

SEGTIME : SEGMENTATION PRÉCISE DE SÉRIES TEMPORELLES SANS FENÊTRE GLISSANTE

Vincent Jouve, Romain Lenoir & Sahli Oussama

Introduction:

Les auteurs mettent en avant une nouvelle approche des réseaux de neurones, SegTime, qui segmente les séries chronologiques pour les étiquettes à changement rapide et lent et évite les **fenêtres coulissantes**

Contexte:

Les séries chronologiques sont utilisés dans plusieurs domaines tels que le partitionnement boursier, l'étiquetage des stades de sommeil et la reconnaissance de l'activité humaine

La segmentation, c'est-à-dire la division des séries chronologiques en segments correspondant à des catégories données, est souvent nécessaire. Une approche courante de la segmentation consiste à sous-échantillonner les séries chronologiques à l'aide d'une fenêtre coulissante avec une certaine longueur et une foulée qui se chevauche, pour créer des sous-séquences de longueur fixe, puis à classer ces sous-séquences dans les catégories données.

Cela réduit la segmentation des séries chronologiques à la classification. Cependant, cette approche garantit de ne trouver que des **points d'arrêt approximatifs** : les **points d'arrêt précis** peuvent apparaître en sous-séquences, et donc la précision de la segmentation se dégrade lorsque les étiquettes changent rapidement. En outre, il ignore les dépendances possibles à long terme entre les sous-séquences. Les auteurs proposent une approche des réseaux neuronaux SegTime qui trouve des points d'arrêt précis, évite les fenêtres coulissantes, gère les dépendances à long terme et est insensible à la fréquence de changement d'étiquette. SegTime le fait, grâce à son architecture bi-passe avec plusieurs structures qui peuvent traiter l'information de manière multi-échelle. Nous avons largement évalué l'efficacité de SegTime avec des résultats très prometteurs.

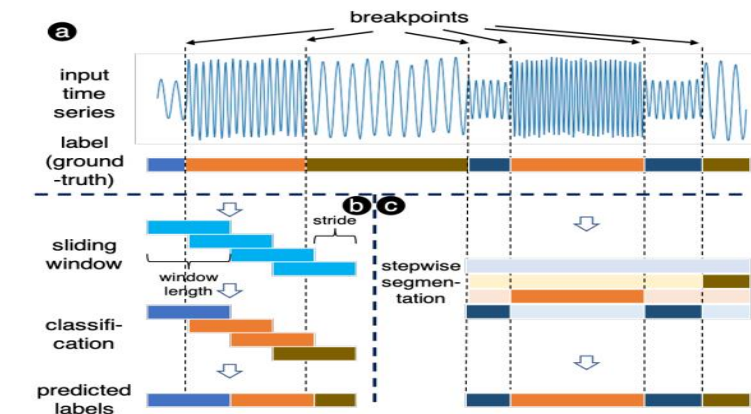


Figure 1: (a) Time series segmentation; (b) sliding window approach; (c) our approach

Segtime

SegTime se compose de **cinq modules** et ils sont organisés dans une architecture bi-passe (Figure 2a). La séquence d'entrée passe par deux passages : un réseau MSSLSTM (Multi-scale skip LSTM) et un module décodeur encodeur 1D. Ce dernier est constitué de trois sous-modules : un 1D-DS-ResNet (depth wise separable), un AMSP (atrous multi-scale pooling) (ces deux-là forment l'encodeur), et le décodeur. Les sorties du module encodeur-décodeur et du réseau MSS-LSTM sont ensuite concaténées. Elles passent ensuite par le module de **segmentation par étapes**, qui peut prédire les étiquettes de sortie au niveau des étapes, réalisant ainsi une **segmentation précise** des séries temporelles.

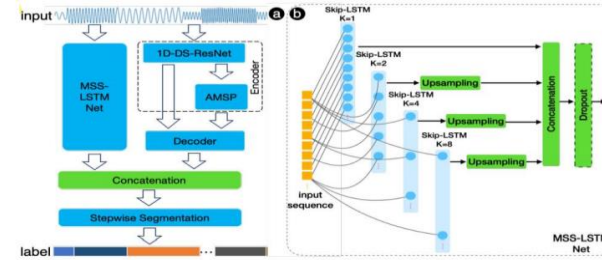


Figure 2 : (a) Architecture de SegTime (une vue d'ensemble étendue est donnée dans la figure 2 de l'annexe) ; (b) Réseau LSTM à saut multi-échelle (MSS-LSTM).

