Incremental and Adaptive Feature Exploration over Time Series Stream

Jingwei ZUO, Karine ZEITOUNI, Yehia TAHER

TransiXplore – 11 October 2019

Contents

1. Introduction - Background

Ultimate goal: An Open Source Platform for Massive Time Series (TS) Flow Analysis (*)

- Context: Dynamic Data Source of Time Series
- Requirement: Adaptive Time Series Model Construction

Knowledge Discovery in Time Series (TS)

- Motif Matching
- (Frequent) Pattern Discovery
- Anomaly Detection
- Time Series Classification/Clustering, etc.

Knowledge Discovery in Data Streams (DS) & Challenges[1]

- Infinite Length ☞ *Memory Cost*
- Feature Evolution ☞ *Incrementality of learning model*
- Concept Drift ☞ *Adaptive adjustment of learning model*
- Concept Evolution ☞ *Emergence of new classes*

(*) This work is conducted under StreamOps project funded by DATAIA institute

1. M. M. Masud, Q. Chen, J. Gao, L. Khan, J. Han, B. Thuraisingham, "Classification and Novel Class Detection of Data Streams in a Dynamic Feature Space", ECML-PKDD'10

1. Introduction - Background

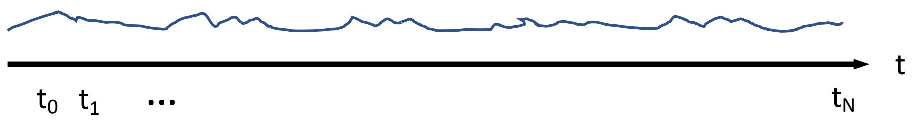
TS + DS =?

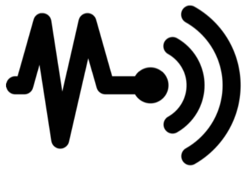
A combination which covers more practical scenarios !

1. Introduction - Definitions

Streaming Time Series S

- A continuous input data stream where each instance is a real-valued data: $S = (t_1, t_2, \ldots, t_N)$, where N is the time tick of the most recent input value.





Smart City Sensor

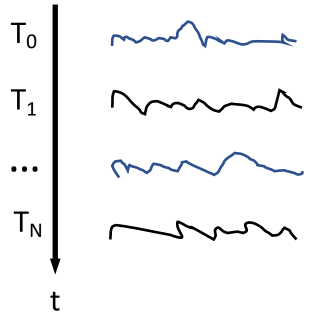
Use Cases:

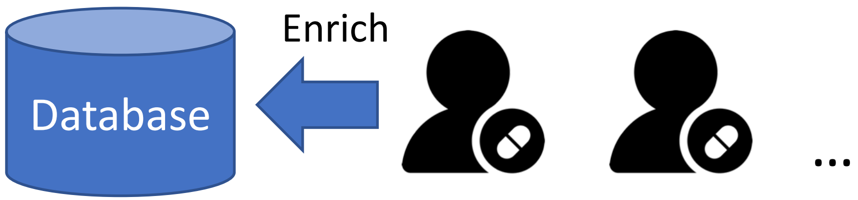
- Online Motif Matching e.g., Human Activity Recognition ☞ Supervised
- Online Frequent Pattern Discovery ☞ Unsupervised
- Online Anomaly/Outlier Detection ☞ (Un)supervised

1. Introduction - Definitions

Time Series Stream S_{TS} (Our context)

- A continuous input data stream where each instance is a Time Series: $S_{TS} = (T_1, T_2, \ldots, T_N)$, notice that N increases with each new time-tick.





Health Care

Use Cases:

- Medical domain
  Patient TS database is getting bigger and bigger
- Astronomy discovery
  New detection of the star light curves, update the features inside the Learning Model

1. Introduction - Research Focus

Time Series Features[1]	
R1	Global features of entire series (1-NN)
R2	Summary statistics of intervals/sub-series
R3	Motif (frequent patterns)
R4	***Shapelet [1] (shape-based features)***
…	

Data Stream Challenges [2]	
C1	***Infinite Length***
C2	***Feature Evolution***
C3	***Concept Drift***
C4	Concept Evolution

new knowledges

R4 + {C1, C2, C3}

1. L. Ye and E. Keogh. "Time series shapelets: A New Primitive for Data Mining." In Proc. SIGKDD 2009
2. M. M. Masud, Q. Chen, J. Gao, L. Khan, J. Han, and B. Thuraisingham, "*Classification and Novel Class Detection of Data Streams in a Dynamic Feature Space*", ECML-PKDD'10

1. Introduction - Cross-domain Collision

Time Series processing

- Real valued data with high temporal dependence

Classic Data Stream processing

- Row or vector data with multiple attributes without temporal dependence

2. Preliminaries

Distance Profile & Matrix Profile[1]

