

An abstract graphic consisting of several thin, multi-colored lines (orange, blue, green, purple, red) that originate from the left side of the slide. These lines curve downwards and then sweep upwards and to the right, creating a sense of motion and flow. They appear to be layered, with some lines in front of others, and they eventually level out towards the right edge of the slide.

Shapelet Transform Accelerator

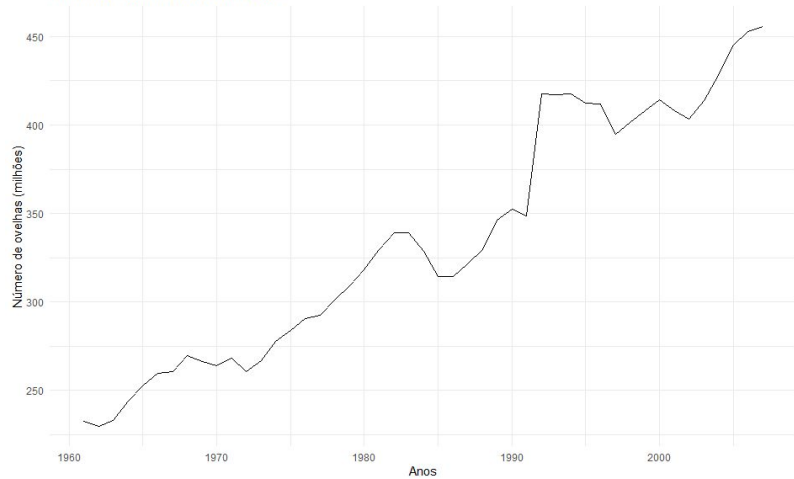
Proposta de ASIC

Mapa da apresentação

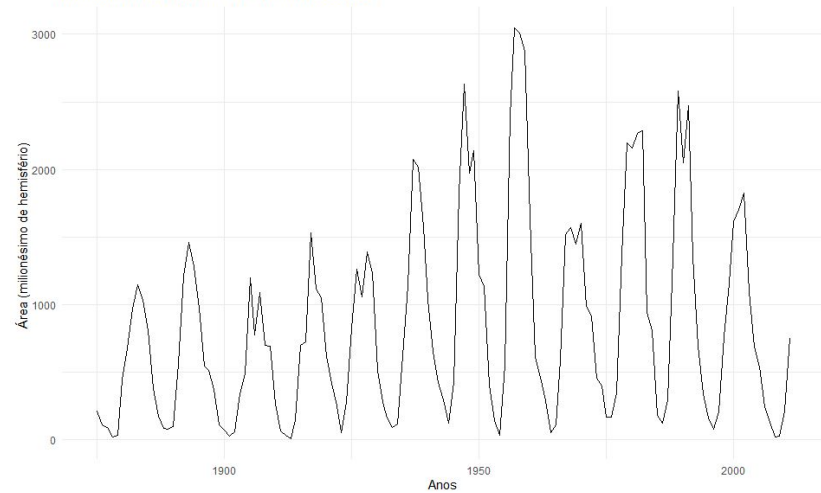
- Séries temporais
- Classificação de séries temporais
 - Divisão em janelas
 - Transformações
- Transformada Shapelet
 - Necessidade de memória
 - Processamento e fluxo de dados
- Acelerando a transformada
 - Paralelismos
 -
- Possibilidades futuras

SÉRIES TEMPORAIS (ST)

