UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E COMPUTAÇÃO
CURSO DE ENGENHARIA ELÉTRICA – ÊNFASE EM SISTEMAS DE ENERGIA E AUTOMAÇÃO

DIOGO DA CRUZ GONÇALVES

Extração de características de ataques epilépticos utilizando base de dados de eletroencefalograma intracraniano

São Carlos
2017

DIOGO DA CRUZ GONÇALVES

Extração de características de ataques epilépticos utilizando base de dados de eletroencefalograma intracraniano

Monografia apresentada ao Curso de Engenharia Elétrica com ênfase em Sistemas de Energia e Automação, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Carlos Dias Maciel

São Carlos
2017

Gonçalves, Diogo da Cruz
G591e Extração de características de ataques epilépticos utilizando base de dados de eletroencefalograma intracraniano / Diogo da Cruz Gonçalves; orientador Carlos Dias Maciel. São Carlos, 2017.

Monografia (Graduação em Engenharia Elétrica com ênfase em Sistemas de Energia e Automação) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2017.

1. Eletroencefalograma. 2. Epilepsia. 3. iEEG. 4. Base de dados. 5. Estatatística básica. I. Título.

FOLHA DE APROVAÇÃO

Nome: Diogo da Cruz Gonçalves

Título: "Modelo de predição de ataques epilépticos utilizando base de dados de eletroencefalograma intracraniano"

Trabalho de Conclusão de Curso defendido e aprovado em 21 / 11 / 2017,

com NOTA 9,6 (nove, seis), **pela Comissão Julgadora:**

Prof. Associado Carlos Dias Maciel - Orientador - SEL/EESC/USP

Mestre Michel Bessani - Doutorando - SEL/EESC/USP

Mestre Tadeu Junior Gross - Doutorando - SEL/EESC/USP

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Rogério Andrade Flauzino

DEDICATÓRIA

Este trabalho é dedicado aos meus pais, e simboliza o fim de uma longa jornada que se iniciou quando eu ainda cursava o Ensino Médio e estava me preparando para ingressar em uma universidade. Apesar das dificuldades que enfrentamos à época, o apoio incondicional de vocês, com cada um contribuindo à sua maneira, me deu condições de chegar à universidade e concluí-la. Esse foi o melhor presente que um dia eu poderia ter sonhado em receber. Amo vocês.

AGRADECIMENTOS

Agradeço à minha família. Sem vocês, nada disso seria possível.

Agradeço muito ao Professor Carlos Maciel Dias e ao Daniel Rodrigues de Lima, ambos me acolheram em um momento difícil da minha graduação e deram todo suporte para que este trabalho fosse concluído. Sempre muito solícitos, sanaram as dezenas de dúvidas que tive ao longo do processo com a maior boa vontade. Daniel continuou me auxiliando mesmo depois de perder o vínculo com a universidade, não tenho palavras para expressar minha gratidão por dedicar tanto tempo me ensinando sobre análise de dados. Agradeço aos companheiros de laboratório, Michel Bessani, Jonas Rossi e Tadeu Gross por me ajudarem nos momentos de dificuldade. Este humilde trabalho também é de todos vocês.

Aos meus irmãos de república – Vitor Gineti, João Rebs, Ronyvon, José Eduardo Tetos, Henrique Trawitzki, Caio Sella Borne, Jack, Lucas Mala, Thales Xita, Luigi Bibre, Mello, Pedro Salam, Vinícius Craque, Samuel Laguesta, Matheus Curto, Gabriel Mouser, Gabriel Fimi, Ricardo Rufus e Ricardo Arafat – que moraram comigo por anos e proporcionaram a melhor experiência da minha vida, foi uma honra morar com vocês. Obrigado por tudo.

Às pessoas maravilhosas que encontrei ao longo da vida acadêmica – Paola Alarcon, Júlio Prates, Lucas Pardal e Ewerton Stanzani.

RESUMO

A epilepsia é doença que afeta quase 1% da população mundial, sendo caracterizada por ocorrências espontâneas. Para muitos pacientes, medicamentos anti-convulsão podem ser eficazes se ministrados em doses altas, entretanto, estes podem sofrer de efeitos colaterais. Para 20-40% dos pacientes, o uso de medicamentos não é eficaz. Mesmo com intervenção cirúrgica, há casos em que os ataques epilépticos continuam a ocorrer. Além disso, pacientes com epilepsia experimentam transtornos de ansiedade devido à chance de ocorrer um ataque. Sistemas que predizem ataques epilépticos podem ser de grande ajuda para essas pessoas pois melhorarão a qualidade de vida destas. Com base no sinal de eletroencefalograma (EEG), predições podem ser desenvolvidas, algoritmos computacionais podem identificar períodos de maior probabilidade de ocorrência de ataque epiléptico de maneira confiável. Desse modo, equipamentos podem ser desenvolvidos para alertar pacientes de um ataque e assim evitar atividades perigosas tais como dirigir e nadar, e medicamentos seriam usados somente quando necessário, reduzindo efeitos colaterais.

ABSTRACT

Epilepsy afflicts nearly 1% of the world's population, and is characterized by the occurrence of spontaneous seizures. For many patients, anticonvulsant medications can be given at sufficiently high doses to prevent seizures, but patients frequently suffer side effects. For 20-40% of patients with epilepsy, medications are not effective. Even after surgical removal of epilepsy, many patients continue to experience spontaneous seizures. Despite the fact that seizures occur infrequently, patients with epilepsy experience persistent anxiety due to the possibility of a seizure occurring. Seizure forecasting systems have the potential to help patients with epilepsy lead more normal lives. In order for electrical brain activity (EEG) based seizure forecasting systems to work effectively, computational algorithms must reliably identify periods of increased probability of seizure occurrence. If these seizure-permissive brain states can be identified, devices designed to warn patients of impeding seizures would be possible. Patients could avoid potentially dangerous activities like driving or swimming, and medications could be administered only when needed to prevent impending seizures, reducing overall side effects.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A epilepsia é uma doença comum e grave que é caracterizada por ataques recorrentes e afetam mais de 60 milhões de pessoas no mundo todo. Entre 30 a 40% dos casos não são adequadamente controlados com os tratamentos disponíveis. Apesar de cirurgias serem eficazes em alguns casos, muitas vezes não são apropriadas para a maioria dos pacientes (Cook et al., 2013).

A aparente natureza aleatória dos ataques é um fator significativo na qualidade de vida dos pacientes com epilepsia (Fisher, 2000; Schulze-Bonhage and Kuhn, 2008). Apesar de fazerem uso diário de medicamentos, muitos pacientes com epilepsia continuam apresentando ataques (Kwan et al., 2010; Kwan and Brodie, 2010). Predições eficazes dos ataques transformariam positivamente o tratamento de epilepsia permitindo que pacientes alterem suas atividades cotidianas a fim de evitar riscos e façam uso de medicamentos apenas quando for necessário.

Entretanto, para que tais predições sejam clinicamente relevantes, é de grande importância que se aprimore os métodos que identificam os períodos nos quais os ataques são mais prováveis de ocorrerem (Cook et al., 2013). Evidências significativas sustentam a ideia de que os ataques surgem a partir de um padrão característico nas ondas cerebrais (Stacey et al., 2011; Cook et al., 2013). Estudos clínicos comprovam mudanças no fluxo de sangue cerebral, oxigenação e excitação na área cortical do cérebro horas ou minutos antes de um ataque (Baumgartner et al., 1998; Adelson et al., 1999; Aarabi et al., 2008; Badawy et al., 2009).

Na busca de se obter predições cada vez mais assertivas, estudos aplicaram aprendizado de máquinas (*machine learning*) para predição dos ataques com resultados promissores (Mirowski et al., 2009; Park et al., 2011; Howbert et al., 2014). Porém, a escassez de dados de longa duração permaneceu um obstáculo para se alcançar resultados satisfatórios, assim como o quase inexistente confrontamento de algoritmos desenvolvidos por diferentes grupos de pesquisadores usando a mesma base de dados.

Recentemente, um dispositivo cirurgicamente implantado desenvolvido pela empresa NeuroVista® Inc., em inglês NeuroVista Seizure Advisory System, tornou possível a obtenção do sinal do eletroencefalograma intracraniano (iEEG) de um

paciente, em tempo real, enquanto um algoritmo de predição realizava o tratamento deste sinal.

Inicialmente, o dispositivo foi validado em cães (Davis et al., 2011; Coles et al., 2013; Howbert et al., 2014). A epilepsia canina é bastante útil para se estudar a epilepsia humana (Leppik et al., 2011; Patterson, 2014) visto que ambas são clinicamente (Potschka et al., 2013; Packer et al., 2014) e neurofisiologicamente semelhantes (Berendt et al., 1999; Berendt and Dam, 2003; Pellegrino and Sica, 2004). Em um estudo referência na área, NeuroVista® e pesquisadores australianos implantaram este dispositivo em 15 pacientes resistentes à drogas que previnem a epilepsia, e alcançaram de 65-100% de eficácia na predição de ataques de 11 pacientes durante o período de teste dos algoritmos, e em 8 pacientes nos quatro meses seguintes (Cook et al., 2013).

Apesar dos notáveis avanços obtidos nos últimos anos, melhorias na eficácia dos algoritmos de predição são necessários para consolidar o uso clínico dos dispositivos.

1.1 Descrição

No ano de 2016, o site Kaggle promoveu uma competição intitulada *Predict Seizures in long-term human intracranial EEG recordings,* no qual disponibilizou a base de dados obtido pelo dispositivo da NeuroVista®. Como dito acima, o modelo de predição utilizado obteve resultado bastante expressivo em 8 dos 15 pacientes, dos 7 pacientes restantes em que o modelo de predição não apresentou bons resultados, 3 destes tiveram seu monitoramento de iEEG disponibilizado na plataforma do site para que os competidores pudessem criar seus próprios algoritmos a fim de contribuírem para um modelo de predição mais assertivo.

1.2 Objetivo Deste Trabalho

A base de dados utilizada nesta monografia foi a mesma disponibilizada no site Kaggle. Assim como na competição, o objetivo deste trabalho é analisar e tentar distinguir os sinais de iEEG em dois períodos distintos, na iminência de um ataque de epilepsia e horas antes de um ataque, também denominados períodos pré e inter-ataques, respectivamente. Para tanto, serão utilizados os conceitos teóricos

abordados na disciplina de Processamento Digital de Sinais do curso de Engenharia Elétrica da EESC-USP.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será abordado os métodos estatísticos e ferramentas matemáticas que serão utilizados para se analisar a base de dados obtida.

2.1 Estatística Básica

Em estatística, quando se deseja resumir uma grande quantidade de dados em valores que sejam representativos de toda a série, lança-se mão de uma das seguintes medidas de posição (ou localização) central: média, mediana ou moda.

A moda representa a realização que ocorre mais frequentemente dentre os valores observados. Por exemplo, considerando todos os sorteios de números da MegaSena, aquele número que foi sorteado com maior frequência em um dado período representa a moda deste espaço amostral. Em alguns casos, pode haver mais de uma moda, ou seja, a distribuição dos valores pode ser bimodal, trimodal e assim por diante.

A mediana representa o valor que ocupa posição central da série de observações quando ordenada em ordem crescente. Assim, se as sete observações forem os valores 1, 3, 4, 7, 8, 8 e 9, a mediana tem valor 7, correspondendo à quarta observação. Caso o número de observações seja par, usa-se como mediana a média aritmética das duas observações centrais.

A média aritmética, conceito bastante utilizado em situações do cotidiano, é a soma da observações dividida pelo número delas. A fim de formalizar o conceito de média, se x_1,..., x_n são os n valores (distintos ou não) da variável X, a média c

$$x_m = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{x_1 + \cdots + x_n}{n} \qquad (1)$$

Para o caso de *n* observações da variável X, no qual n_1 são iguais a x_1, n_2 são iguais a x_2 e assim por diante, a média de X pode também ser escrita por

$$x_m = \frac{n_1 x_1 + \cdots + n_k x_k}{n} = \frac{1}{n}\sum_{i=1}^{n} n_i x_i \qquad (2)$$

Analisando dados somente com medidas de posição pode esconder informações importantes sobre a variabilidade do conjunto de observações. Por exemplo, para cinco grupos de alunos que obtiveram as seguintes notas:

Grupo A (variável X): 3, 4, 5, 6, 7
Grupo B (variável Y): 1, 3, 5, 7, 9
Grupo C (variável Z): 5, 5, 5, 5, 5
Grupo D (variável W): 3, 5, 5, 7
Grupo E (variável V): 3, 5, 5, 5, 6, 6

Vemos que a média de notas de todos os grupos acima é igual a 5. E este valor nada nos informa a respeito de suas diferentes variabilidades. Desse modo, surge a necessidade de se analisar os dados segundo parâmetros que sumarizem a variabilidade de um conjunto de observações. Um critério comumente utilizado para suprimir tal necessidade é utilizar um parâmetro que mede a dispersão em torno de sua média, e as duas medidas mais usadas são o desvio médio e a variância.

Para o Grupo A, os desvios $x_i - x_m$ são: -2,-1,0,1,2. É fácil de se verificar que a soma dos desvios é igual a zero para qualquer conjunto de dados. Portanto, a soma dos desvios não é uma boa medida de dispersão para o conjunto A. É mais conveniente exprimir as medidas como médias, desvios absolutos ou quadráticos (variância):

$$dm(X) = \sum_{i=1}^{n} \frac{|x_i - x_m|}{n} \qquad (3)$$

$$var(X) = \sum_{i=1}^{n} \frac{(x_i - x_m)^2}{n} \qquad (4)$$

respectivamente.

Para o Grupo A, temos

$$dm(X) = \frac{6}{5} = 1.2$$

$$var(X) = \frac{10}{5} = 2.0$$

Enquanto para o Grupo D,

$$dm(W) = \frac{4}{4} = 1.0$$

$$var(W) = \frac{8}{4} = 2.0$$

Conclui-se que, para o desvio médio, o grupo D é o mais homogêneo que o Grupo A. enquanto ambos são igualmente homogêneos do ponto de vista da variância. (BUSSAB, 2010)

Já que a variância tem medida igual aos quadrados da dimensão dos dados (por exemplo, se os dados estão expressos em cm, a variância será expressa em cm ao quadrado), e isso pode levar a problemas de interpretação. E por conta disso, costuma-se usar o desvio padrão, que é definido como a raiz quadrada de valor positiva do valor da variância. Para o Grupo A, o desvio padrão é

$$dp(X) = \sqrt{var(X)} = \sqrt{2} = 1.41$$

Tanto o desvio médio quanto o desvio padrão indicam qual o “erro” em relação à média do conjunto de dados. (BUSSAB, 2010)

2.2 Transformada de Fourier – Análise Espectral

O eletroencefalograma (EEG) pode ser considerado um sinal neural básico em engenharia biomédica (DEUTSCH, 1993) e na medicina (ZSCHOCKE, 1993), por se tratar de um exame simples, comum, de custo reduzido e não invasivo quando comparado com os procedimentos que envolvem neuroimagens (ANDRÄ, 2007), tais procedimentos garantem elevado precisão diagnóstica, possuem custos não condizentes com a realidade hospitalar brasileira.

