is with respect to the source electrode(X).
- Every conditional electrode is taken individually in the computation.
- Xn^-, Yn^-, and Zn^- are of length 2.

Fig. 9 shows results of [7].

Fig. 7 and Fig. 8 show results of Test 2. These results were taken by computing the ratio:

- R = (optimum value of TE) / (surrogate value of TE).
- If R >= threshold (1.25 * mean value of R in our case).

In Fig. 7 when a flow from source to sink is considered true, it is marked by white square. When there is no flow we mark it by black.

Fig. 8 shows the number of outflow for every electrode.

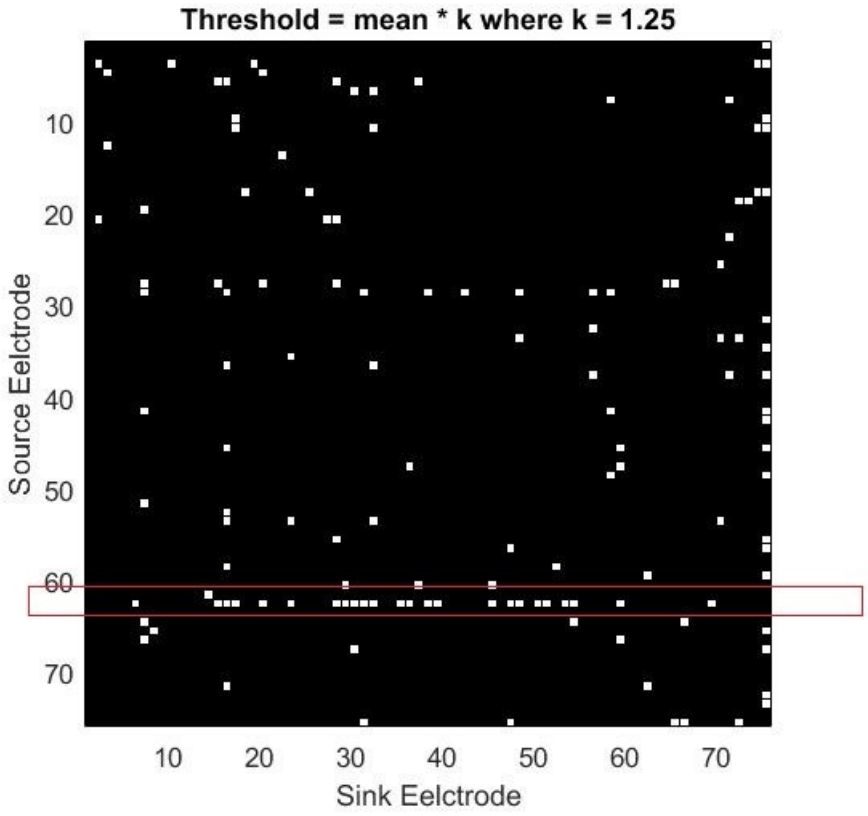

Figure 7

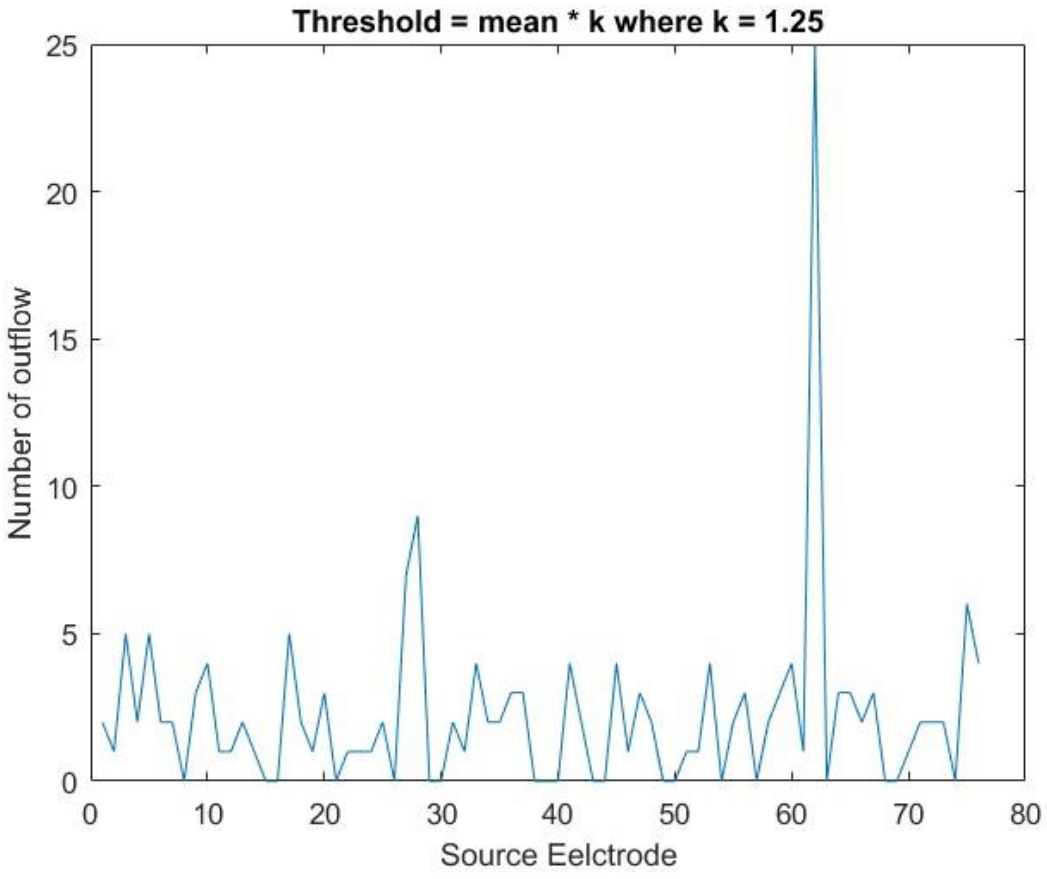
Threshold = mean * k where k = 1.25
Number of outflow
Source Eelctrode
0
5
10
15
20
25
0
10
20
30
40
50
60
70
80