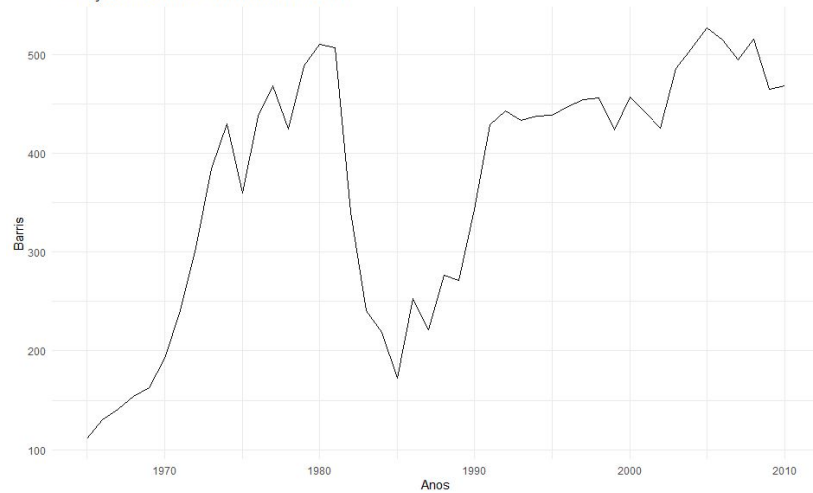
Número anual de ovelhas na Ásia



Média anual do tamanho de manchas solares

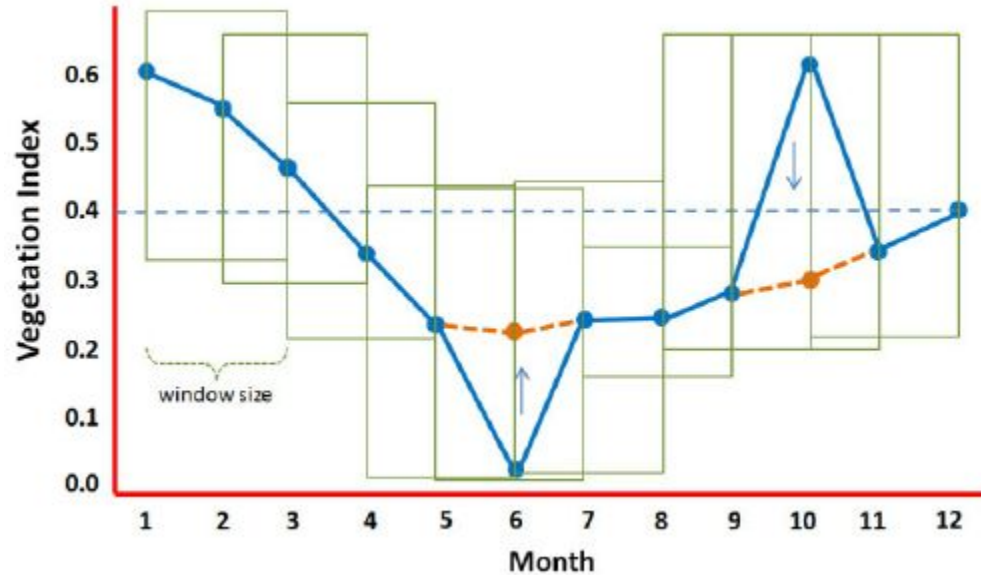