Apesar das vantagens do EEG citadas acima, este também possui limitações (NIEDERMEYER, 2004), a saber: 1) baixa resolução – haja vista que o sinal gerado é resultado da ação conjunta de neurônios corticais, o que impede a obtenção de informação sobre o estado fisiológico de áreas cerebrais mais profundas; 2) baixa amplitude, sofrendo alterações relevantes na presença de ruídos; e interpretação controversa ou pouco conclusiva quando comparado à exames diagnósticos de neuroimagem.

Na engenharia biomédica, sinais de EEG são essenciais para aplicações relacionadas à interface cérebro-máquina, a terapias de reabilitação (próteses) e *biofeedback* (WOLPAW, 2002).

Na medicina, o EEG constitui no principal exame diagnóstico das epilepsias (NIEDERMEYER, 2004); de diversos tipos de demências, incluindo a doença de Alzheimer. No campo da biologia, especificamente da neurociência computacional (KANDEL, 2000), o EEG é utilizado para exames fisiológicos e cognitivos em animais, principalmente à investigação dos efeitos de eletroestimulação funcional (WOLPAW, 2002).

Sendo o EEG um sinal aleatório no domínio do tempo, a abordagem é focada nas transformadas espectrais (AKAY,1997) visto que a análise no domínio da frequência fornece diversas informações importante sobre a natureza do sinal, tais como os diversos ritmos cerebrais (ondas alfa, beta, teta, gama, delta). Desta forma, os profissionais e pesquisadores da área de exatas tendem a privilegiar este tipo de abordagem.

Em contrapartida, os profissionais da área da saúde estão habituados para analisar o EEG tão somente no domínio do tempo, com base na inspeção visual das formas de onda. Tal análise dá maior importância aos aspectos de amplitude e

forma (morfologia), sendo também possível, dependendo da experiência de quem analisa o sinal, extrair informações à respeito também dos ritmos cerebrais.

Nesse sentido, há um descompasso entre a análise puramente visual feita pelos médicos e a análise espectral feita por profissionais da área de exatas, o que dificulta o aprofundamento da compreensão do EEG por ambas as partes. Sabendo que a análise espectral fornece informações essenciais acerca do sinal estudado, a análise meramente visual amplamente utilizada em clínicas de neurofisiologia é insuficiente para um estudo assertivo do problema exposto neste trabalho.

Portanto, o sinal de EEG será submetido ao modelamento matemático conhecido como transformada de Fourier.

Como exposto anteriormente, sinais de EEG são aleatórios no tempo e por esta característica não podem ser descritos como uma função matemática explícita. Para estuda-los é necessário introduzir uma descrição probabilística, ou seja, é necessário considerar todas" histórias temporais" prováveis do sinal.

Para que a análise de Fourier seja aplicável o sinal deve ser periódico no tempo, mas sabemos que sinais aleatórios não apresentam esta característica. Deste modo, lança-se mão de uma aproximação que considera cada fragmento do sinal de EEG periódico no tempo, tornando assim possível considerar o EEG como uma soma de funções periódicas (LATHI, 1987). Para melhor explicitar como a análise de Fourier é realizada, alguns conceitos importantes serão discutidos a seguir

A análise do sinal de EEG no domínio do tempo considera a amplitude como um dos parâmetros mais relevantes e ela pode ser medida de várias maneiras, como por exemplo, amplitude de pico, amplitude pico a pico, amplitude média, valor RMS (valor quadrático médio)

Definindo-se um sinal qualquer $x(t)$ no tempo, a função *X(f)* é a transformada direta de Fourier de *x(t)*, e representa as amplitudes das várias componentes de frequência que constituem o sinal. Portanto, X(f) pode ser interpretada como o grau de participação de cada componente de frequência que compõe x(t), tal fato pode ser mais claramente observado na equação (5) abaixo:

$$X(f) = \int_{-\infty}^{\infty} x(t) \exp(-2j\pi f t)\, dt \qquad (5)$$

No domínio do tempo, para cada instante de tempo t há um valor correspondente x(t) para o sinal, ao passo que no domínio da frequência, X(f) nos informa à respeito das amplitudes para cada componente de frequência do sinal.

A representação gráfica de X(f) pode ser bastante complexa em certos casos, para um melhor entendimento do espectro do sinal, usualmente decompõe-se X(f) em espectro de amplitude e espectro de fase indicados por |X(f)| e θ(f), respectivamente. (Macedo, 2014)

A transformada de Fourier de x(t) é uma função X(f) cuja imagem está no conjunto dos números complexos, logo ela pode ser decomposta em suas partes real e imaginária, mas também pode ser escrita em sua forma polar. Tomaremos $X_r(f)$ e $X_i(f)$ como as partes reais e imaginárias de $X(f)$, respectivamente, e $j = \sqrt{-1}$. Assim, $X(f) = X_r(f) + jX_i(f) = |X(f)|exp(j\theta(x))$

A amplitude da transformada de Fourier X(f) = Xr(f) + jXi(f), ou espectro de amplitude, é definido por $|X(f)| = \sqrt{Xr(f)^2 + Xi(f)^2}$, enquanto o ângulo de fase, ou espectro de fase, é definido por θ(x) = arctan(Xi(f)/Xr(f)).

Outro parâmetro importante na análise de Fourier é o espectro de potência, definido por $P(f) = |X(f)|^2$.

2.3 Transformada de Fourier: tempo discreto

A transformada de Fourier discreta é a transformada de Fourier para sinais em tempo discreto. Neste caso, tem-se que a representação espectral do sinal amostra é composto pelo espectro do sinal com banda limitada e deslocado de kF_s , onde $F_s = 1/T$ conhecido como frequência de amostragem.

O par de transformada de Fourier de um sinal x[n] é definido por:

$$x[n] = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(e^{jw})e^{iwn}\, dw \qquad (6)$$

$$X\left(e^{jw}\right) = \sum_{n=-\infty}^{\infty} x[n]e^{-jwn} \qquad (7)$$

A condição de existência da transformada de Fourier é: $X\left(e^{jw}\right) < \infty$

2.4 O algoritmo FFT – Fast Fourier Transform

O algoritmo proposto por Cooley e Tukey (1966) explora as propriedades de simetria da DFT e diminui sensivelmente o número de operações realizadas.

A DFT é dada por

$$x[k] = \sum_{k=0}^{N-1} x[n] W_N^{kn} \qquad (8)$$

k = 0,1,...,N-1 onde $W_n = e^{-j2\pi/N}$

Reescrevendo a DFT por seus termos pares e ímpares

$$X[k] = \sum_{n_0=0}^{(N-1)/2} x[2n_0] W_K^{(2n_0)k} + \sum_{n_1=0}^{(N-1)/2} x[2n_1+1] W_K^{(2n_1+1)k} \qquad (9)$$

2.5 Densidade espectral de energia ou potência

A densidade espectral de energia ou potência é matematicamente descrita por :

$$E_x = \sum_{n=-\infty}^{\infty} |x[n]|^2 = \sum_{n=-\infty}^{\infty} x[n]x^*[n]$$

$$= \sum_{n=-\infty}^{\infty} x[n]\{\frac{1}{2\pi}\int_{-\pi}^{\pi} X^*(e^{jw})e^{-jw}\ dw\}$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|X(e^{jw})\right|^2 dw \qquad (10)$$

A quantidade $|X(e^{jw})^2 = S_x(f)$ é conhecida como densidade espectral de energia, Quando o sinal possui potência média a justificativa fica equivalente embora $S_x(f)$ fica conhecida como densidade espectral de potência média.

3 MATERIAIS E MÉTODOS

Este capítulo descreve todos os passos realizados para se obter a análise da base de dados.

3.1 Ambiente virtual do Python 3.5.3 – Spyder

Anaconda® é uma plataforma de código aberto voltada aos cientistas de dados e hospeda diversos ambientes virtuais para análise de dados, contendo cerca de 720 bibliotecas também de código aberto que podem ser facilmente instaladas caso seja necessidade do usuário. Dentre estes ambientes virtuais, há alguns com aplicações bastante interessantes, à saber:

- Jupyter Notebook – São páginas visíveis em um browser (Chrome, Firefox, Internet Explorer, entre outros) que permite misturar textos, códigos executáveis em Python, imagens e figuras. Este tipo de página é um recurso que viabiliza o estudo interativo, no qual o usuário pode executar e modificar o código e ver os resultados sem sair da página em que está programando. O nome *Jupyter* advém das três linguagens de programação suportadas pela aplicação – Julia, Python e R.
- PyQT – Ambiente que cria uma ponte entre o Python e a biblioteca QT (framework multiplataforma para desenvolvimento de interfaces gráficas criado pela norueguesa Trolltech. Com ele é possível desenvolver aplicativos e bibliotecas uma única vez e compila-los para diversas plataformas sem que haja necessidade de alterar o código fonte.
- Spyder – É um Ambiente de Desenvolvimento Integrado (IDE – na sigla em inglês) para Python que contém modos avançados de edição, testes interativos, depuração e recursos de introspecção. Além disso, por possuir suporte do IPython, a instalação de diversos módulos científicos tais como Pandas, NumPy, Matplotlib e SciPy é fácil e acrescenta funcionalidades poderosas para se trabalhar com análise de dados.
- Rstudio – Software livre, IDE para linguagem de programação em R, bastante utilizado para estudos gráficos e cálculos estatísticos.

O ambiente escolhido para trabalhar com a base de dados em questão foi o Spyder, visto que a fácil manipulação e acesso às bibliotecas Pandas, NumPy, Matplotlib conferem a instrumentação necessária para realizar a análise estatística que está no escopo deste trabalho.

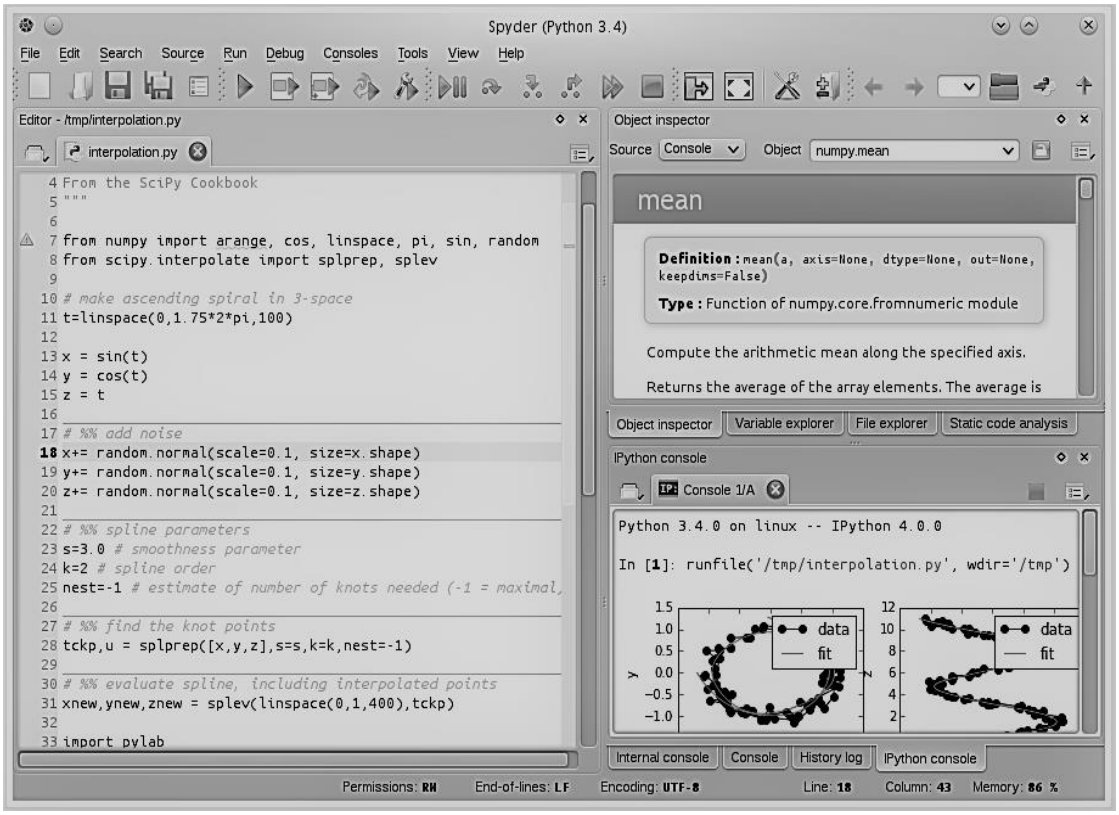

Figura 1- Ambiente virtual Spyder

Cada uma destas bibliotecas tem funcionalidades bastante interessantes e serão exploradas a seguir:

3.2 Bibliotecas Utilizadas

- Pandas – Garante manipulação fácil e flexível de estruturas de dados. Adequa-se bem à diferentes tipos de dados como por exemplo: dados tabulares heterogêneos em Excel ou SQL, dados de séries temporais ordenados ou não, matrizes de dados heterogêneas. Os dois principais tipos de estruturas de dados no Pandas, *series* e *dataframes*, possibilitam uma ampla aplicação em finanças, estatística, ciências sociais e diversas áreas da engenharia. Além disso, as facilidades em se trabalhar com estas estruturas do Pandas incluem dividir, indexar, manipular e transformar grande estruturas de dados de forma inteligente e fácil conversão de outros formatos para formato *dataframe*.
- NumPy – Biblioteca básica da linguagem Python, que viabiliza o trabalho com arranjos, vetores, matrizes com N dimensões. Possui sintaxe semelhante ao do *software Matlab*. Possui funções e operações sofisticadas tais como: objeto array para implementação de arranjos multidimensionais, objeto *matrix* para cálculo com matrizes, ferramentas para álgebra linear, Transformada de Fourier básicas e geração de números aleatórios.
- Matplotlib – Biblioteca com recursos para geração de gráficos 2D a partir de arrays, abrange uma séria de possibilidades gráficas, como gráficos de barra, linha, histogramas, entre muitos outros.
- SciPy – Biblioteca que possui módulos para estatística, otimização de sistemas, integração, álgebra linear, Transformada de Fourier, processamento digital de sinais entre outros.

Outras bibliotecas também serão utilizadas para funções mais básicas como importar arquivos, por exemplo. As mais importantes e relevantes para este trabalho foram discutidas acima.