Figure 8

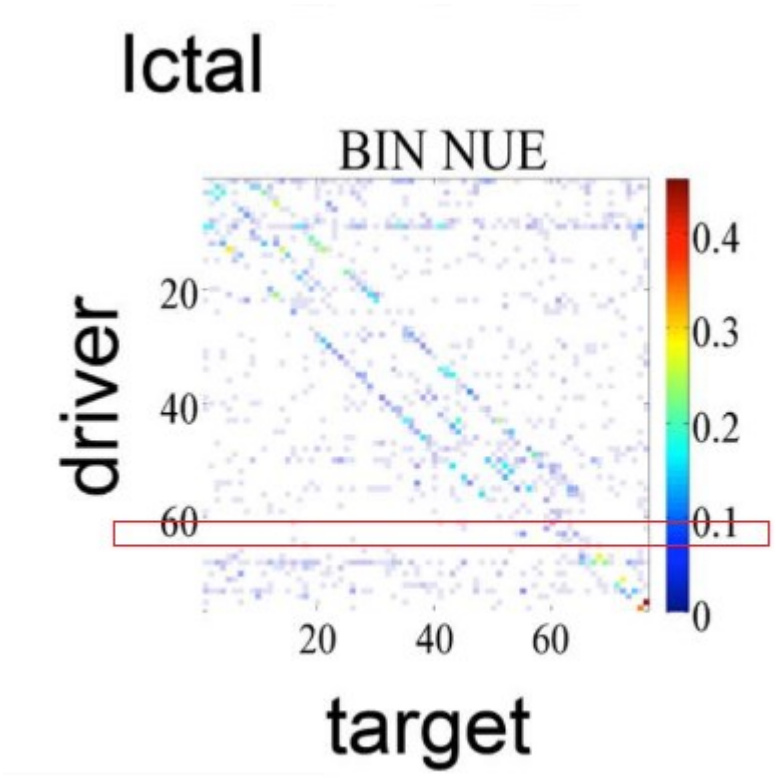

Figure 9

C. Test 3

We repeated Test 2 naming it Test 3 with the following criteria (The only difference is that the conditional Z term in the TE expression is considered to be all other channels as one block)

- Fs (sampling rate) = 100 Hz.
- Number of samples is 1000.
- Maximum shift is 60 msec as backward and forward shift for the sink electrode(Y) and for the conditional electrode (Z) respectively. This shift is with respect to the source electrode(X).
- When considering flow from electrode X to electrode Y all remaining electrodes are considered as one conditional block.
- Xn-, Yn-, and Zn- are of length 2.

R should be >= (k * mean of R) . To consider a true outflow and mark it with a white square. Fig. 10 and Fig. 11 have a k =2.4 meanwhile Fig. 12 and Fig. 13 have a k=1.25 (same k as Test 2) for the same result data.

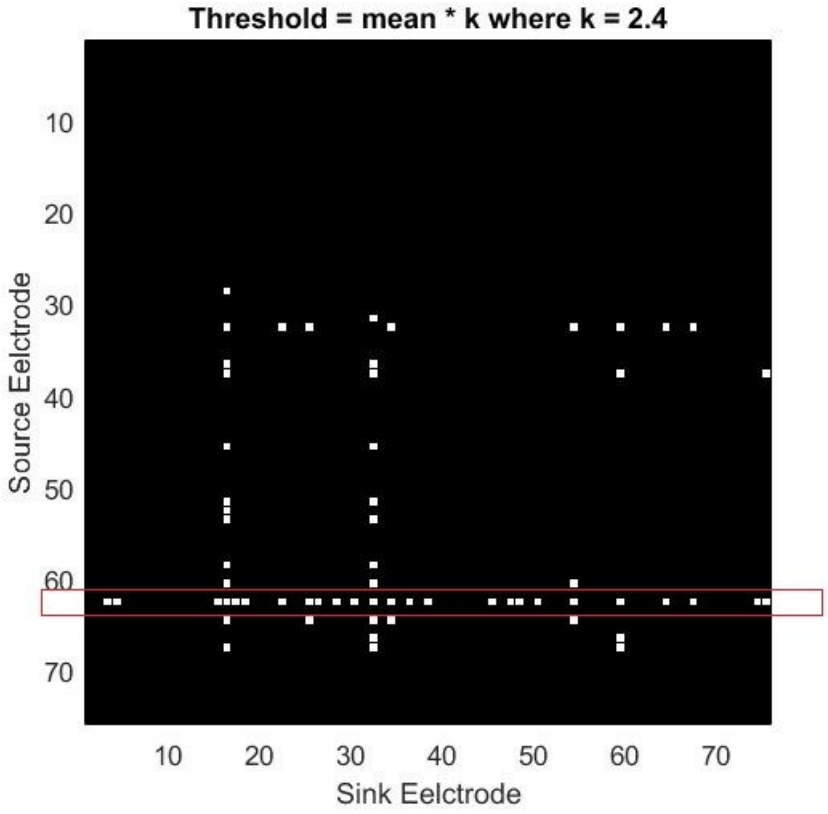
Threshold = mean * k where k = 2.4
Source Eelctrode
Sink Eelctrode
10
20
30
40
50
60
70