Produção anual de óleo na Arábia Saudita

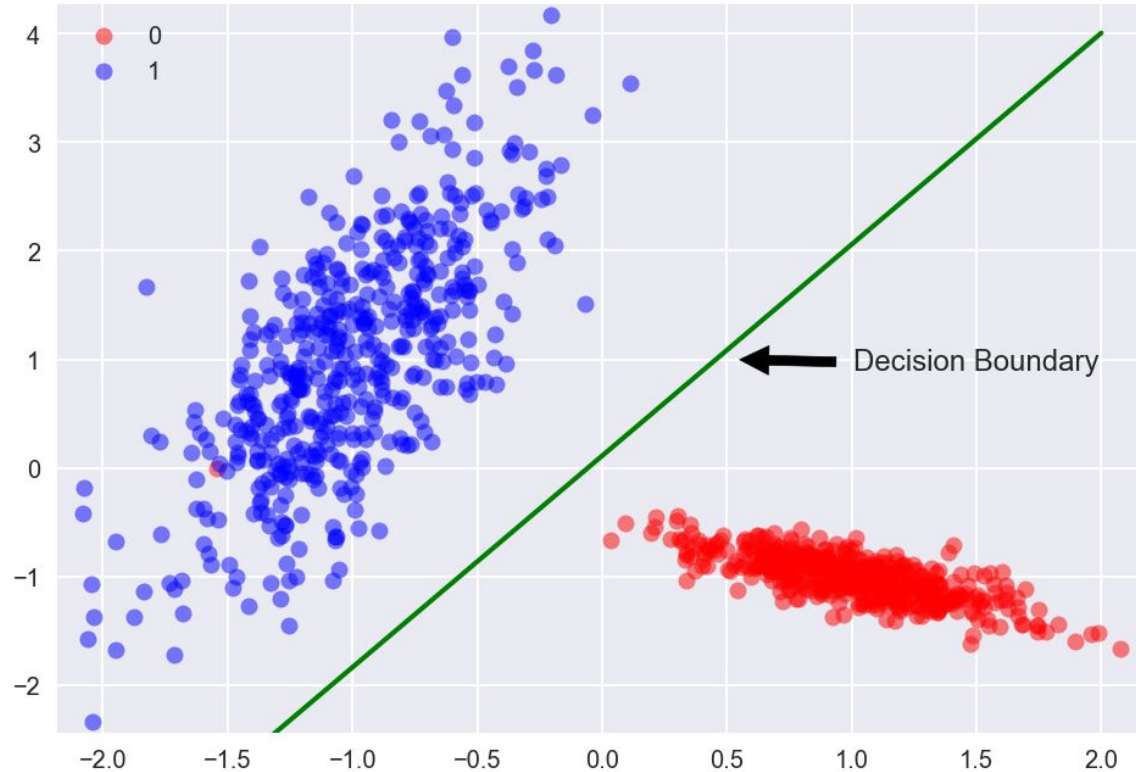


CLASSIFICAÇÃO

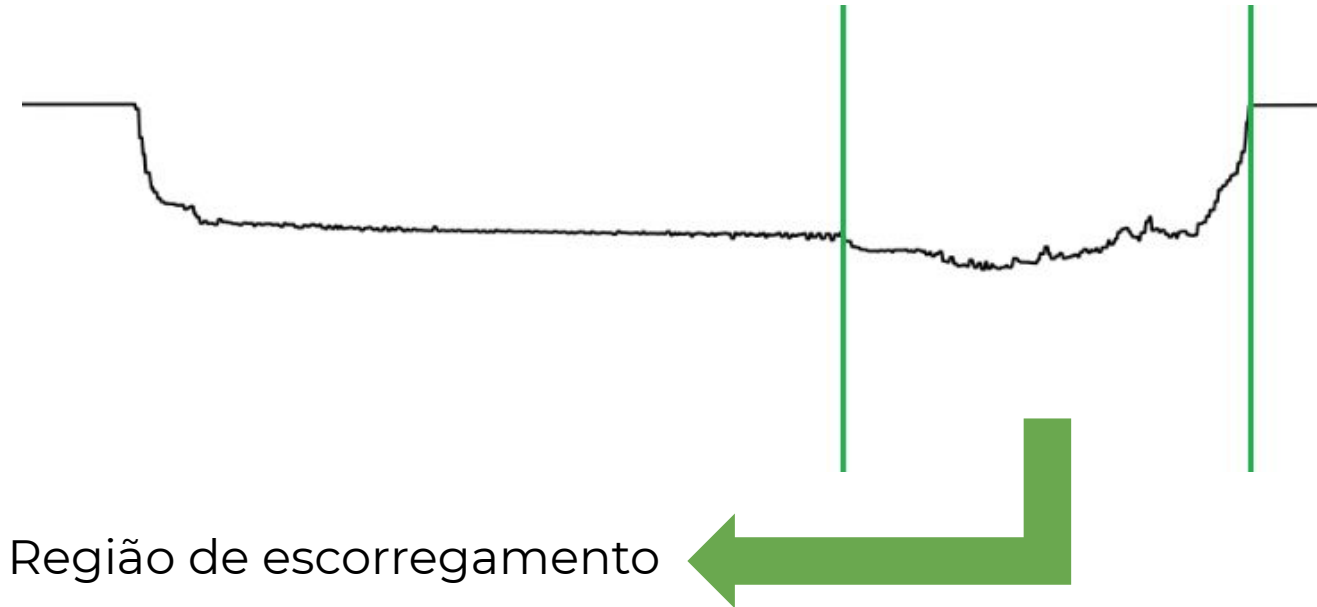
Fluxo básico para classificação de ST