3.3 Fluxograma

Uma vez familiarizados com ambiente virtual adotado para a realização deste trabalho, segue o fluxograma dos etapas a serem feitas.

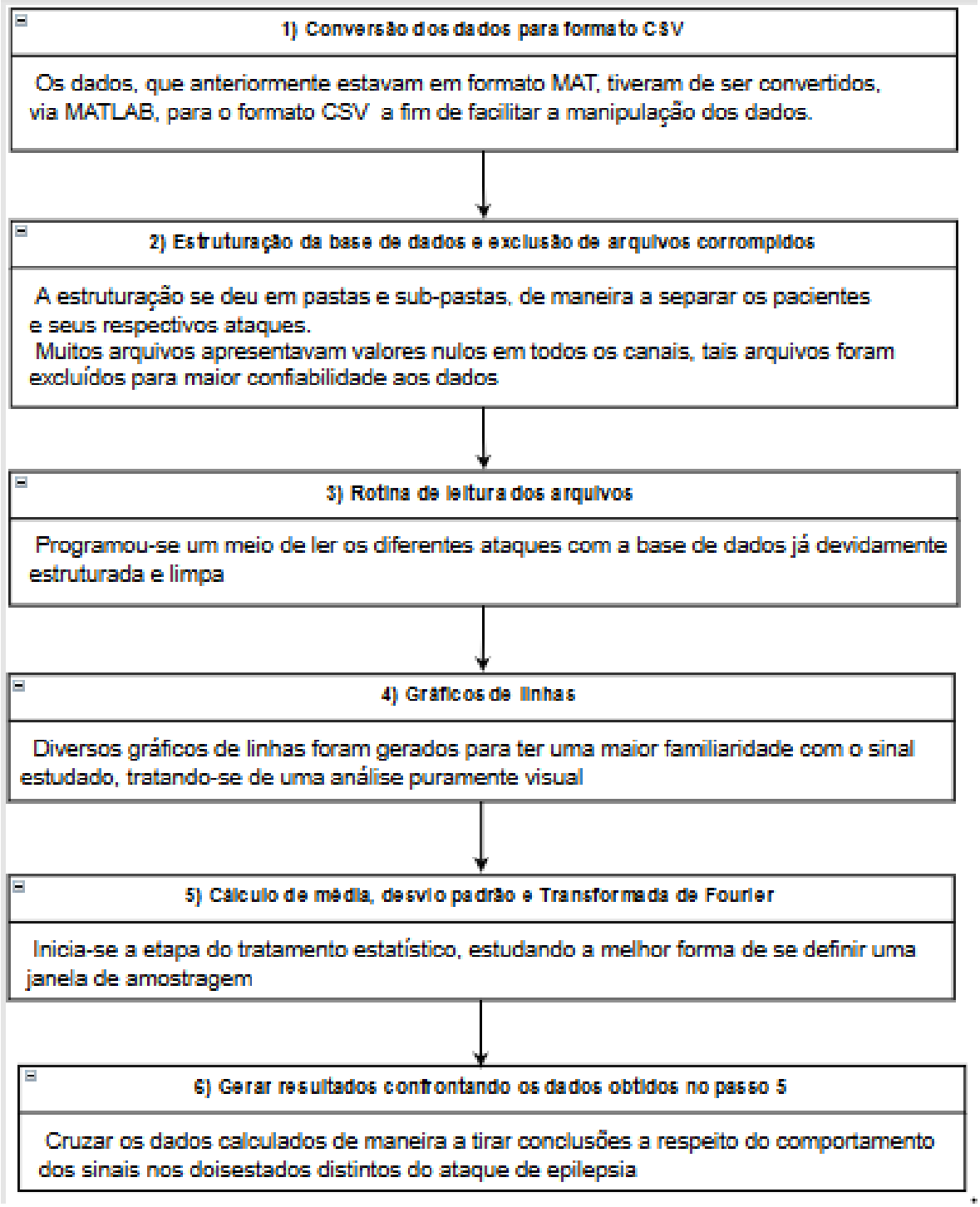

Figura 2 - Passo a passo das tarefas realizadas no trabalho

3.4 Descrição da Base de Dados

A base de dados de eletroencefalograma intracraniano (iEEG) está organizada em pastas contendo dados de três pacientes. Os dados estão dispostos em arquivos de 10 minutos e foram nomeados de maneira a discriminar períodos pré e inter-ataques. Toda esta base de dados foi obtida originalmente em arquivos .mat com a informações a seguir:

- I_J_K.mat - o J-ésimo segmento de dados correspondente K-ésima classe (K=0 para pré-ataque, K=1 para inter-ataque) para o I-ésimo pacientes (três pacientes no total).

  Cada arquivo .mat contém um *struct* de dados, chamada de *dataStruct,* que possui os seguintes campos:

  1) *data*: uma matriz com valores das amostras de iEEG dispostas em número da amostra x eletrodo.
  2) *nSamplesSegment*: número de amostras.
  3) *iEEGsamplingRate*: taxa de amostragem do iEEG.
  4) *channelIndices*: um *array* com os índices dos eletrodos correspondendo às colunas no campo *data.*
  5) *sequence*: O índice que informa em qual período de 10 minutos dentro de uma hora antes de um ataque o arquivo foi obtido. Por exemplo, 1_12_1.mat tem sequência número 6 e assim representa o iEEG do intervalo de 50 a 60 minutos.

Assim, ao agrupar seis arquivos sequenciais de pré-ataque temos o registro de uma hora. Por exemplo, os arquivos de 1_1_0 ao 1_6_0 agrupados dizem respeito à um certo pré-ataque do “Paciente 1”, os arquivos de 1_7_0 ao 1_12_0 dizem respeito a outro pré-ataque do mesmo paciente. O mesmo vale para os arquivos inter-ataques.

Importante salientar que foi verificado que, em todos arquivos da base de dados, a frequência de amostragem é de 400 Hz e o número de amostras igual à 240 mil, confirmando a informação que cada arquivo *data* possui um trecho de 10 minutos de monitoramento de iEEG.

Os arquivos pré-ataque são fornecidos cobrindo necessariamente uma hora antes do ataque com uma margem de 5 minutos (ou seja, de 1:05 a 0:05 minuto

antes da ocorrência do ataque). Esta margem garante que os ataques possam ser identificados em tempo hábil para que os pacientes façam uso de medicamentos de ação rápida.

Os dados de inter-ataques foram escolhidos aleatoriamente a partir do registro completo, com a restrição que os segmentos referentes a este período foram extraídos, no máximo, quatro horas antes da ocorrência de um ataque. Isso evita a contaminação de dados com os sinais pré-ataque e inter-ataque.

3.5 Conversão dos Dados do Formato MAT para CSV

A leitura dos arquivos em formato MAT apresentou-se problemática no ambiente interativo do Python. Por conta disto, foi necessário converter a base de dados para o formato CSV facilitando a manipulação dos dados para se realizar as análises posteriores. Apenas os dados de iEEG foram convertidos, visto que os outros dados, tais como taxa de conversão, número de amostras de índices dos canais do eletrodo são os mesmos para todos arquivos. Tal conversão foi feita através do *software* Matlab 7.14 (R2012b). Somente a conversão dos dados foi realizada pelo software Matlab, todas as etapas seguintes foram realizadas no ambiente interativo do Python chamado *Spyder*.

3.6 Estruturação dos Dados e Exclusão de Arquivos Corrompidos

Ao converter os arquivos para CSV, tanto os arquivos de pré-ataque quanto inter-ataque estavam misturados na mesma pasta. Para facilitar a leitura de tais arquivos, as duas classes foram separadas em sub-pastas distintas. Assim, a organização dos dados ficou bastante simples: a pasta principal recebeu o nome “Paciente_1”, “Paciente_2” e “Paciente_3” e cada pasta principal continha duas sub-pastas denominadas “Pre_ataques” e “Inter_ataques”. E cada sub-pasta, por sua vez, continha os ataques agrupados em seis arquivos para designar uma hora de monitoramento de iEEG.

A figura a seguir mostra de forma simplificada como se deu a estruturação dos dados para o “Paciente_1”. A mesma estrutura se aplica para os outros dois pacientes.

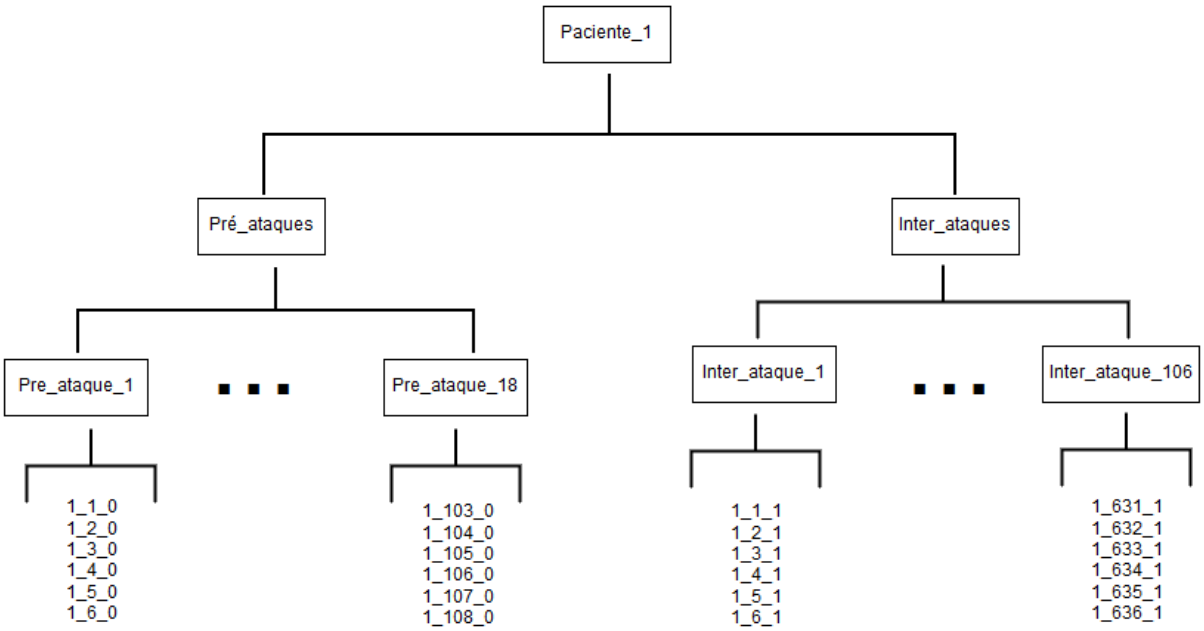

Figura 3 - Estruturação da base de dados.

Durante o monitoramento do sinal cerebral, ocasionalmente ocorria um mau funcionamento do eletrodo que captava este sinal, e esta falha pode ser verificada nos arquivos que contém valores nulos para todos os 16 canais. Alguns arquivos estão integralmente corrompidos (10 minutos de monitoramento com valor zero em todos canais) e alguns estão parcialmente corrompidos (entre 0 e 10 minutos com valor zero em todos canais). Para uma maior confiabilidade na análise dos dados, todos arquivos que continham pelo menos 5 minutos sem registro do sinal cerebral foram excluídos, assim como a pasta em que este arquivo estava inserido. A exemplo da Figura 3, se um ou mais arquivos da sub-pasta “Pre_ataque_18” apresentasse tal falha, esta subpasta seria excluída da análise.

Tanto a estruturação quanto a exclusão das pastas que continham os arquivos corrompidos foram realizadas com algoritmos em Python utilizando as bibliotecas “os” para varrer os arquivos, “shutil” para criar as pastas de cada pré e inter-ataque. Este algoritmo encontra-se na seção de Apêndice 1.

É válido ressaltar que o maior período de monitoramento de iEEG que é possível se remontar é de uma hora, e por conta disso, os arquivos foram agrupados dessa forma para que a comparação de pré e inter-ataques em um período fixo de tempo possa ser realizada. Outra ressalva importante é que dados não estão dispostos em ordem cronológica. Por exemplo, o “Pre_ataque_1” não necessariamente ocorreu

antes do “Pre_ataque_2”, da mesma forma que o “Inter_ataque_3” não necessariamente ocorreu antes do “Inter_ataque_4”.

Após a etapa inicial de estruturação e exclusão dos arquivos corrompidos, a base de dados apresentou um volume total de 76GB e o número total de pré e inter-ataques disponíveis para análise é apresentado na seguinte tabela:

	Paciente_1	**Paciente_2**	**Paciente_3**
Pré_ataques	18	23	24
Inter_ataques	106	217	341

Tabela 1- Pré e inter-ataques disponíveis por paciente após organização de exclusão de arquivos corrompidos.

3.7 Leitura dos Arquivos

A próxima etapa do trabalho foi criar uma rotina que facilitasse a leitura dos arquivos de pré e inter-ataques que estão no formato CSV, uma vez que estão organizados em pastas, esta tarefa se deu de forma simples. Resumidamente, cada arquivo era lido utilizando a biblioteca Pandas e carregado em um *DataFrame*, e cada grupo de 6 arquivos era sequencialmente concatenado para formar um ataque, e após essa concatenação de arquivos, tal ataque era adicionado a uma lista que, ao término da leitura de todos arquivos, continha todos pré ou inter-ataques de determinado paciente. O algoritmo responsável pela leitura dos arquivos está na seção de Apêndice 1.

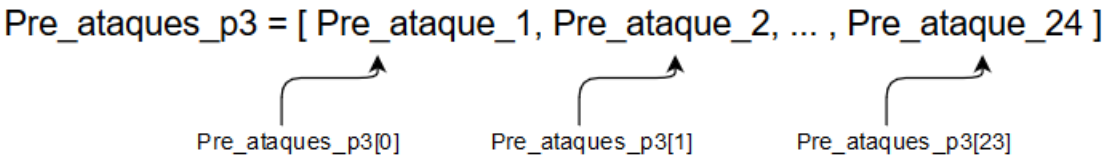

Figura 4 - Exemplo de leitura de arquivos agrupando todos pré-ataques do Paciente 3 em uma única lista.

3.8 Gráficos de Linha

Uma vez concluída a etapa de leitura de todos ataques dos três pacientes, deu-se início ao processo de inspeção e análise visual dos pré e inter-ataques. Os gráficos de linhas são importantes ferramentas para identificar características e padrões nos sinais estudados. Como citado na seção 2.2., esta análise é a mais utilizada pelos médicos e aqueles que possuem a devida experiência conseguem extrair informações valiosas do sinal de iEEG no domínio do tempo.

Contudo, quando se trata de uma base de dados extensa, como é o caso deste trabalho que possui centenas de gráficos de linha para serem analisados, o método de análise puramente visual se torna inviável para determinar algum padrão presente em todos os sinais. Em virtude disso, serão apresentados um gráfico de linha de pré-ataque e um gráfico de linha de inter-ataque que são considerados representativos destas duas diferentes fases e suas características serão discutidas mais adiante.