Figure 10

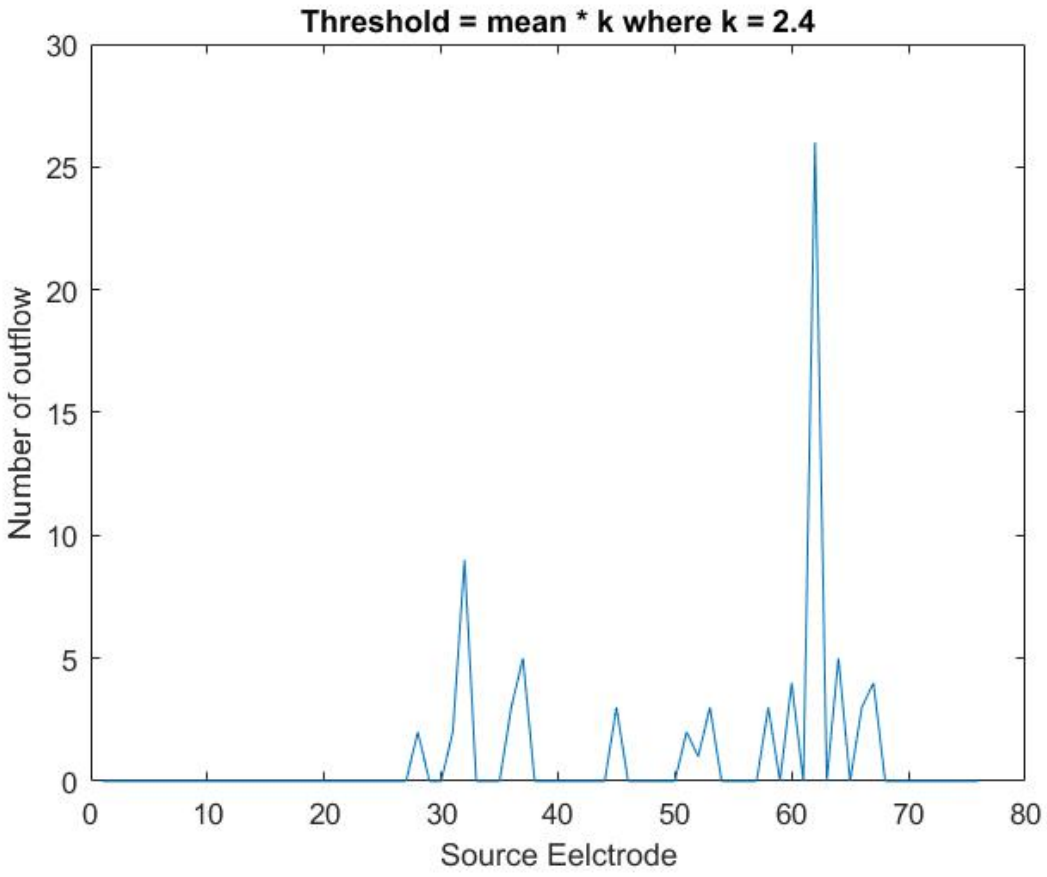
Threshold = mean * k where k = 2.4
Number of outflow
Source Eelctrode
0
5
10
15
20
25
30
0
10
20
30
40
50
60
70
80

Figure 11

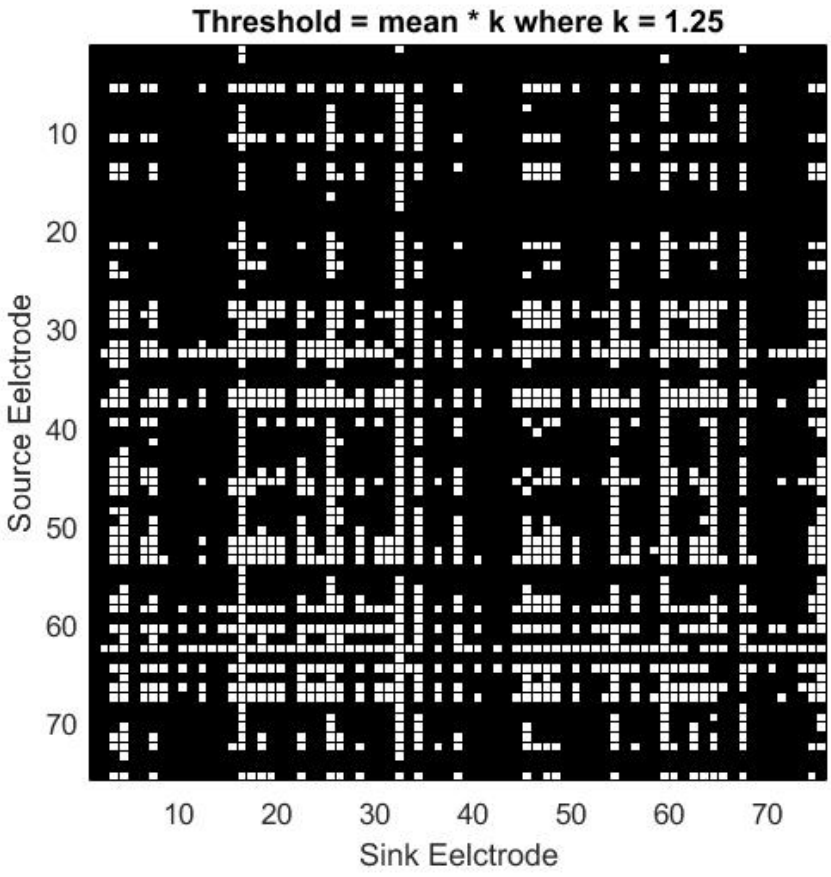
Threshold = mean * k where k = 1.25
Source Eelctrode
Sink Eelctrode
10
20
30
40
50
60
70