Cada janela é um ponto no espaço

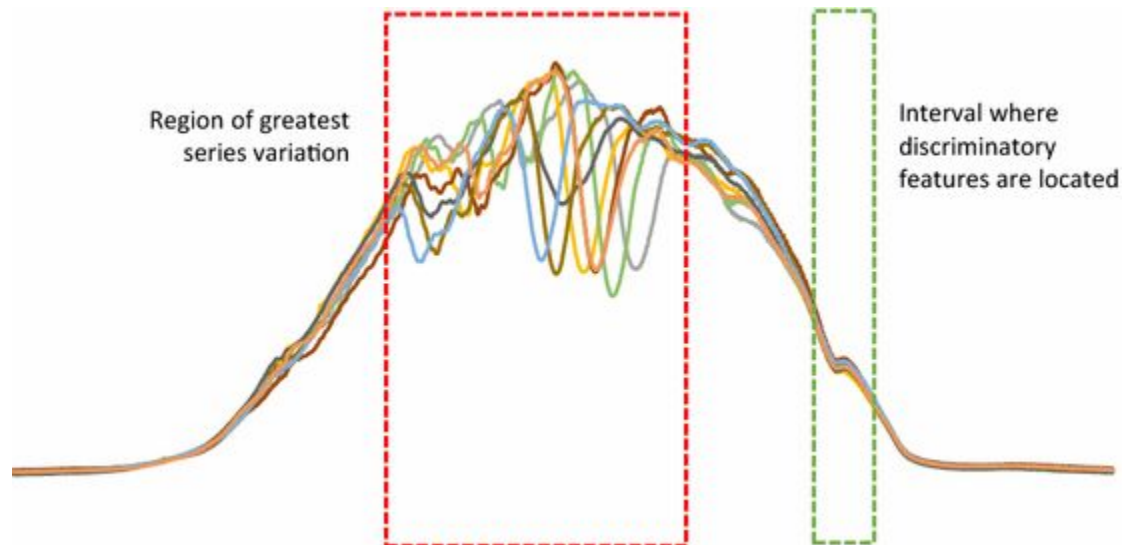


Caso de uso - sinal de sensor FSR

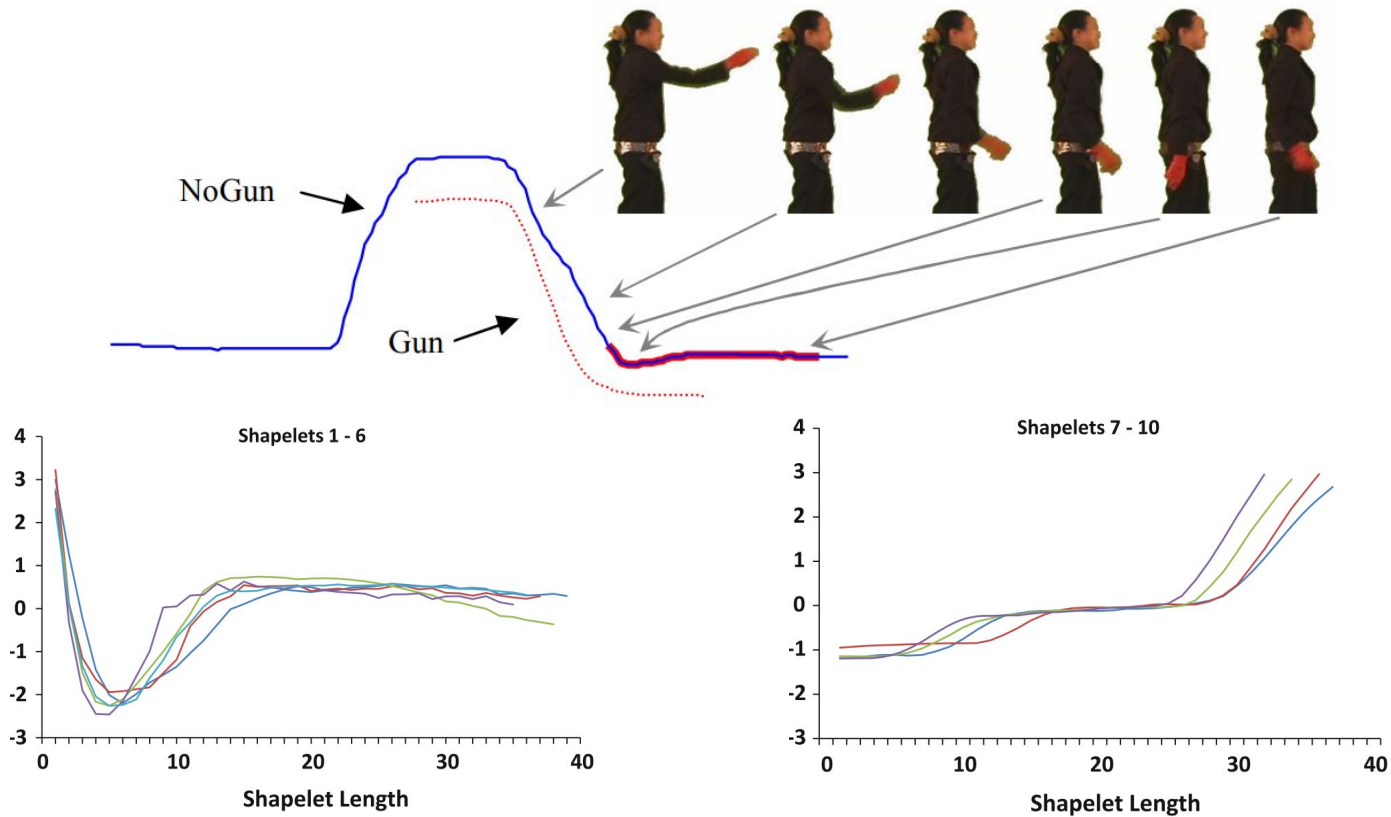


TRANSFORMADA SHAPELET (ST)