Importante citar que não é fornecida nenhuma informação a respeito da ordem de grandeza de iEEG coletado neste trabalho (se é Volts, mV ou µV). Apesar de em alguns artigos científicos da área citarem que este sinal é coletado em mV, não é possível afirmar que o sinal estudado neste trabalho também foi coletado em mV.

Lembrando que cada ataque possui uma hora de duração, e nesta uma hora de duração há um total de 1,44 milhão de amostras, foi necessário diminuir o número de amostras na plotagem para que ficasse visualmente agradável. Esse processo que constitui na diminuição do número de amostras é conhecido como decimação (em inglês, *downsampling*). A decimação utilizada para se gerar as figuras tem um fator de cem, ou seja, uma a cada cem amostras do total de 1,44 milhão foram escolhidas, reduzindo o número de amostras para 14.400. A biblioteca do *matplotlib* e suas funcionalidades se mostraram excelentes para todas as figuras geradas neste trabalho. As plotagens são apresentadas a seguir.

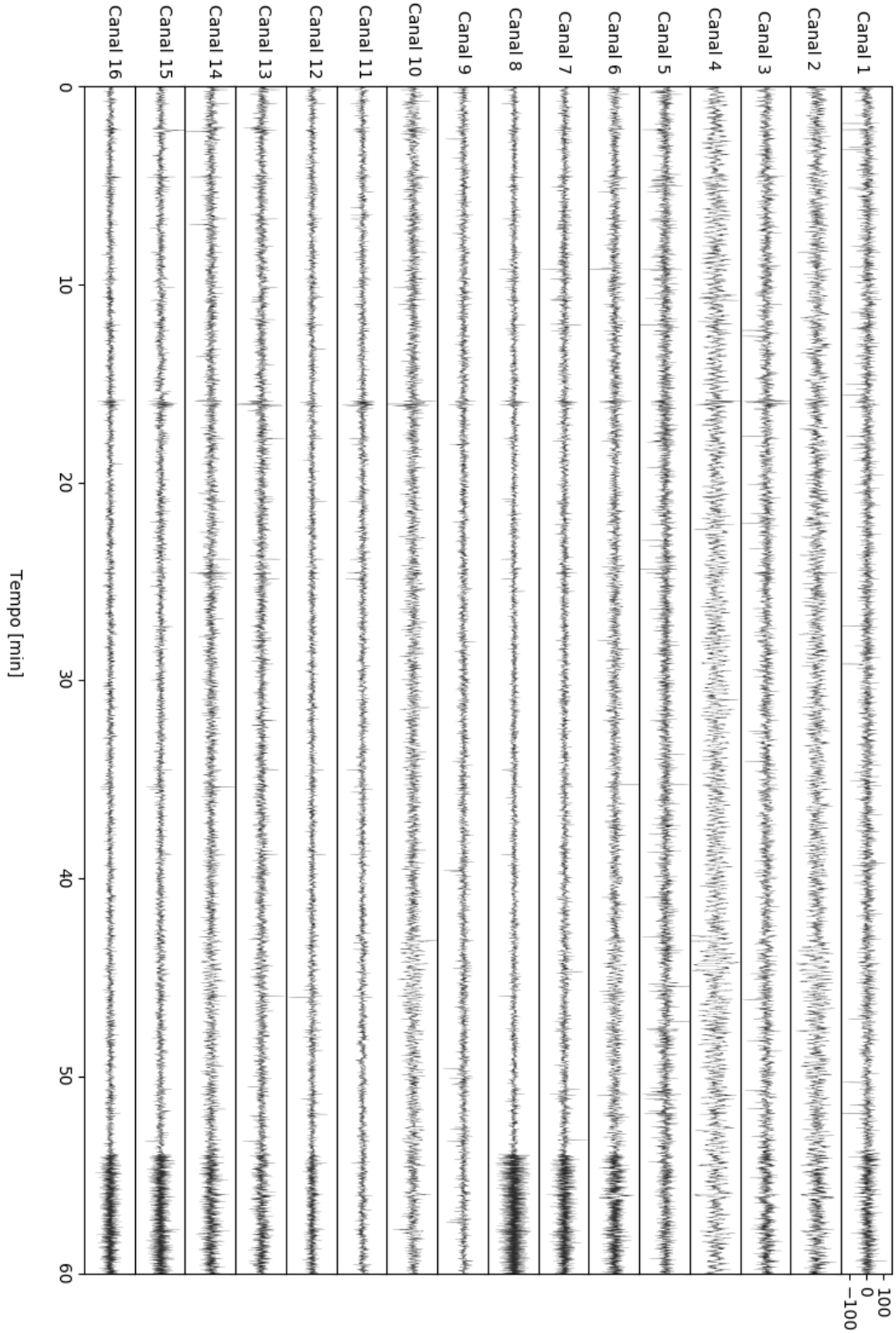

Figura 5 - Pré-ataque característico com uma hora de duração com amplitude do sinal variando de +100 a -100 em todos canais.

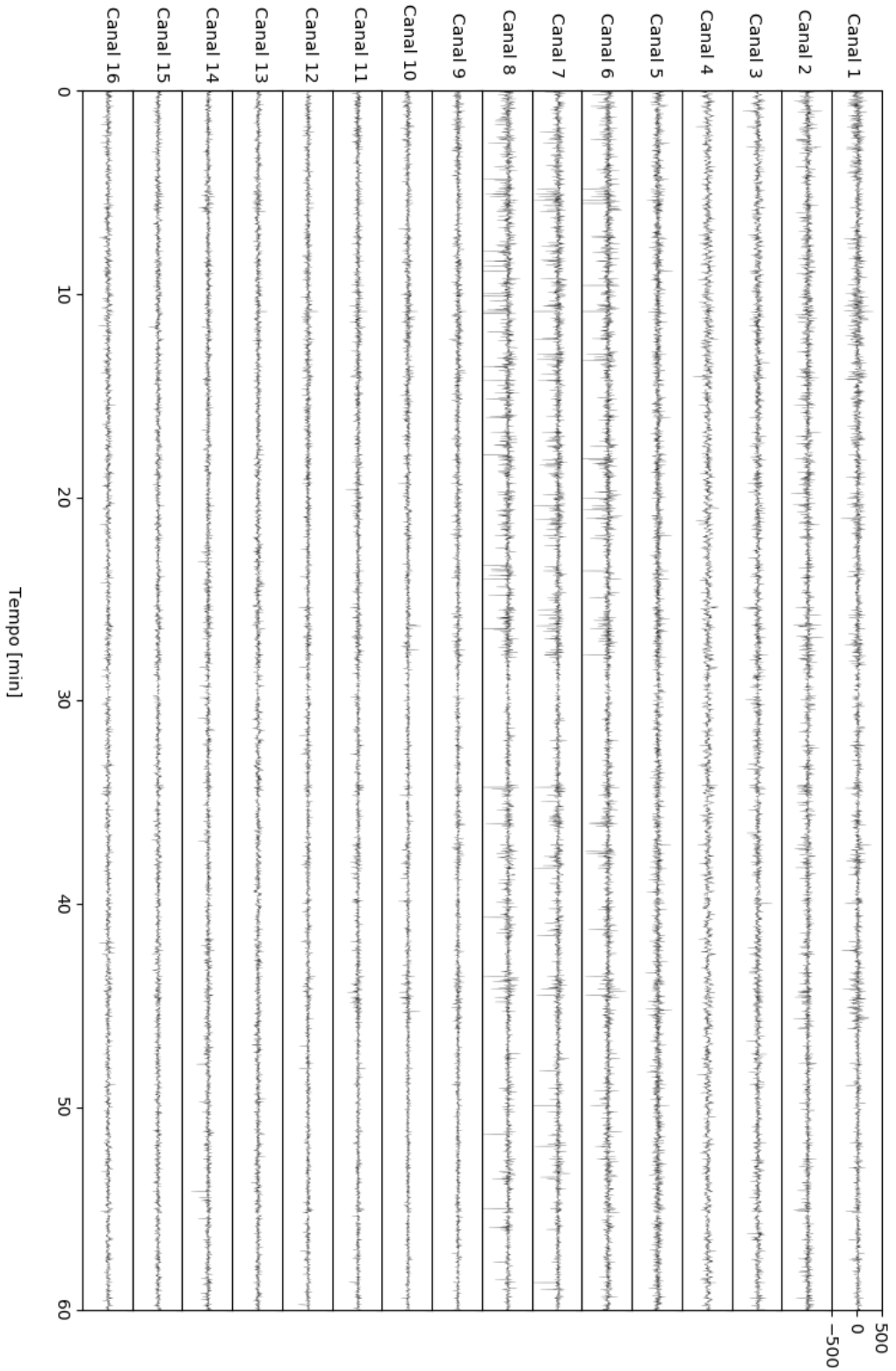

Figura 6 - Inter-ataque característico com uma hora de duração com amplitude do sinal variando de +500 a -500 em todos os canais

3.9 Média e Desvio Padrão Deslizantes, Transformada de Fourier e Determinação da Janela de Amostragem

A próxima etapa do trabalho consistiu em determinar qual a melhor maneira para se realizar a análise estatística e espectral da base de dados. Inicialmente, a maneira encontrada foi dividir os sinais com duração de uma hora em partes iguais, em janelas em torno de 5 minutos, e calcular a média e o desvio padrão destas janelas. Contudo, esta abordagem apresentou alguns problemas, a escolha do tamanho da janela foi demasiado longa porque os valores das médias destas janelas sempre resultavam muito próximo de zero. A razão pela qual a média apresentou valores quase nulos é devido ao fato de que o filtro de média é um tipo de filtro passa-baixa, que tem como principal propriedade filtrar frequências tão mais baixas quanto maior for a janela. Aplicando esta propriedade ao sinal estudado, se tornou evidente que para que as janelas apresentassem valores de média significativos, o tamanho da janela teria que ser reduzido.

Além de se observar a necessidade da diminuição do tamanho da janela de amostragem, a implementação de uma sobreposição, ou *overlap*, entre as janelas subsequentes se mostrou bastante interessante. O uso de janelas que caminham ao longo do sinal se sobrepondo nada mais é do que uma janela deslizante, e esta foi a ferramenta escolhida.

No caso da análise espectral, aplicar uma janela deslizante lançando mão da Transformada de Fourier é, na verdade, utilizar uma ferramenta matemática chamada espectrograma. O espectrograma fornece o perfil espectral do sinal ao longo do tempo. Dessa forma, ficou estabelecido que seria usada a janela deslizante para as análises de média e desvio padrão deslizante, e espectrograma para análise espectral. A fim de validar análises futuras, é essencial que tanto as janelas deslizantes quanto o espectrograma atuem com um mesmo tamanho de janela e sobreposição de janela nos sinais em questão, para que seja possível um confrontamento entre os valores de média, desvio padrão e componentes espectrais dos diferentes pré-ataques e inter-ataques.

A biblioteca *Pandas* possui uma função já implementada de janela deslizante para média e desvio padrão, chamadas de *pandas.rolling.mean* e *pandas.rolling.std*, respectivamente. Porém, o uso destas funções se apresentou problemático por conta de dificuldades em se configurar a sobreposição de janelas. A solução

encontrada foi gerar novas funções responsáveis por realizar estas tarefas. Tais funções se encontram na seção de apêndices.

Então, outra etapa importante do trabalho foi determinar uma janela de dados que garantisse uma satisfatória análise dos dados. Novamente, recorreu-se aos gráficos de linhas para auxiliar na determinação desta janela. A seguir serão mostrados gráficos para média, desvio padrão e espectrograma de diferentes janelas.

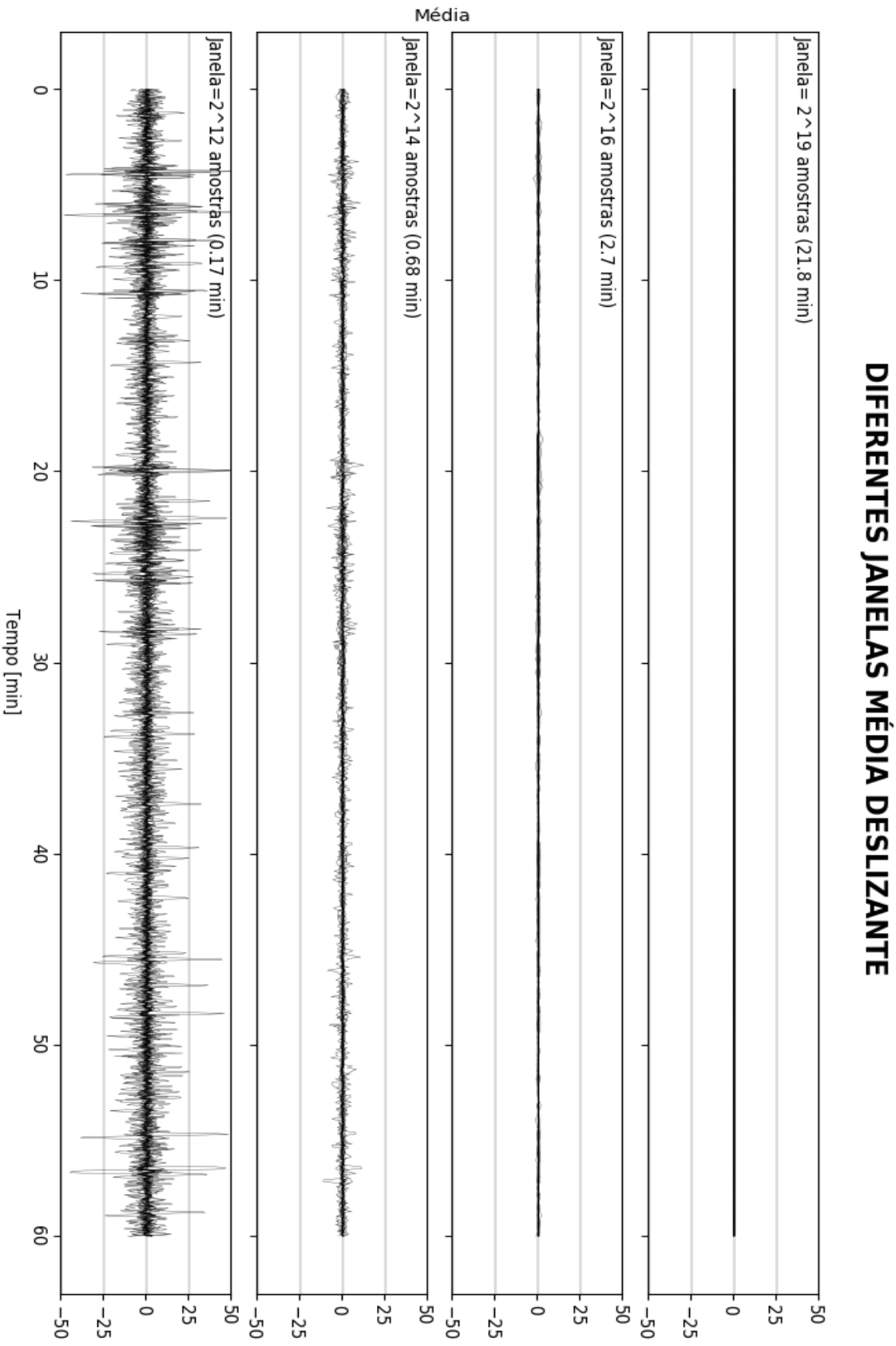

Figura 7 - Media deslizante para diferentes janelas de amostragem.