Figure 12

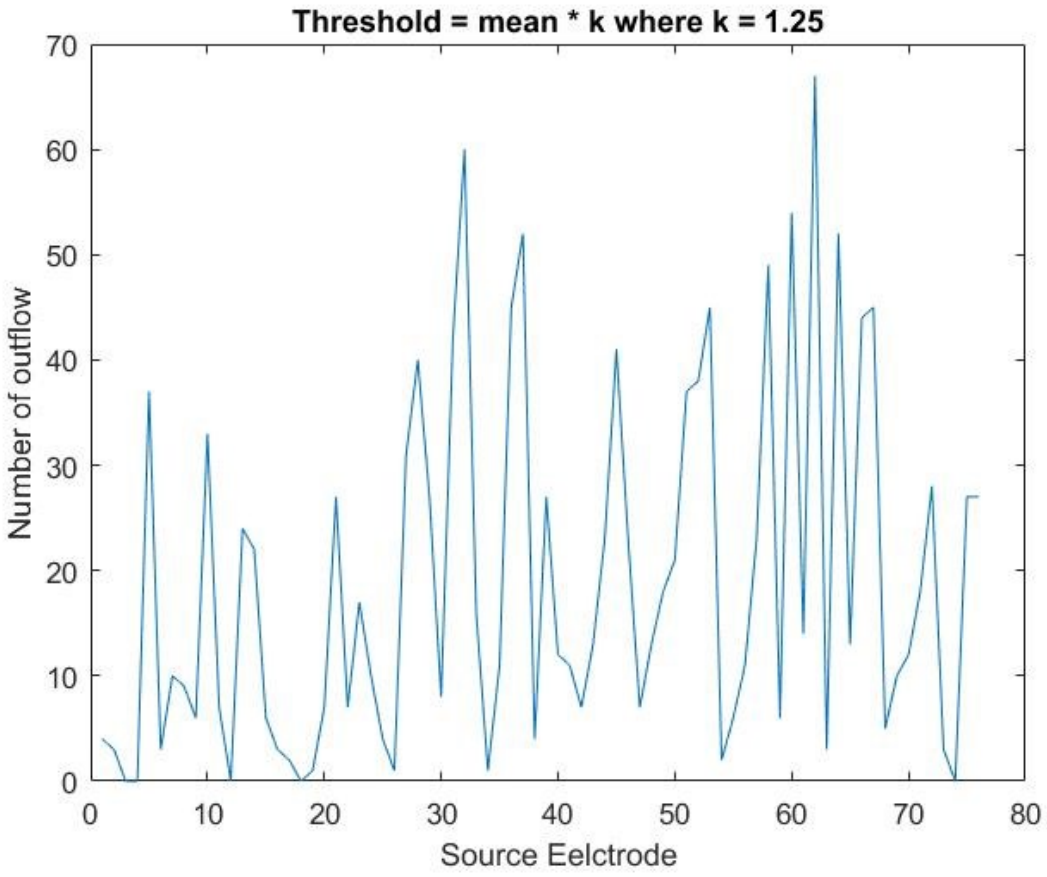

Figure 13

D. Test 4

We repeated Test 2 with the same data using GPUs with the only difference that maximum shift is 20 msec instead of 60 msec and name it Test 3. Test 3 Criteria are as follows:

- Fs (sampling rate) = 100 Hz.
- Samples number is 1000.
- Maximum shift is 20 msec as backward and forward shift for the sink electrode(Y) and for the conditional electrode (Z) respectively. This shift is with respect to the source electrode(X).
- Every conditional electrode is taken individually in the computation.
- Xn-, Yn-, and Zn- are of length 2.

Fig. 14, Fig. 15, and Fig. 16 show results for this experiment.