Discriminação por formato



Caso de uso - Gun/NoGun



Geração de shapelets

Algorithm 3 ShapeletCachedSelection(\mathbf{T} , min , max , k)

```
1:  $kShapelets \leftarrow \emptyset$ 
2: for all  $T_i$  in  $\mathbf{T}$  do
3:    $shapelets \leftarrow \emptyset$ 
4:   for  $l \leftarrow min$  to  $max$  do
5:      $W_{i,l} \leftarrow generateCandidates(T_i, l)$ 
6:     for all subsequence  $S$  in  $W_{i,l}$  do
7:        $D_S \leftarrow findDistances(S, \mathbf{T})$ 
8:        $quality \leftarrow assessCandidate(S, D_S)$ 
9:        $shapelets.add(S, quality)$ 
10:   $sortByQuality(shapelets)$ 
11:   $removeSelfSimilar(shapelets)$ 
12:   $kShapelets \leftarrow merge(k, kShapelets, shapelets)$ 
13: return  $kShapelets$ 
```

$$\mathbf{T} = \{T_1, T_2, \dots, T_n\}$$

$$T_i = \langle t_{i1}, t_{i2}, \dots, t_{im} \rangle$$

Transformação dos dados

Algorithm 5 ShapeletTransform(Shapelets S , Dataset \mathbf{T})

```
1:  $output \leftarrow \emptyset$ 
2: for all  $T_i$  in  $\mathbf{T}$  do
3:    $transformed \leftarrow \emptyset$ 
4:   for all shapelets  $s$  in  $S$  do
5:      $dist \leftarrow subsequenceDist(s, T_i)$ 
6:      $transformed.add(dist)$ 
7:    $output.add(transformed)$ 
8: return  $output$ 
```

CIRCUITO ACELERADOR

Elementos de hardware

- Lógica de controle para indexação de subsequências por janela
- Somadores, subtratores e potência quadrada para cálculo de distâncias euclidianas
- Comparador para manter apenas menor distância entre cada shapelet e cada T_i .
- Hardware para cálculo de métrica de qualidade
- Ordenador para manter apenas melhores k shapelets

Paralelismos

- No algoritmo original, para uma sub-série T_i , cada shapelet é considerada sequencialmente, resultando em ler todo \mathbf{T} da memória para cada uma das $m - l + 1$ shapelets de tamanho l em T_i .

Caso as shapelets de uma sub-série T_i sejam consideradas simultaneamente, é possível diminuir a quantidade de acessos a \mathbf{T} sem aumentar a necessidade de memória

- Cálculo da qualidade de cada shapelet de forma paralela

Adaptação aos parâmetros

Para que seja aplicável ao mundo real, o circuito deve permitir flexibilidade nos parâmetros do algoritmo, como l_{\min} , l_{\max} , m e n .

Mudanças nesses parâmetros são mudanças nas necessidades em processamento e memória.

A parametrização do hardware deverá obedecer a certas restrições. É desejável o desenvolvimento de um software simulador de recursos computacionais para cada conjunto de parâmetros do algoritmo.

Decisões

- Quais loops do algoritmo serão implementados?
- Qual será a magnitude de paralelismo escolhida?
- Qual medida de qualidade será utilizada?
- Adotar otimizações algorítmicas para diminuir custo computacional? Se sim, quais?

REFERÊNCIAS

Ye, L. and Keogh, E., 2009, "Time series shapelets: a new primitive for data mining". In Proceedings of the 15th ACM SIGKDD, pp. 947-956.

Hills, J., et al., 2014, "Classification of time series by shapelet transformation", Data Mining and Knowledge Discovery, vol. 28, no. 4, pp. 851-881.

K. W. Chang et al., "Efficient pattern based time series classification on gpu". In Proceedings of the 12th IEEE International Conference on Data Mining, 2012, pp. 131-140.

FIM