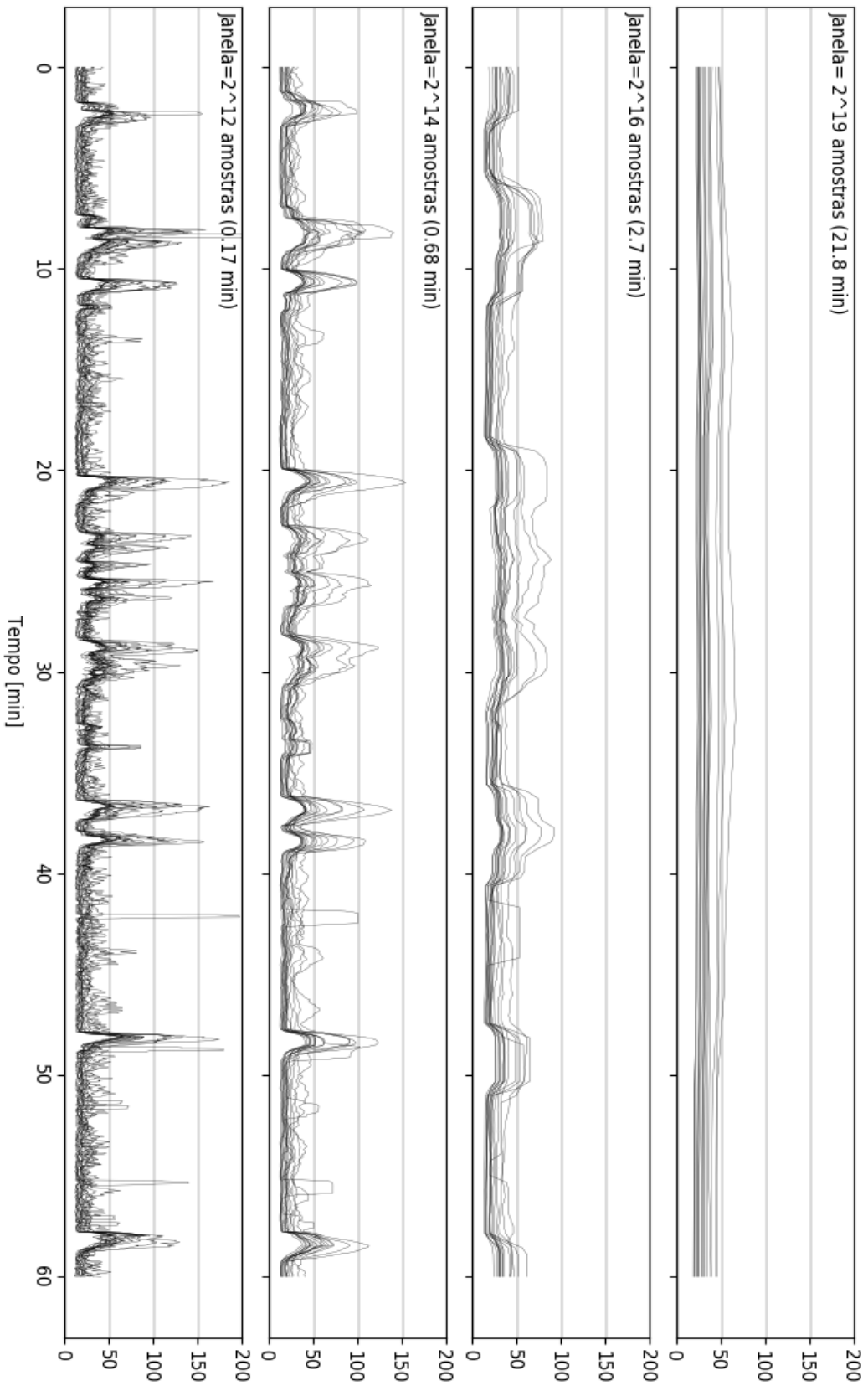

Figura 8 - Desvio Padrão deslizante para diferentes janelas de amostragem.

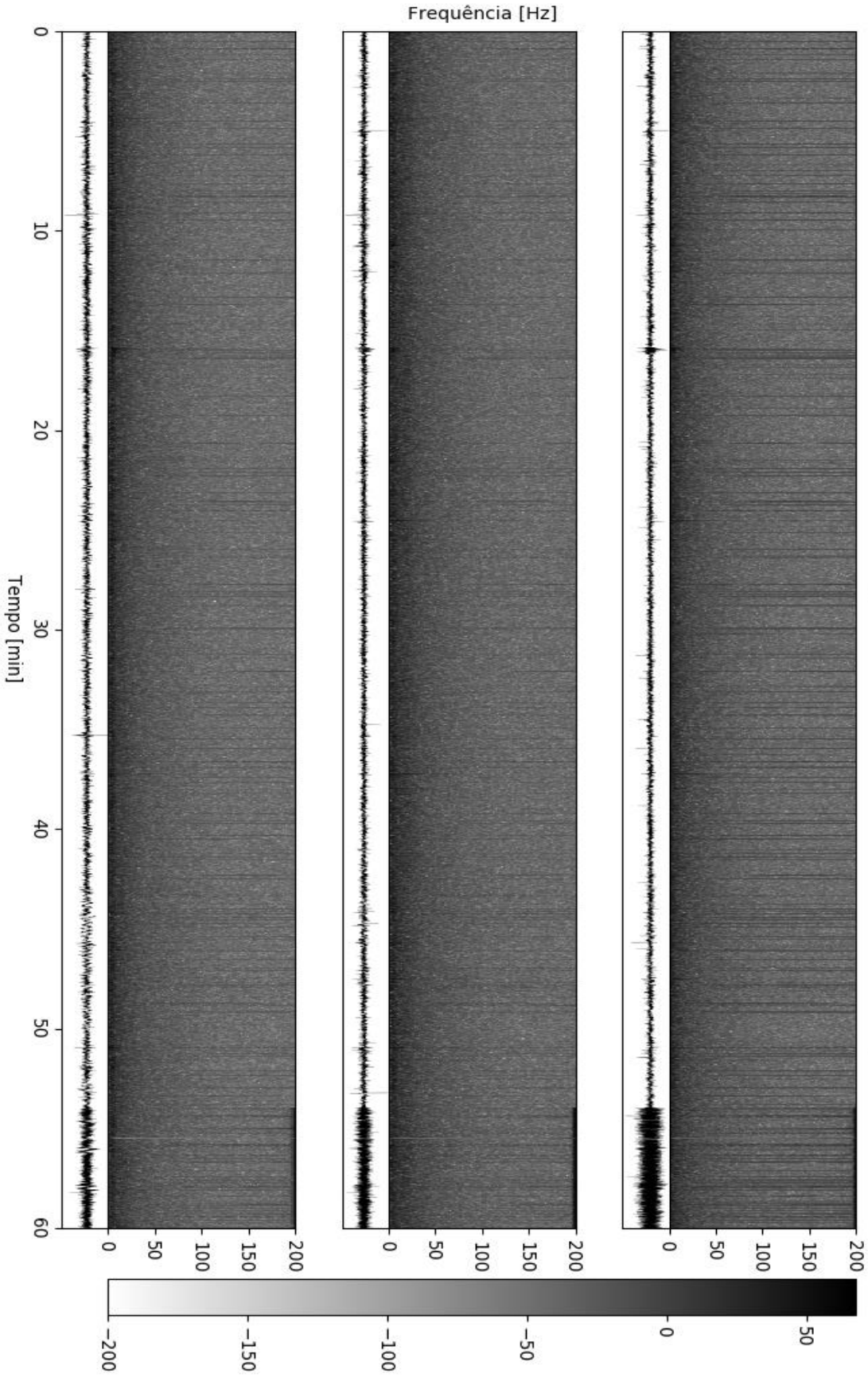

Figura 9 - Espectograma para janela de 0.01 minuto (ou 256 amostras).

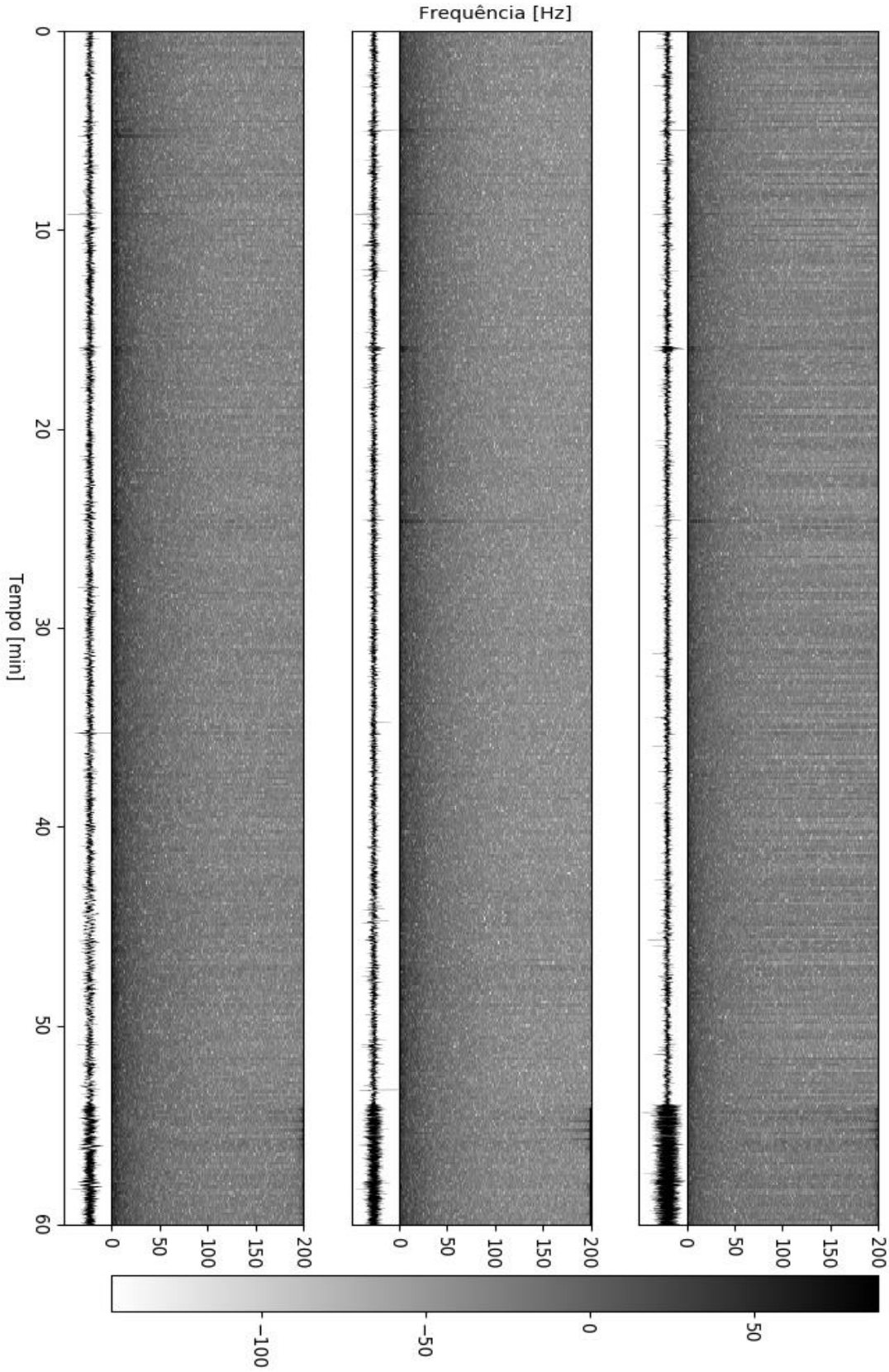

Figura 10 - Espectograma para janela de 0.17 minuto (ou 4096 amostras).

A partir da Figuras 7 e 8, conclui-se que a janela de amostragem que apresentou os melhores resultados foi a de 0,17 minuto (ou 4096 amostras), janela esta em que é possível discriminar valores de média e desvio padrão significativos. Entretanto, a janela de 4096 amostras não poderia ser mais reduzida por conta do que mostra o espectrograma da Figura 9, na qual se utiliza uma janela de 0,01 minuto (256) amostras e parece haver sobreposição de espectro devido à uma janela demasiadamente pequena. Portanto, a janela de 0,17 minuto se mostrou como a melhor opção para se realizar a análise espectral, de média e desvio padrão dos dados. A janela de sobreposição escolhida possui ¼ do tamanho da janela de amostragem, ou seja, 1024 amostras.

4 RESULTADOS E DISCUSSÕES

Na seção anterior, foi apresentada a metologia para se encontrar o melhor meio de análise estatística e espectral da base de dados, o que representou a maior demanda de tempo em todo o trabalho. E assim, foi estabelecido que média e desvio padrão com uma janela de amostragem de 4096 e *overlap* de 1024 é a melhor opção para este trabalho. Este tamanho de janela também foi utilizado para se calcular as Transformadas de Fourier do sinal. Umas das valiosas informações que a Transformada de Fourier fornece sobre o sinal é o espectro de potência, a fim de se obter um único valor representativo da janela amostrada, calculou-se a energia total do sinal, soma das potências espectrais, para que este valor possa ser comparado com os outros parâmetros calculados como veremos nas figuras finais deste trabalho.

Reiterando que o objetivo deste trabalho é estabelecer diferenças entre os dois estados distintos de um ataque epiléptico, pré e inter-ataques, a preocupação central para gerar os resultados consistiu em sintetizar o máximo de dados possíveis para que as diferenças entre os dois estados possam ser mais facilmente destacadas. Nesse sentido, gráficos de dispersão que relacionassem média, desvio padrão e energia dos sinais foram a melhor solução encontrada. Apenas ataques do mesmo paciente foram confrontados, como por exemplo, não foram feitas comparações entre pré-ataques do paciente 1 com o inter-ataques do paciente 3 porque cada paciente possui um desenvolvimento cerebral distinto, também há de se considerar que não há informações que garantem que os eletrodos dos três pacientes foram implantados na mesma região cerebral, tornando a comparação entre diferentes pacientes pouco assertiva.