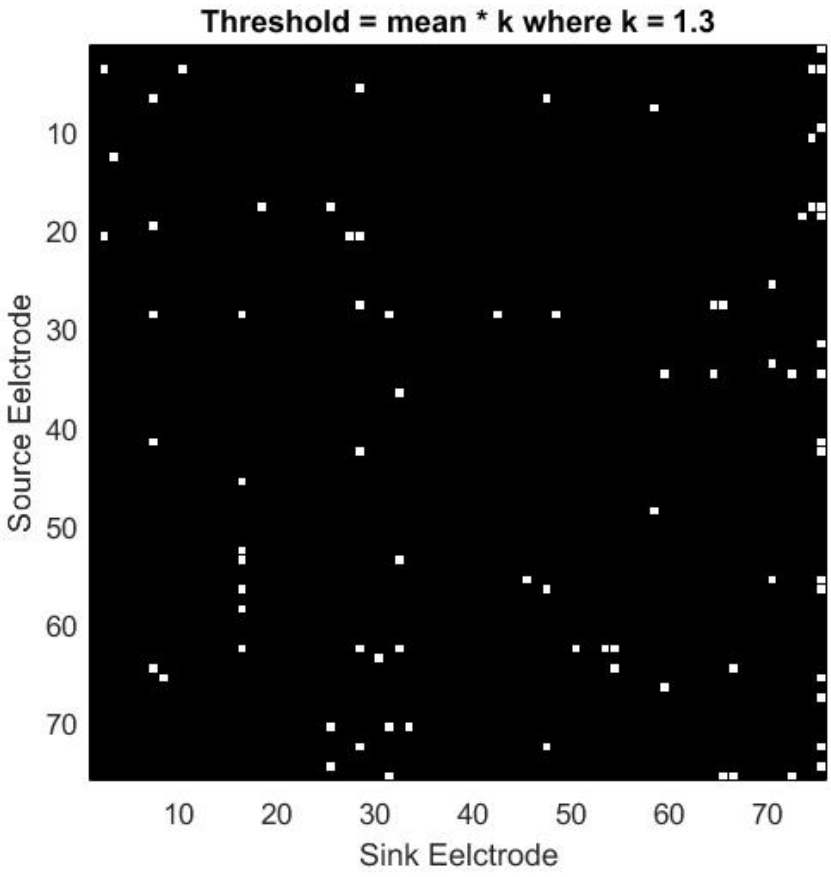
Threshold = mean * k where k = 1.3
Source Eelctrode
Sink Eelctrode
10
20
30
40
50
60
70

Figure 14

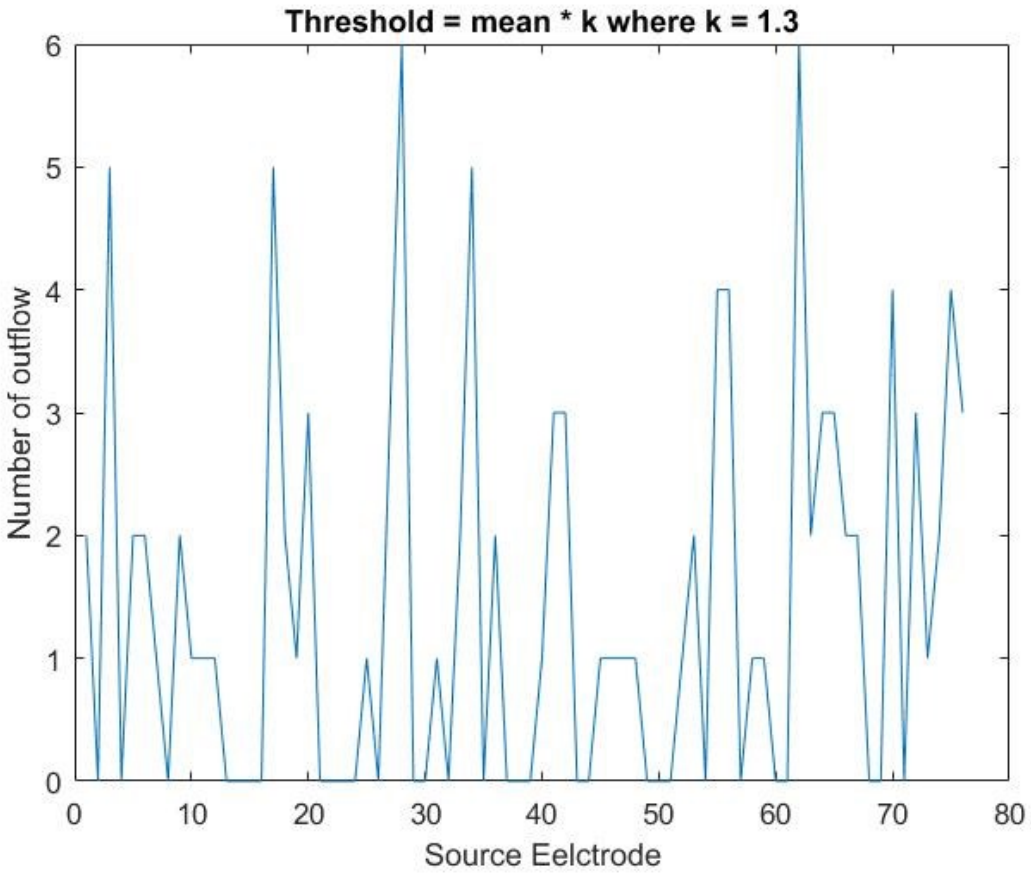
Threshold = mean * k where k = 1.3
Number of outflow
Source Eelctrode
0
1
2
3
4
5
6
0
10
20
30
40
50
60
70
80