Encodeur-Décodeur

La structure **encodeur-décodeur** (Sutskever et al., 2014) est une approche éprouvée dans divers domaines et connaît un grand succès dans la segmentation sémantique du problème 2D. Les auteurs adoptent un **encodeur-décodeur 1D** (figure 3) avec le **1D-DS-ResNet** et l'**AMSP** comme encodeur et deux couches convolutives comme décodeur (inspiré de Chen et al. (2018)). De l'encodeur, deux niveaux de caractéristiques sortent vers le décodeur : (1) les caractéristiques de bas niveau provenant des premières couches de 1D-DS ResNet ; (2) les caractéristiques de sortie après l'AMSP. Ces deux niveaux de caractéristiques exploitent l'information multi-échelle et contribuent ainsi à la propriété de SegTime de traiter les fréquences multi-échelles des données. Notez que l'encodeur a une architecture très profonde (en 1D-DS-ResNet) alors que le décodeur n'a que quelques couches. Ceci est dû au fait que cette structure peut apprendre la représentation de manière efficace et les compresser dans une couche goulot d'étranglement.

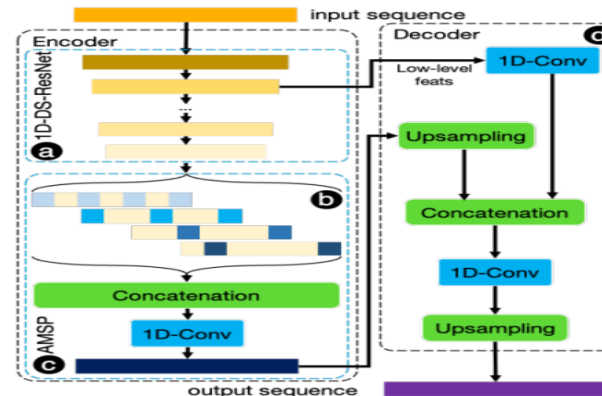


Figure 3 : La structure de l'encodeur-décodeur 1D

Jeu de données et Evaluation

Dataset	#Time steps	Sampling rate	Average segment length	Minimal segment length
Opportunity Drill	114257	30Hz	102	1
Sleep-EDF-39	123723k	100Hz	26045	3000

Résultat sur Opportunity Drill

Methods	Metrics	Opportunity Drill with Background					Opportunity Drill without Background				
		Loc.	Ges.(R)	Obj.(R)	Ges.(B)	Mean _w	Loc.	Ges.(R)	Obj.(R)	Ges.(B)	Mean _w
Deep-Conv-LSTM	Acc	0.84	0.86	0.87	0.88	0.86	0.91	0.81	0.83	0.83	0.85
	F _w	0.80	0.86	0.87	0.87	0.85	0.91	0.84	0.87	0.88	0.88
	IoU _w	0.72	0.77	0.78	0.78	0.76	0.84	0.73	0.79	0.79	0.79
U-Time	Acc	0.89	0.83	0.93	0.79	0.86	0.89	0.75	0.92	0.68	0.81
	F _w	0.89	0.83	0.93	0.78	0.86	0.90	0.78	0.95	0.70	0.83
	IoU _w	0.80	0.73	0.87	0.68	0.77	0.81	0.66	0.91	0.59	0.74
SegTime*	Acc	0.90	0.85	0.90	0.87	0.88	0.89	0.77	0.87	0.84	0.84
	F _w	0.90	0.85	0.90	0.87	0.88	0.90	0.80	0.91	0.87	0.87
	IoU _w	0.81	0.74	0.82	0.77	0.79	0.81	0.68	0.84	0.79	0.78
SegTime	Acc	0.89	0.88	0.88	0.89	0.88	0.88	0.81	0.85	0.86	0.85
	F _w	0.88	0.87	0.88	0.89	0.88	0.89	0.84	0.89	0.91	0.88
	IoU _w	0.80	0.78	0.80	0.81	0.80	0.80	0.74	0.82	0.84	0.80

Résultat sur Sleep-Edf

Dataset	Model	Metrics	W	N1	N2	N3	REM	Mean _w
Sleep-EDF-39	Deep-Conv-LSTM	Acc	0.28	0.00	0.91	0.17	0.14	0.51
		F _w	0.41	0.00	0.68	0.21	0.19	0.44
		IoU _w	0.26	0.00	0.51	0.12	0.10	0.31
	U-Time	Acc	0.90	0.46	0.88	0.83	0.84	0.83
		F _w	0.87	0.52	0.86	0.84	0.84	0.82
		IoU _w	0.77	0.35	0.76	0.73	0.72	0.71
	SegTime	Acc	0.35	0.84	0.92	0.89	0.83	0.82
		F _w	0.87	0.40	0.89	0.82	0.77	0.82
		IoU _w	0.77	0.25	0.79	0.70	0.63	0.71
Sleep-EDF-39	SegTime	Acc	0.31	0.89	0.91	0.86	0.76	0.82
		F _w	0.82	0.38	0.88	0.81	0.81	0.82
		IoU _w	0.70	0.24	0.78	0.68	0.69	0.70

Références:

- The encoder-decoder structure (Sutskever et al., 2014)
- A 1D depthwise separable (DS) convolution (Howard et al., 2017)
- The AMSP (Figure 3c, inspired by Chen et al. (2018))
- <https://anonymous.4open.science/r/SegTime-0546/>