Os gráficos de dispersão de cada paciente possuem duas colunas, pré ataque à esquerda e inter-ataque à direita. Nos gráficos das três primeiras linhas, foram plotados energia, média, e desvio padrão seguindo uma combinação simples. Nas últimas três linhas, foram confrontados diferentes parâmetros respeitando a ordem do pré e inter-ataques. Por exemplo, os valores de média do “pre_ataque_1” foi confrontado os valores de desvio padrão do “pre_ataque_1”, depois valores de média do “pre_ataque_2” com os valores de desvio padrão do “pre_ataque_2” e assim por diante. Os resultados são apresentados a seguir:

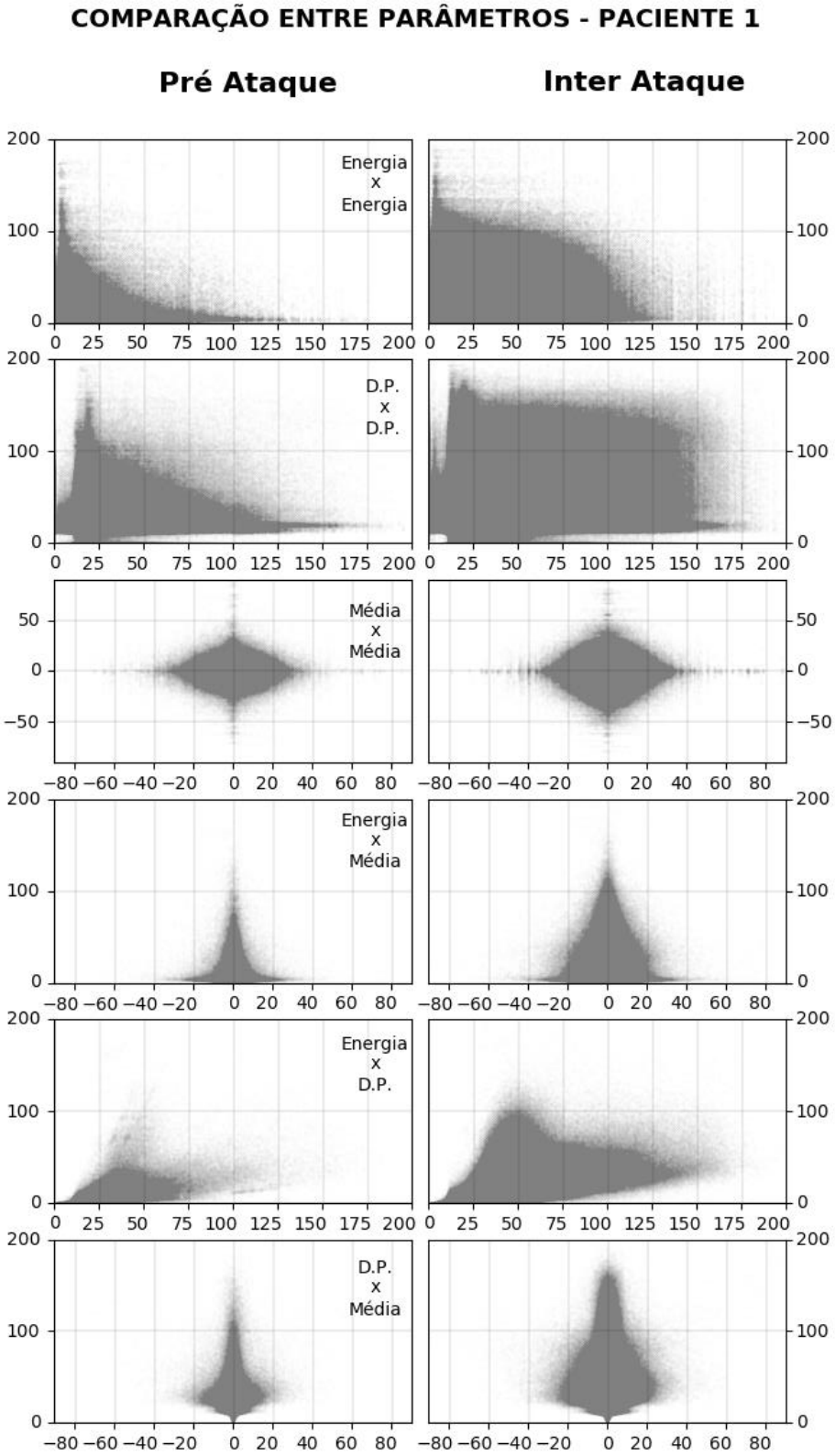

Figura 11 - Gráficos de dispersão cruzando parâmetros do Paciente 1.

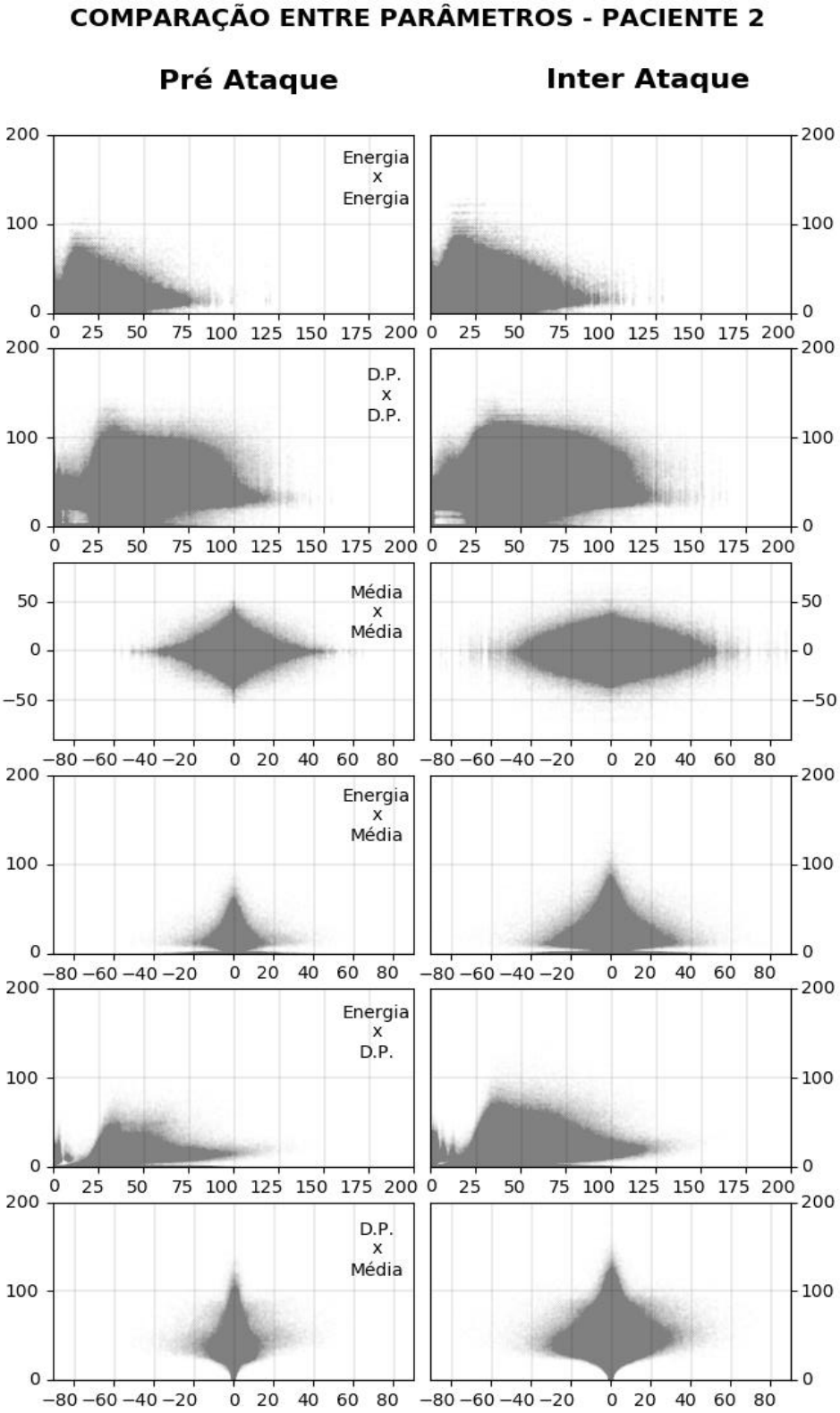

Figura 12 - - Gráficos de dispersão cruzando parâmetros do Paciente 2.

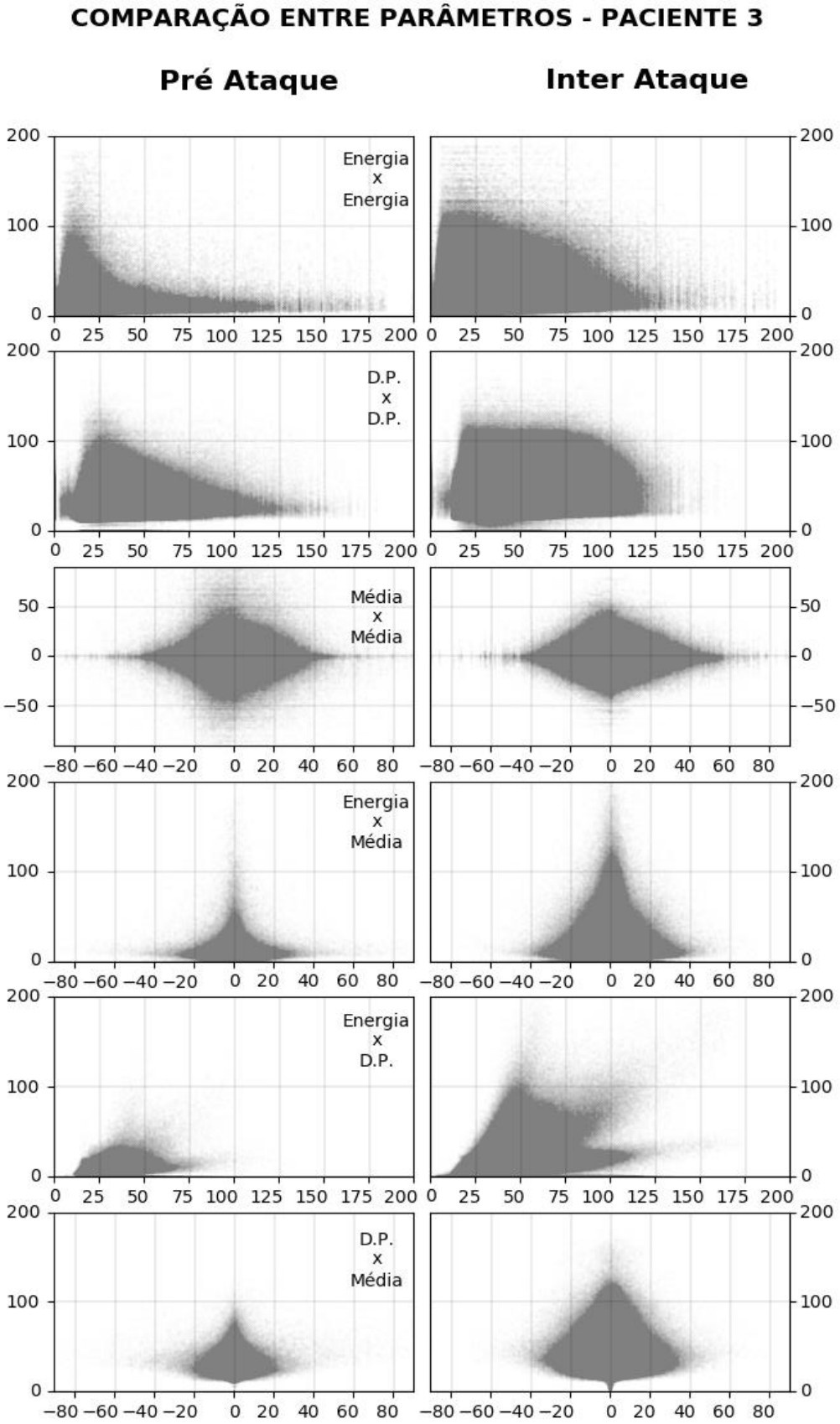

Figura 13 - - Gráficos de dispersão cruzando parâmetros do Paciente 3.

Os gráficos de dispersão mostrados podem oferecer uma direção a fim de se distinguir os dois estados distintos de ataques de epilepsia.

No paciente 1, tanto os gráficos da primeira quanto da segunda linha demonstram claramente que há uma faixa de valores de energia e desvio padrão que somente é atingido no inter-ataque, essas características – que inter-ataques possuem valores de energia e desvio padrão mais elevados – já tinham sido constadas visualmente quando se plotou alguns gráficos de linhas para analisar o comportamento dos parâmetros. Mas, o gráfico de dispersão levando em consideração os diferentes estados representa esta informação melhor. Em relação aos valores de média, percebe-se que também há uma maior variação do valor deste parâmetro nos inter-ataques, porém bem menos discrepantes do que no caso da energia e desvio padrão. Os gráficos das três últimas linhas reafirmam que há diferenças bastante significativas nos diferentes estados.

No paciente 2, ao contrário do paciente 1, os gráficos das duas primeiras linhas não fornecem nenhuma diferença significativa em termos de energia e desvio padrão, apresentando gráficos bastante semelhantes. Em relação ao valores de média, na terceira linha do gráfico, este parâmetro apresentou o melhor desempenho no sentido de se diferenciar os dois estados, no inter-ataque os valores de média oscilar numa faixa mais ampla, tal como acontece no paciente 1, mas de maneira mais intensa. O gráfico da última linha, que compara a média com desvio padrão, representa bem como os valores de média no inter-ataque assumem valores que em nenhum momento é assumido nos pré-ataques.

No paciente 3, à semelhança do paciente 1, os gráficos das duas primeiras linhas também apresentar valores discrepantes no caso de inter-ataques, claramente este estado assume faixas de valores que não são assumidas no caso complementar. O gráfico da terceira linha, das médias, é novamente pouco conclusivo. O gráfico que confronta energia e desvio padrão, da quarta linha, parece ser o mais significativo em termos de diferenças entre os dois estados

No geral, os resultados se mostraram satisfatórios. Dados de energia e desvio padrão se configuraram, neste caso estudado, como as melhores opções para se distinguir os diferentes estados. Os valores de média, em menor medida, também podem ser um indicativo de diferenciação. Cruzar os diferentes parâmetros se

mostrou bastante útil para elucidar algumas características que, isoladamente, os parâmetros não mostravam.