Figure 15

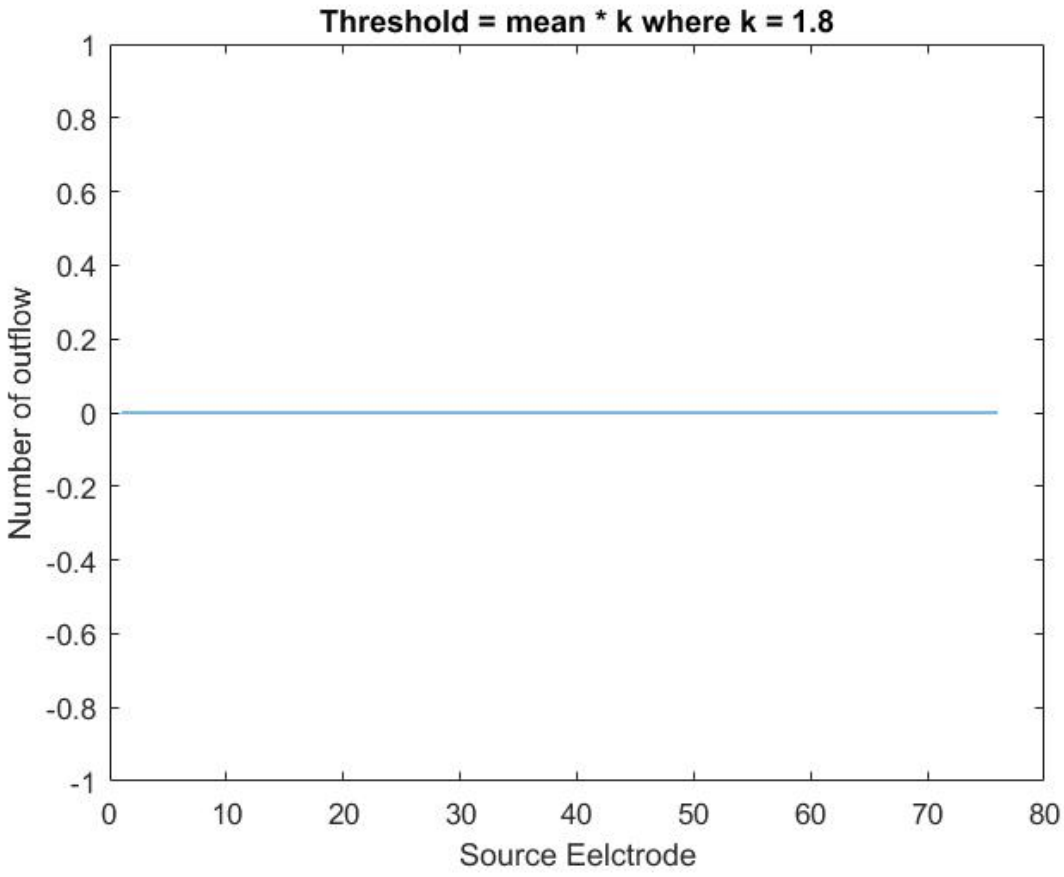
Threshold = mean * k where k = 1.8
Number of outflow
Source Eelctrode
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
10
20
30
40
50
60
70
80

Figure 16

Table - 3 summarizes results of Test 2, Test 3, and Test 4

	Time (seconds)	Individual Conditional Z	Maximum Shift (msec)	Max TE (optimum)	Accuracy	K(threshold= k * ratio of TE/surrogate TE)
Test2 (384 cores-Nvidia Geforce 940MX GPU)	3122	Yes	60	1.5448	Good	1.25
Test2 (3584 cores-Nvidia Titan X-Pascal GPU)	1323	Yes	60	1.5448	Good	1.25
Test3 (384 cores-Nvidia Geforce 940MX GPU)	1052	NO	60	0.6714	Good	2.4
Test3 (384 cores-Nvidia Geforce 940MX GPU)	1052	NO	60	0.6714	Not Good	1.25
Test4 (384 cores-Nvidia Geforce 940MX GPU)	430	Yes	20	0.2782	Bad	No value

Table – 3

E. Test 5

Last experiment named Test 5 was conducted to compare our proposed algorithm with [7] Mute toolbox. However, since there is a lot of differences between

two algorithms we modified many parameters of both so that comparison is fair. Fig. 17 shows the TE where the width of X_n^- , Y_n^- , and Z_n^- ,the order is equal to one. GPU result is to the left and Mute result is to the right.

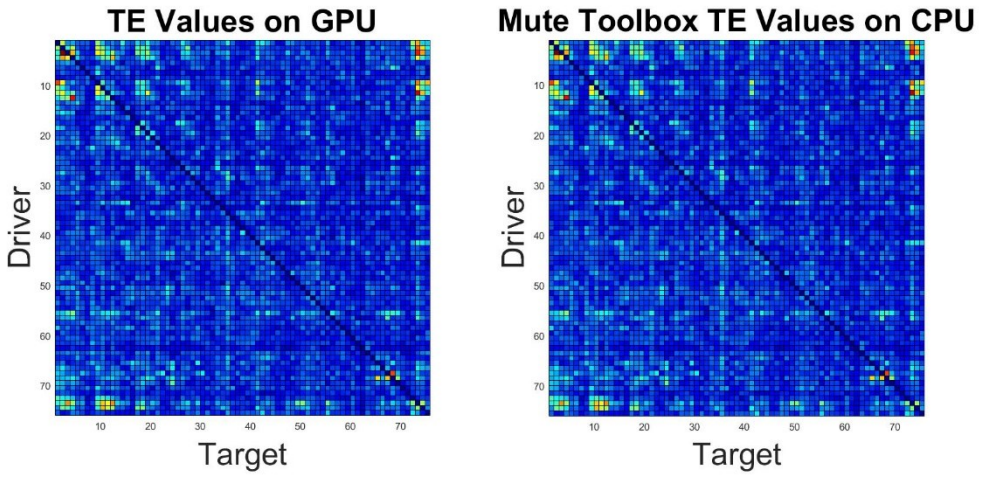

Figure 17

The experiment is repeated with order = 2 and the result is shown in Fig. 18 (GPU result to the left and Mute results to the right).

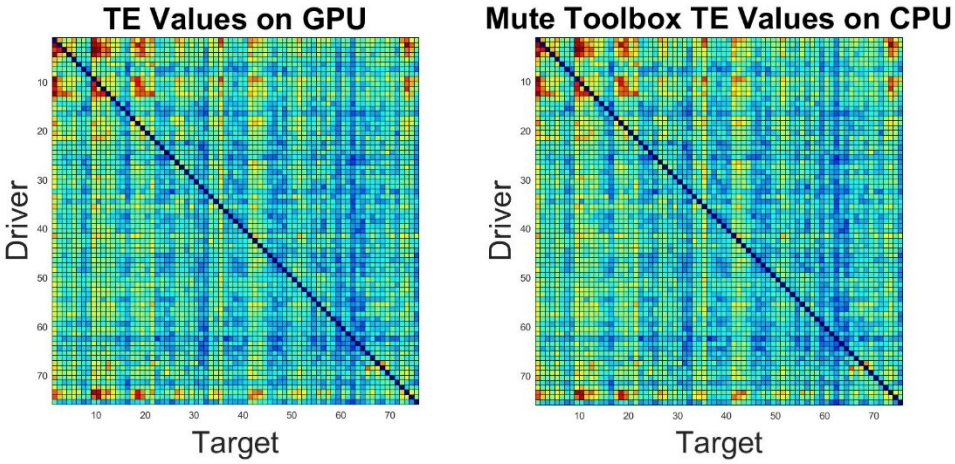

Figure 18

	Intel Core i7 -7500 CPU	384 cores - Nvidia Geforce 940MX GPU	Speed Up
Order one TE	56 sec	0.457 sec	123
Order two TE	183 sec	0.943 sec	194

Table – 4

Table 4 summarizes the timings of our algorithm on GPU compared to Mute toolbox on CPU.

CHAPTER 6

DISCUSSION

A. Test 1

For Test 1, it's clear from table 1 that there is a flow to be considered from electrode 1 to all following, from electrode 2 to all following, from electrode 3 to all following and so on as if the TE is unconditional although it's conditional. i.e. the similarity of flow from electrode 1 to electrodes 3,4,5,6,7,8 didn't cancel the flow from electrode 2 to electrodes 3,4,5,6,7,8 although it's a predecessor of electrode 2 and it is the main source of information flow. Similarly, electrode 1 flow didn't cancel all other flew from electrode 3 to the following electrodes and from electrode 4 to the following and so on so forth. This is due to the fact that when considering the conditional entropy electrode 1 is merged will others so that it loose its effect and it couldn't cancel other similar flow. However, it's clear from table 2 that there is only flow from electrode 1 to all remaining whereas the all other flows were neglected and have a very low value of optimum TE due to considering individual conditional entropy. Electrode 1 where there is high similarity in information flow and it's a predecessor of all remaining electrodes

succeeded to minimize all remaining optimum TE and succeeded to be considered the only original source of information flow which is logical. Moreover, in table 1 notice that the value of flow is low compared to table 2. For example TE(1->2) is 0.047 whereas it is 0.871 in table 2. This high difference is due to the fact that number of possible outcomes in table 1 is much higher than number of possible outcomes in table 2. So in table 2 the computation of probabilities is more precise and have better approximation of probabilities.

In case 1 (Table 1)

Possible Outcomes=PO=Q^{dN}

Where N is the total number of electrodes=8, d is the dimension of X_n^- =2 , Q=5 is the number of quantization levels=> PO1= 152587890625

In case 2 (Table 2)

PO2=Q^{3d} , d=2, Q=5 => PO= 15625

PO2 is much less than PO1.

B. Test 2

After applying real data as in [7] good accuracy is achieved by our proposed algorithm, with good timing, and better separation of SOZ from other electrodes. Although there is difference between our algorithm and that used by [7] there is

similarity in the final result where electrode 62 region seems to be SOZ in both. Fig. 9 shows results of [7]. Fig. 7 and Fig. 8 show results of Test 2.These results were taken by computing the ratio:

Notice that in Fig. 8 the separation between SOZ electrode and its nearest one is around 15 for a peak value of 25 as number of outflows. In this figure it's clear how much the separation between considering electrode 62 and all remaining is good enough to easily consider electrode 62 as SOZ. We may notice also that electrode 62 has completely suppressed its neighbors 60, 61, 63, and 64 although they have much more outflow as considered by [7]. They are not clearly separated in [7].

This test showed better accuracy compared to other experiments. The proposed algorithm increased the load too much on the GPU to achieve such accuracy (individual Z condition, shift of every Z in addition to the shift in Y with respect to X). However, due to parallel programming using CUDA the time of execution on Titan X-Pascal GPU was 1323 seconds = 22 minutes 3 seconds as indicated in table – 2. Which is acceptable for such application with such algorithm.

C. Test 3

Good accuracy is achieved, with good timing, but worse separation of SOZ from other electrodes. For a good separation, R should be >= (k * mean of R where k=2.4) which means that wrong outflow are more spread than Test 2 and we need more criteria to isolate them from true outflow. In other words wrong outflows are spread around the ratio mean by more than 2.4 times of this mean. Comparing Test 2 and Test 3 with same k = 1.25 as indicated in figures 8 and 13 respectively, it clear that Test 2 has better separation. Notice that the difference between both tests is that in Test 3 all remaining conditional electrodes (other than driver and target) are considered as one block. Notice from the figure that this algorithm didn't succeed to clearly suppress the neighbors of SOZ which is electrode 62.