Outra ferramenta estatística que pode auxiliar no problema é o uso de teste de hipóteses, que podem avaliar probabilisticamente a chance do paciente se encontrar em um estado de pré ou inter-ataque.

5 CONCLUSÃO

Epilepsia é uma doença crônica que atinge uma parcela significativa da população mundial, que diminui consideravelmente a qualidade de vida daqueles que sofrem da doença. Avanços no sentido de estabelecer métodos de predição seriam de enorme importância para muitas pessoas. Como abordado neste Trabalho de Conclusão de Curso, no qual foi dada uma abordagem simples do ponto de vista da estatística, já foi possível estabelecer algumas diferenças que podem auxiliar a distinção dos diferentes estados da doença. Parâmetros como energia do sinal e desvio padrão, analisados em pequenas janelas de tempo, a fim de se considerar variações mais abruptas, podem ser o caminho para se estabelecer predições assertivas. Com efeito, é necessária uma análise estatística mais profunda, utilizando ferramentas mais poderosas, para que se possa estabelecer um quadro confiável de diferenciação de estados de epilepsia. E este é uma tarefa, em grande parte, dos cientistas de dados, que através do conhecimento de programação e estatística, podem fornecer *insights* valiosos para melhorar a qualidade de vida destas pessoas.

6 BIBLIOGRAFIA

Cook MJ, O'Brien TJ, Berkovic SF, Murphy M, Morokoff A, Fabinyi G, D'Souza W, Yerra R, Archer J, Litewka L, Hosking S, Lightfoot P, Ruedebusch V, Sheffield WD, Snyder D, Leyde K, Himes D (2013) Prediction of seizure likelihood with a long-term, implanted seizure advisory system in patients with drug-resistant epilepsy: a first-in-man study. LANCET NEUROL 12:563-571.

Brinkmann, B. H., Wagenaar, J., Abbot, D., Adkins, P., Bosshard, S. C., Chen, M., ... & Pardo, J. (2016). Crowdsourcing reproducible seizure forecasting in human and canine epilepsy. Brain, 139(6), 1713-1722.

Gadhoumi, K., Lina, J. M., Mormann, F., & Gotman, J. (2016). Seizure prediction for therapeutic devices: A review. Journal of neuroscience methods, 260, 270-282.

Karoly, P. J., Freestone, D. R., Boston, R., Grayden, D. B., Himes, D., Leyde, K., ... & Cook, M. J. (2016). Interictal spikes and epileptic seizures: their relationship and underlying rhythmicity. Brain, aww019.

Andrzejak RG, Chicharro D, Elger CE, Mormann F (2009) Seizure prediction: Any better than chance? Clin Neurophysiol.

Snyder DE, Echauz J, Grimes DB, Litt B (2008) The statistics of a practical seizure warning system. J Neural Eng 5: 392–401.

Mormann F, Andrzejak RG, Elger CE, Lehnertz K (2007) Seizure prediction: the long and winding road. Brain 130: 314–333.

Haut S, Shinnar S, Moshe SL, O'Dell C, Legatt AD. (1999) The association between seizure clustering and status epilepticus in patients with intractable complex partial seizures. Epilepsia 40:1832–1834.

ZSCHOCKE, P. D., SCHILTZ, E. and SCHULZ, G. E. (1993), Purification and sequence determination of guanylate kinase from pig brain. European Journal of Biochemistry, 213: 263–269. doi:10.1111/j.1432-1033.1993.tb17757.x

DEUTSCH, Andreas; DRESS, Andreas; RENSING, Ludger. Formation of morphological differentiation patterns in the ascomycete Neurospora crassa. Mechanisms of development, v. 44, n. 1, p. 17-31, 1993.

NIEDERMEYER, E. The electrocerebellogram. Clinical EEG and neuroscience, v. 35, n. 2, p. 112-115, 2004.

WOLPAW, Jonathan R. et al. Brain–computer interfaces for communication and control. Clinical neurophysiology, v. 113, n. 6, p. 767-791, 2002.

Siegelbaum, Steven A., and A. James Hudspeth. Principles of neural science. Eds. Eric R. Kandel, James H. Schwartz, and Thomas M. Jessell. Vol. 4. New York: McGraw-hill, 2000.

AKAY, Metin. Wavelet applications in medicine. IEEE spectrum, v. 34, n. 5, p. 50-56, 1997.

LATHI, Bhagwandas Pannalal. Signals and systems. Berkeley-Cambridge Press, 1987.

Apêndice

1. FUNÇÕES IMPLEMENTADAS

```
import matplotlib.pyplot as plt
from scipy.stats.stats import pearsonr
import numpy as np
import pandas as pd
import os
import glob
import scipy.fftpack
```

```
                                                    #FUNÇÃO DFT
def mov_fft(df,janela,passo):
 amostras = len(df)
 aux = True
 fft_raise_list =[]
 n=0

 while aux:
   a = scipy.fftpack.fft(df[n:n+janela])
   fft_abs = 2.0/janela*np.abs(a[:janela//2][1:])
   for canal in range(16):
      fft_raise = sum(np.power(fft_abs[:,canal],2))
      fft_raise_list.append(fft_raise)

   energy = [fft_raise_list[x:x+16] for x in range(0, len(fft_raise_list), 16)]

   n = passo + n
   if n > (amostras-janela):
         aux = False
 return energy
```

```
                                                    #FUNÇÃO MÉDIA DESLIZANTE
def mov_avg(df,janela,passo):
 amostras = len(df)
 aux = True
 media = []
 n=0
 while aux:
   a =df[n:n+janela].mean()
   media.append(a)
   n = passo + n
   if n > (amostras-janela):
         aux = False
 return media
```

```
                                                    # FUNÇÃO DESVIO PADRAO DESLIZANTE
def mov_std(df,janela,passo):
 amostras = len(df)
 aux = True
 std = []
 n=0
 #fig = plt.figure()
 while aux:
   a =df[n:n+janela].std()
   std.append(a)
   n = passo + n
   if n > (amostras-janela):
         aux = False

 return std
```

```
#FUNÇÃO RETORNA TODOS PRÉ-ATAQUES - PACIENTE 1

def pre_ataques_p1():

  path1 = 'C:/Backup/Paciente_1/pre_ataque'
  numero_ataques = os.listdir(path1)

  ataque_inteiro = pd.DataFrame()
  todos_pre_ataques_p1 = []

  for i in range(1,len(numero_ataques)+1):

     path = 'C:/Backup/Paciente_1/pre_ataque/Pre_ataque_'+str(i)
     seis_arquivos = glob.glob(path + "/*.csv")
     lista = []


     for arquivo in seis_arquivos:
        print("Reading file", arquivo, " attack:",i)
        df = pd.read_csv(arquivo,index_col=None, header=None, dtype=np.float32)
        lista.append(df)

     ataque_inteiro = pd.concat(lista,ignore_index=True)
     ataque_inteiro.columns = ['canal_1',
'canal_2','canal_3','canal_4','canal_5','canal_6','canal_7','canal_8','canal_9','canal_10','canal_11','canal_12','canal_13','canal_14','canal_1
5','canal_16']

     todos_pre_ataques_p1.append(ataque_inteiro)

  return todos_pre_ataques_p1
```

```
#FUNÇÃO RETORNA TODOS PRÉ-ATAQUES - PACIENTE 2

def pre_ataques_p2():

  path1 = 'C:/Backup/Paciente_2/pre_ataque'
  numero_ataques = os.listdir(path1)

  ataque_inteiro = pd.DataFrame()
  todos_pre_ataques_p2 = []

  for i in range(1,len(numero_ataques)+1):

     path = 'C:/Backup/Paciente_2/pre_ataque/Pre_ataque_'+str(i)
     seis_arquivos = glob.glob(path + "/*.csv")
     lista = []


     for arquivo in seis_arquivos:
        print("Reading file", arquivo, " attack:",i)
        df = pd.read_csv(arquivo,index_col=None, header=None, dtype=np.float32)
        lista.append(df)

     ataque_inteiro = pd.concat(lista,ignore_index=True)
     ataque_inteiro.columns = ['canal_1',
'canal_2','canal_3','canal_4','canal_5','canal_6','canal_7','canal_8','canal_9','canal_10','canal_11','canal_12','canal_13','canal_14','canal_1
5','canal_16']

     todos_pre_ataques_p2.append(ataque_inteiro)

  return todos_pre_ataques_p2
```

```
#FUNÇÃO RETORNA TODOS PRÉ-ATAQUES - PACIENTE 3

def pre_ataques_p3():

  path1 = 'C:/Backup/Paciente_3/pre_ataque'
  numero_ataques = os.listdir(path1)

  ataque_inteiro = pd.DataFrame()
  todos_pre_ataques_p3 = []

  for i in range(1,len(numero_ataques)+1):

     path = 'C:/Backup/Paciente_3/pre_ataque/Pre_ataque_'+str(i)
     seis_arquivos = glob.glob(path + "/*.csv")
     lista = []


     for arquivo in seis_arquivos:
        print("Reading file", arquivo, " attack:",i)
        df = pd.read_csv(arquivo,index_col=None, header=None, dtype=np.float32)
        lista.append(df)

     ataque_inteiro = pd.concat(lista,ignore_index=True)
     ataque_inteiro.columns = ['canal_1',
'canal_2','canal_3','canal_4','canal_5','canal_6','canal_7','canal_8','canal_9','canal_10','canal_11','canal_12','canal_13','canal_14','canal_1
5','canal_16']

     todos_pre_ataques_p3.append(ataque_inteiro)

  return todos_pre_ataques_p3
```

```
#FUNÇÃO RETORNA TODOS INTER-ATAQUES - PACIENTE 1

def inter_ataques_p1():

  path1 = 'C:/Backup/Paciente_1/inter_ataque'
  numero_ataques = os.listdir(path1)

  ataque_inteiro = pd.DataFrame()
  todos_inter_ataques_p1 = []

  for i in range(0,len(numero_ataques)):

     path = 'C:/Backup/Paciente_1/inter_ataque/Inter_ataque_'+str(i)
     seis_arquivos = glob.glob(path + "/*.csv")
     lista = []


     for arquivo in seis_arquivos:
        print("Reading file", arquivo, " attack:",i)
        df = pd.read_csv(arquivo,index_col=None, header=None, dtype=np.float32)
        lista.append(df)

     ataque_inteiro = pd.concat(lista,ignore_index=True)
     ataque_inteiro.columns = ['canal_1',
'canal_2','canal_3','canal_4','canal_5','canal_6','canal_7','canal_8','canal_9','canal_10','canal_11','canal_12','canal_13','canal_14','canal_1
5','canal_16']

     todos_inter_ataques_p1.append(ataque_inteiro)

  return todos_inter_ataques_p1
```

```
#FUNÇÃO RETORNA TODOS INTER-ATAQUES - PACIENTE 2

def inter_ataques_p2():

  path1 = 'C:/Backup/Paciente_2/inter_ataque'
  numero_ataques = os.listdir(path1)

  ataque_inteiro = pd.DataFrame()
  todos_inter_ataques_p2 = []

  for i in range(100,len(numero_ataques)):

     path = 'C:/Backup/Paciente_2/inter_ataque/Inter_ataque_'+str(i)
     seis_arquivos = glob.glob(path + "/*.csv")
     lista = []


     for arquivo in seis_arquivos:
        print("Reading file", arquivo, " attack:",i)
        df = pd.read_csv(arquivo,index_col=None, header=None, dtype=np.float32)
        lista.append(df)

     ataque_inteiro = pd.concat(lista,ignore_index=True)
     ataque_inteiro.columns = ['canal_1',
'canal_2','canal_3','canal_4','canal_5','canal_6','canal_7','canal_8','canal_9','canal_10','canal_11','canal_12','canal_13','canal_14','canal_1
5','canal_16']

     todos_inter_ataques_p2.append(ataque_inteiro)

  return todos_inter_ataques_p2
```

```
#FUNÇÃO RETORNA TODOS INTER-ATAQUES - PACIENTE 3

def inter_ataques_p3():

  path1 = 'C:/Backup/Paciente_3/inter_ataque'
  numero_ataques = os.listdir(path1)

  ataque_inteiro = pd.DataFrame()
  todos_inter_ataques_p3 = []

  for i in range(0,len(numero_ataques)):

     path = 'C:/Backup/Paciente_3/inter_ataque/Inter_ataque_'+str(i)
     seis_arquivos = glob.glob(path + "/*.csv")
     lista = []


     for arquivo in seis_arquivos:
        print("Reading file", arquivo, " attack:",i)
        df = pd.read_csv(arquivo,index_col=None, header=None, dtype=np.float32)
        lista.append(df)

     ataque_inteiro = pd.concat(lista,ignore_index=True)
     ataque_inteiro.columns = ['canal_1',
'canal_2','canal_3','canal_4','canal_5','canal_6','canal_7','canal_8','canal_9','canal_10','canal_11','canal_12','canal_13','canal_14','canal_1
5','canal_16']

     todos_inter_ataques_p3.append(ataque_inteiro)

  return todos_inter_ataques_p3
```

2. GRÁFICOS DE LINHA

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd



pre_p2 = pre_ataques_p2()
pre_p3 = pre_ataques_p3()
pre_p1 = pre_ataques_p1()
inter_p2 = inter_ataques_p2()

fig,axs = plt.subplots(16, 1, sharex=True,figsize=(11.69,8.27),dpi=100)
fig.subplots_adjust(hspace=0)

x =  np.linspace(0,60,len(pre_p2[0].iloc[::200,0]))

for k in range(16):

   axs[k].plot(x, pre_p1[4].iloc[::200,k],color = 'black',  linewidth =0.1)
   axs[k].yaxis.set_visible(False)
   axs[k].set_xlim([0,60])
   axs[k].set_ylim([-150,150])

fig.text(0.5, 0.04, 'Tempo [min]', ha='center', va='center')
fig.subplots_adjust(bottom =0.11,right=0.96,left=0.12,hspace=0.00,top=0.92)
plt.suptitle('INTER-ATAQUE - PACIENTE 2',fontsize= 16, fontweight="bold")



fig.text(0.089, 0.90-.01, 'Canal 1', ha='center', va='center')
fig.text(0.089, 0.85-.01, 'Canal 2 ', ha='center', va='center')
fig.text(0.089, 0.80-.01, 'Canal 3', ha='center', va='center')
fig.text(0.089, 0.75-.01, 'Canal 4 ', ha='center', va='center')
fig.text(0.089, 0.70-.01, 'Canal 5', ha='center', va='center')
fig.text(0.089, 0.65-.01, 'Canal 6', ha='center', va='center')
fig.text(0.089, 0.60-.01, 'Canal 7', ha='center', va='center')
fig.text(0.089, 0.55-.01, 'Canal 8 ', ha='center', va='center')
fig.text(0.089, 0.50-.01, 'Canal 9', ha='center', va='center')
fig.text(0.089, 0.45-.01, 'Canal 10 ', ha='center', va='center')
fig.text(0.089, 0.40-.01, 'Canal 11', ha='center', va='center')
fig.text(0.089, 0.35-.01, 'Canal 12', ha='center', va='center')
fig.text(0.089, 0.30-.01, 'Canal 13', ha='center', va='center')
fig.text(0.089, 0.25-.01, 'Canal 14', ha='center', va='center')
fig.text(0.089, 0.20-.01, 'Canal 15', ha='center', va='center')
fig.text(0.089, 0.15-.01, 'Canal 16', ha='center', va='center')



axs[0].yaxis.set_visible(True)
axs[0].yaxis.tick_right()
```

3. ALGORITMO PARA ARMAZENAR OS DADOS ESTATÍSTICOS EM ARQUIVOS .TXT