Although the main advantage of our algorithm in Test 2 is that it's applied on GPUs the way Z condition is applied gave much improvement so that the original source of information suppresses all remaining.

Moreover notice that max value of TE in Test 2 is much higher than that of Test 3 because number of possible outcomes in Test 1 is much less than that of Test 2 as indicated in Table-3. Therefore, probabilities in Test 2 are more realistic due to the same reason mentioned in Test 1.

D. Test 4

Test 4 is the same as Test 2. It's applied to the same data. The only exception is that the maximum shift is 20 msec instead of 60 msec in both direction. The accuracy was bad with better timing. The better timing is due to the need of fewer shift steps. No separation of SOZ from other electrodes could be achieved. See Fig. 14, 15, and 16. This experiment indicates that information flow due to seizure between brain parts needs more than 20 msec to propagate and 60 msec as in Test 2 is good enough. This amount of time needed for propagation will indicates how many steps you need to shift depending on the sampling rate.

Max shift step = Fs * 60 /1000 where Fs is the sampling rate.

E. Test 5

Test 5 is conducted to compare between our proposed algorithm and Mute toolbox. Both algorithms used the same data. Fig.17 shows the TE values where the length of X_n^- or its order is equal to one. It's clear how much they are similar although there is a big difference in timing as shown in table 4. The experiment is repeated with

order = 2 and the result is shown in Fig.18. You may also notice the high similarities between two results.

Table-4 summarizes the timings of GPU and CPU. You may notice the high speed up of GPU. This speedup will increase as the order increase due the fact that in our GPU algorithm we compare all possible samples within the same signal and store every comparison in one byte. Then these bytes are allocated in cashed memory to acquire fast access. So during counting you have only to fetch only one byte meanwhile you need to fetch three bytes for order 3 TE implemented in a traditional way. Moreover, Mute computation is based on the calculation of entropies and many matrices are constructed for every TE. The size of these matrices is in general proportional to the square of order variable and it'll grow exponentially for conditional case. So a lot of memory access is needed compared to process time in the Mute toolbox. So this good speedup is achieved due to huge number of cores in the GPU compared to CPU and due to code optimization between our proposed algorithm and Mute toolbox algorithm.

CHAPTER 7

CONCLUSION

TE succeeded to prove itself a powerful tool in determining SOZ especially in its Binning form. The parallelized form of transfer entropy computations proposed on this thesis aimed to show the huge time savings attained in solving the problem specifically for large number of channels. Since our primary purpose was employability of highly parallelizable code that achieve load balance across thousands of cores, we proposed and implemented Triangular Transfer Entropy as a specific instantiation of TE computations. Under the assumptions of single SoZ foci, TTE seems to produce better results with better accuracy. The introduction of TTE provide a practical GPU-based tool where amount of time needed for 100-channel study is reduced from essentially prohibitive (order of many days) to feasible and in the order of tens of minutes. More general TE computations are expected to produce less savings due to less-than-optimal parallelizability of underlying computations. In all cases, more code optimization could be done for the computation of most inner loop (basic block of TE). Moreover, using multiple GPU cards will be a good choice and will guarantee more speed up. Increasing

speed up will allow improving accuracy of TE by increasing sampling rate, number of samples considered, or number of quantization levels within an acceptable time. Finally, and while our tests on benchmark data produced accurate SoZ localization, it is essential to study the performance of TTE for a wider database of IEEG recordings in order to identify the specific types of seizures where the proposed approach (single channels as focus) can accurately detect Seizure onset zones.

BIBLIOGRAPHY

1. I. Vlachos, B. Krishnan, D. M. Treiman, K. Tsakalis, D. Kugiumtzis and L. D. Iasemidis, "The Concept of Effective Inflow: Application to Interictal Localization of the Epileptogenic Focus From iEEG," in *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 9, pp. 2241-2252, Sept. 2017.
2. J. A. Adkinson, R. Liu, I. Vlachos and L. Iasemidis, "Connectivity Analysis for Epileptogenic Focus Localization," *2016 32nd Southern Biomedical Engineering Conference (SBEC)*, Shreveport, LA, 2016, pp. 3-4.
3. J. H. Cho, Y. J. Jung, H. C. Kang, H. D. Kim and C. H. Im, "Localization of ictal onset zones in Lennox-Gastaut syndrome using directed transfer function method," *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Boston, MA, 2011, pp. 5020-5023.
4. K. Lin, Y. Wang, K. Xu, J. Zhu, J. Zhang and X. Zheng, "Localizing seizure onset zone by convolutional transfer entropy from iEEG," *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Orlando, FL, 2016, pp. 6335-6338.
5. Schreiber T (2000) Measuring information transfer. Phys Rev Lett 85: 461.
6. M. Lindner, R. Vicente, V. Priesemann, M. Wibral (2011). TRENTOOL: a Matlab open source toolbox to analyse information flow in time series data with transfer entropy. *BMC Neurosci., 12*, p.119.
7. A. Montalto, “MuTE toolbox to evaluate multivariate transfer entropy,” figshare (2014), retrieved Sep 10, 2014; http: //dx.doi.org/10.6084/m9.figshare.1005245.

8. Jason Sanders , Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley Professional, 2010
9. David B. Kirk , Wen-mei W. Hwu, Programming Massively Parallel Processors: A Hands-on Approach, Morgan Kaufmann Publishers Inc., San Francisco, CA, 2010
10. "Epilepsy." *World Health Organization*, World Health Organization, Feb. 2017, www.who.int/mediacentre/factsheets/fs999/en/.