```
import pickle

pre_p1 = pre_ataques_p1()
pre_p2 = pre_ataques_p2()
pre_p3 = pre_ataques_p3()
inter_p1 = inter_ataques_p1()
inter_p2 = inter_ataques_p2()
inter_p3 = inter_ataques_p3()

energia_inter_p3 = []
for ataque in range (len(inter_p3)):
  print('fft inter-ataque:' ,ataque)
  fft = mov_fft(inter_p3[ataque].iloc[:,:],4096,4096//4)
  energia_inter_p3.append(fft)



std_pre_p1 = []
for ataque in range (len(pre_ataque_p1)):
  print('std ataque:' ,ataque)
  std = mov_std(pre_ataque_p1[ataque].iloc[:,:],4096,4096//4)
  std_pre_p1.append(std)



media_inter_p3 = []
for ataque in range (len(inter_p3)):
  print('media ataque:' ,ataque)
  media = mov_avg(inter_p3[ataque].iloc[:,:],4096,4096//4)
  media_inter_p3.append(media)

#PACIENTE UM

with open('C:/Users/User/Desktop/Salvando_listas/energia_pre_p1.txt', 'rb') as f:
  energia_pre_p1 = pickle.load(f)

with open('C:/Users/User/Desktop/Salvando_listas/energia_inter_p1.txt', 'rb') as f:
  energia_inter_p1 = pickle.load(f)

with open('C:/Users/User/Desktop/Salvando_listas/std_pre_p1.txt', 'rb') as f:
  std_pre_p1 = pickle.load(f)

with open('C:/Users/User/Desktop/Salvando_listas/std_inter_p1.txt', 'rb') as f:
  std_inter_p1 = pickle.load(f)

with open('C:/Users/User/Desktop/Salvando_listas/media_pre_p1.txt', 'rb') as f:
  media_pre_p1 = pickle.load(f)
```

```
with open('C:/Users/User/Desktop/Salvando_listas/media_inter_p1.txt', 'rb') as f:
  media_inter_p1 = pickle.load(
#PACIENTE DOIS

with open('C:/Users/User/Desktop/Salvando_listas/energia_pre_p2.txt', 'rb') as f:
  energia_pre_p2 = pickle.load(f)

with open('C:/Users/User/Desktop/Salvando_listas/energia_inter_p2.txt', 'rb') as f:
  energia_inter_p2 = pickle.load(f)

with open('C:/Users/User/Desktop/Salvando_listas/std_pre_p2.txt', 'rb') as f:
  std_pre_p2 = pickle.load(f)


with open('C:/Users/User/Desktop/Salvando_listas/std_inter_p2.txt', 'rb') as f:
  std_inter_p2 = pickle.load(f)


with open('C:/Users/User/Desktop/Salvando_listas/media_pre_p2.txt', 'rb') as f:
  media_pre_p2 = pickle.load(f)


with open('C:/Users/User/Desktop/Salvando_listas/media_inter_p2.txt', 'rb') as f:
  media_inter_p2 = pickle.load(f)

#PACIENTE TRÊS

with open('C:/Users/User/Desktop/Salvando_listas/energia_pre_p3.txt', 'rb') as f:
  energia_pre_p3 = pickle.load(f)

with open('C:/Users/User/Desktop/Salvando_listas/energia_inter_p3.txt', 'rb') as f:
  energia_inter_p3 = pickle.load(f)

with open('C:/Users/User/Desktop/Salvando_listas/std_pre_p3.txt', 'rb') as f:
  std_pre_p3 = pickle.load(f)


with open('C:/Users/User/Desktop/Salvando_listas/std_inter_p3.txt', 'rb') as f:
  std_inter_p3 = pickle.load(f)

with open('C:/Users/User/Desktop/Salvando_listas/media_pre_p3.txt', 'rb') as f:
  media_pre_p3 = pickle.load(f)

with open('C:/Users/User/Desktop/Salvando_listas/media_inter_p3.txt', 'rb') as f:
  media_inter_p3 = pickle.load(f)
```

4. ALGORITMO PARA GERAR ESPETROGRAMA E GRÁFICOS DE LINHAS

```
from funcoes_TCC import mov_avg, mov_std,pre_ataques_p1, pre_ataques_p2, pre_ataques_p3, inter_ataques_p1, inter_ataques_p2, inter_ataques_p3
from scipy import signal
from scipy.signal import spectrogram
import matplotlib.gridspec as gridspec



# Carrega pré - ataques
#pre_p1 = pre_ataques_p1()
for nome in range(0,3):

#Cria o Grid com os subplots
   figure = plt.figure(figsize=((11.69,8.27)),dpi=100)
   G = gridspec.GridSpec(18,30)

   ax1 = plt.subplot(G[0:4,0:28])
   ax1.xaxis.set_visible(False)
   ax2 = plt.subplot(G[4,0:28])
   ax2.yaxis.set_visible(False)
   ax2.xaxis.set_visible(False)
   ax3 = plt.subplot(G[6:10,0:28])
   ax3.xaxis.set_visible(False)
   ax4 = plt.subplot(G[10,0:28])
   ax4.yaxis.set_visible(False)
   ax4.xaxis.set_visible(False)
   ax5 = plt.subplot(G[12:16,0:28])
   ax5.xaxis.set_visible(False)
   ax6 = plt.subplot(G[16,0:28])
   ax6.yaxis.set_visible(False)
   ax7 = plt.subplot(G[:-2,29])
   ax7.yaxis.set_visible(True)
   ax7.xaxis.set_visible(False)



   tipo_janela = ['boxcar','triang','blackman']
   nome_janela = ['Retangular','Triangular','de Blackman']



   plt.subplots_adjust(top=0.93, left = 0.04, bottom = 0.02,right = 0.95,hspace=0)
   plt.suptitle('ESPECTROGRAMA (Janela '+nome_janela[nome]+' - 0.01 min)',fontsize= 16, fontweight="bold")









#Plota espectograma e sinal no tempo
```

```
ataque=4
nps = 256

x=pre_p1[ataque].iloc[:,7]
x2 =  np.linspace(0,60,len(pre_p1[ataque].iloc[::100,7]))

f, t, Sxx = signal.spectrogram(x,window=tipo_janela[nome], fs=400, nperseg = nps)
test = ax1.pcolormesh(t, f, 20*np.log10(Sxx+10**-10),cmap='Spectral')
ax2.plot(pre_p1[ataque].iloc[::100,7],lw=0.1, color='black')
ax2.set_xlim([0,len(pre_p1[ataque].iloc[:,7])])


x=pre_p1[ataque].iloc[:,6]

f, t, Sxx = signal.spectrogram(x,window=tipo_janela[nome], fs=400, nperseg = nps)
ax3.pcolormesh(t, f, 20*np.log10(Sxx+10**-10),cmap='Spectral')
ax4.plot(pre_p1[ataque].iloc[::100,6],lw=0.1, color='black')
ax4.set_xlim([0,len(pre_p1[ataque].iloc[:,6])])


x=pre_p1[ataque].iloc[:,5]

f, t, Sxx = signal.spectrogram(x,window=tipo_janela[nome], fs=400, nperseg = nps)
ax5.pcolormesh(t, f, 20*np.log10(Sxx+10**-10),cmap='Spectral')
ax6.plot(x2,pre_p1[ataque].iloc[::100,5],lw=0.1, color='black')
ax6.set_xlim([0,60])

ax3.set_ylabel('Frequência [Hz]')
ax6.set_xlabel('Tempo [min]')


ax1.yaxis.tick_right()
ax3.yaxis.tick_right()
ax5.yaxis.tick_right()



figure.colorbar(test, cax=ax7)
```

5. ALGORITMO PARA GERAR GRÁFICOS DE DISPERSÃO COM COMPARAÇÃO DOS PARÂMETROS ESTATÍSTICOS

```
figure,axs = plt.subplots(2, 6,figsize=(11.69,8.27))
axs[1][0].set_title('Pré Ataque', rotation='vertical',x=-.3,y=0.7,fontsize= 16, fontweight="bold")
axs[0][0].set_title('Inter Ataque', rotation='vertical',x=-.3,y=0.8,fontsize= 16, fontweight="bold")

for g in range (6):
  axs[0][g].grid(color='black', linestyle='-', linewidth=0.1)
  axs[1][g].grid(color='black', linestyle='-', linewidth=0.1)



for k in range (6):
  axs[0][k].xaxis.tick_top()
  #axs[1][k].xaxis.tick_top()
  axs[0][k].yaxis.tick_right()
  axs[1][k].yaxis.tick_right()



for k in range (6):
  for tick in axs[0,k].get_xticklabels():
    tick.set_rotation(90)
  for tick in axs[1,k].get_xticklabels():
    tick.set_rotation(90)
  for tick in axs[0,k].get_yticklabels():
    tick.set_rotation(90)
  for tick in axs[1,k].get_yticklabels():
    tick.set_rotation(90)

figure.subplots_adjust(bottom =0.08,right=0.96,left=0.10,hspace=0.05,top=0.92)

figure.text(0.13, 0.45, 'Energia\nx\nEnergia',rotation=90, ha='center', va='center')
figure.text(0.28, 0.46, 'D.P.\nx\nD.P.',rotation=90, ha='center', va='center')
figure.text(0.43, 0.45, 'Média\nx\nMédia',rotation=90, ha='center', va='center')
figure.text(0.57, 0.45, 'Energia\nx\nMédia',rotation=90, ha='center', va='center')
figure.text(0.72, 0.45, 'Energia\nx\nD.P.',rotation=90, ha='center', va='center')
figure.text(0.87, 0.45, 'D.P.\nx\nMédia',rotation=90, ha='center', va='center')
figure.text(0.02, 0.5, 'COMPARAÇÃO ENTRE PARÂMETROS - PACIENTE 1 ',rotation=90, ha='center', va='center',fontsize= 14,
fontweight="bold")


#   PRIMEIRA LINHA
for i in range(40):
  for j in range (40):
    if i>=j:
      print('11:50')
    else:
      axs[0][0].scatter(energia_inter_p1[i+20],energia_inter_p1[j+20],alpha =0.5 ,marker ="." ,color = 'gray',s=0.001)
```

```
for i in range(18):
  for j in range (18):
    if i>=j:
      print('2')
    else:
      axs[1][0].scatter(energia_pre_p1[i],energia_pre_p1[j],alpha =0.5 ,marker ="." ,color = 'gray',s=0.001)

#  SEGUNDA LINHA

for i in range(40):
  for j in range (40):
    if i>=j:
      print('3')
    else:
      axs[0][1].scatter(std_inter_p1[i+20],std_inter_p1[j+20],alpha =0.5 ,marker ="." ,color = 'gray',s=0.001)

for i in range(18):
  for j in range (18):
    if i>=j:
      print('4')
    else:
      axs[1][1].scatter(std_pre_p1[i],std_pre_p1[j],alpha =0.5 ,marker ="." ,color = 'gray',s=0.001)


#  TERCEIRA LINHA

for i in range(40):
  for j in range (40):
    if i>=j:
      print('5')
    else:
      axs[0][2].scatter(media_inter_p1[i+20],media_inter_p1[j+20],alpha =0.5 ,marker ="." ,color = 'gray',s=0.001)

for i in range(18):
  for j in range (18):
    if i>=j:
      print('6')
    else:
      axs[1][2].scatter(media_pre_p1[i],media_pre_p1[j],alpha =0.5 ,marker ="." ,color = 'gray',s=0.001)
```

```
#   QUARTA LINHA

for i in range(106):
        axs[0][3].scatter(energia_inter_p1[i],media_inter_p1[i],alpha =0.5 ,marker ="." ,color = 'gray',s=0.001)

for j in range(18):
        axs[1][3].scatter(energia_pre_p1[j],media_pre_p1[j],alpha =0.5 ,marker ="." ,color = 'gray',s=0.001)

#  QUINTA LINHA

for i in range(106):
        axs[0][4].scatter(energia_inter_p1[i],std_inter_p1[i],alpha =0.5 ,marker ="." ,color = 'gray',s=0.001)

for j in range(18):
        axs[1][4].scatter(energia_pre_p1[j],std_pre_p1[j],alpha =0.5 ,marker ="." ,color = 'gray',s=0.001)


#   SEXTA LINHA

for i in range(106):
        axs[0][5].scatter(std_inter_p1[i],media_inter_p1[i],alpha =0.5 ,marker ="." ,color = 'gray',s=0.001)

for j in range(18):
        axs[1][5].scatter(std_pre_p1[j],media_pre_p1[j],alpha =0.5 ,marker ="." ,color = 'gray',s=0.001)



axs[1][0].set_xlim([200,0])
axs[0][0].set_xlim([200,0])


axs[1][1].set_xlim([200,0])
axs[0][1].set_xlim([200,0])

axs[1][2].set_xlim([90,-90])
axs[0][2].set_xlim([90,-90])


axs[1][3].set_xlim([200,0])
axs[0][3].set_xlim([200,0])

axs[1][4].set_xlim([200,0])
axs[0][4].set_xlim([200,0])

axs[1][5].set_xlim([200,0])
axs[0][5].set_xlim([200,0])
```

```
axs[1][0].set_ylim([0,200])
axs[0][0].set_ylim([0,200])

axs[1][1].set_ylim([0,200])
axs[0][1].set_ylim([0,200])

axs[1][2].set_ylim([-90,90])
axs[0][2].set_ylim([-90,90])

axs[1][3].set_ylim([-90,90])
axs[0][3].set_ylim([-90,90])

axs[1][4].set_ylim([0,200])
axs[0][4].set_ylim([0,200])

axs[1][5].set_ylim([-90,90])
axs[0][5].set_ylim([-90,90])

for g in range (6):
  axs[0][g].grid(color='black', linestyle='-', linewidth=0.1)
  axs[1][g].grid(color='black', linestyle='-', linewidth=0.1)
```