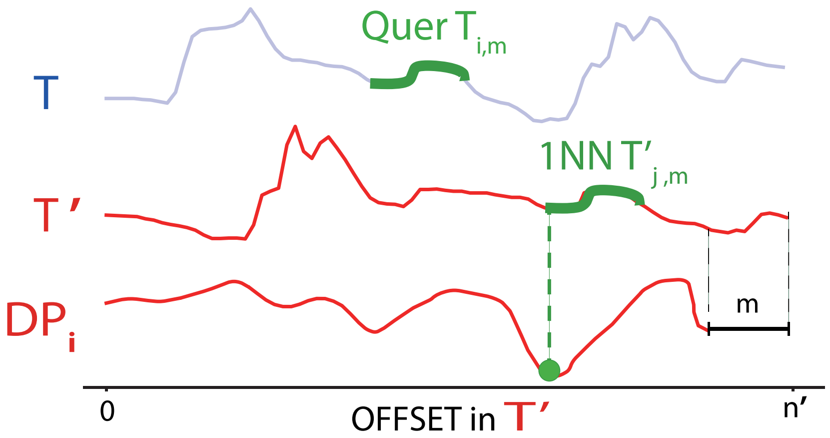


Figure 2.1: ***Distance Profile*** *between Query* $T_{i,m}$ *and target time series* T'*, where* n' *is the length of* T'*.* $DP_{i,j}$ *can be considered as a meta TS annotating target* T'

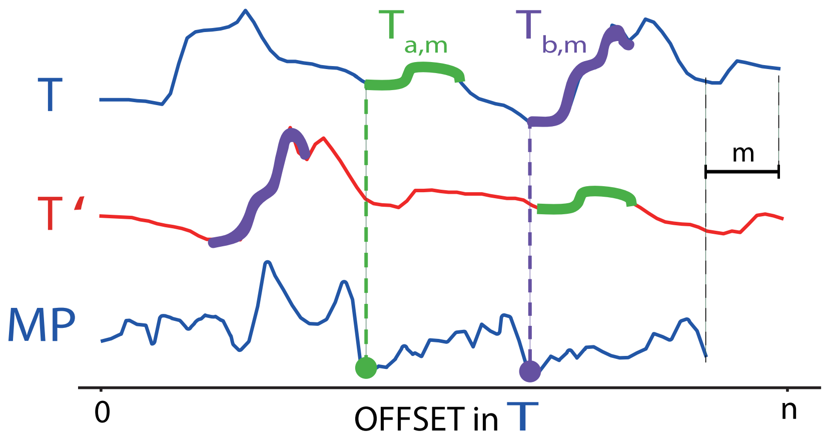


Figure 2.2: ***Matrix Profile*** *between Source* T *and Target* T'*, where* n *is the length of* T*. Intuitively,* MP_i *shares the same offset as source* T

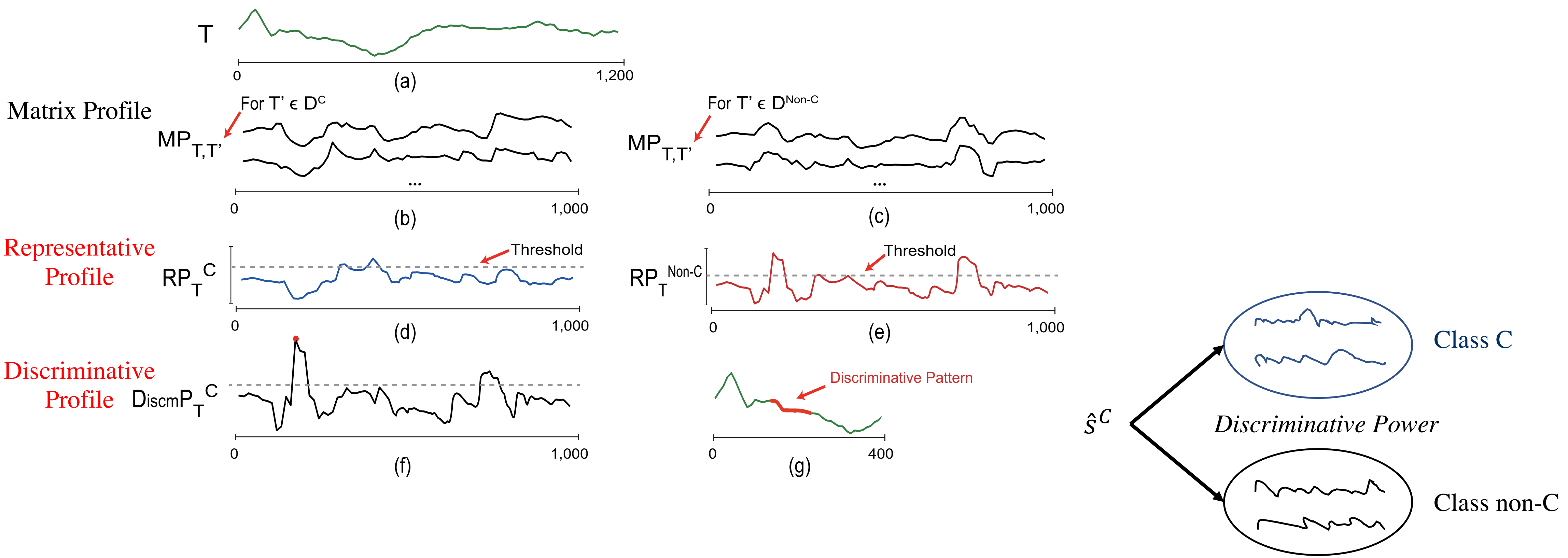
➢ Find the Nearest Neighbor of the Query

➢ Find the closest pairs between two TS

1. Chin-Chia Michael Yeh et al. "*Matrix Pro le I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets*." In Proc. ICDM 2016

2. Preliminaries - Our previous work

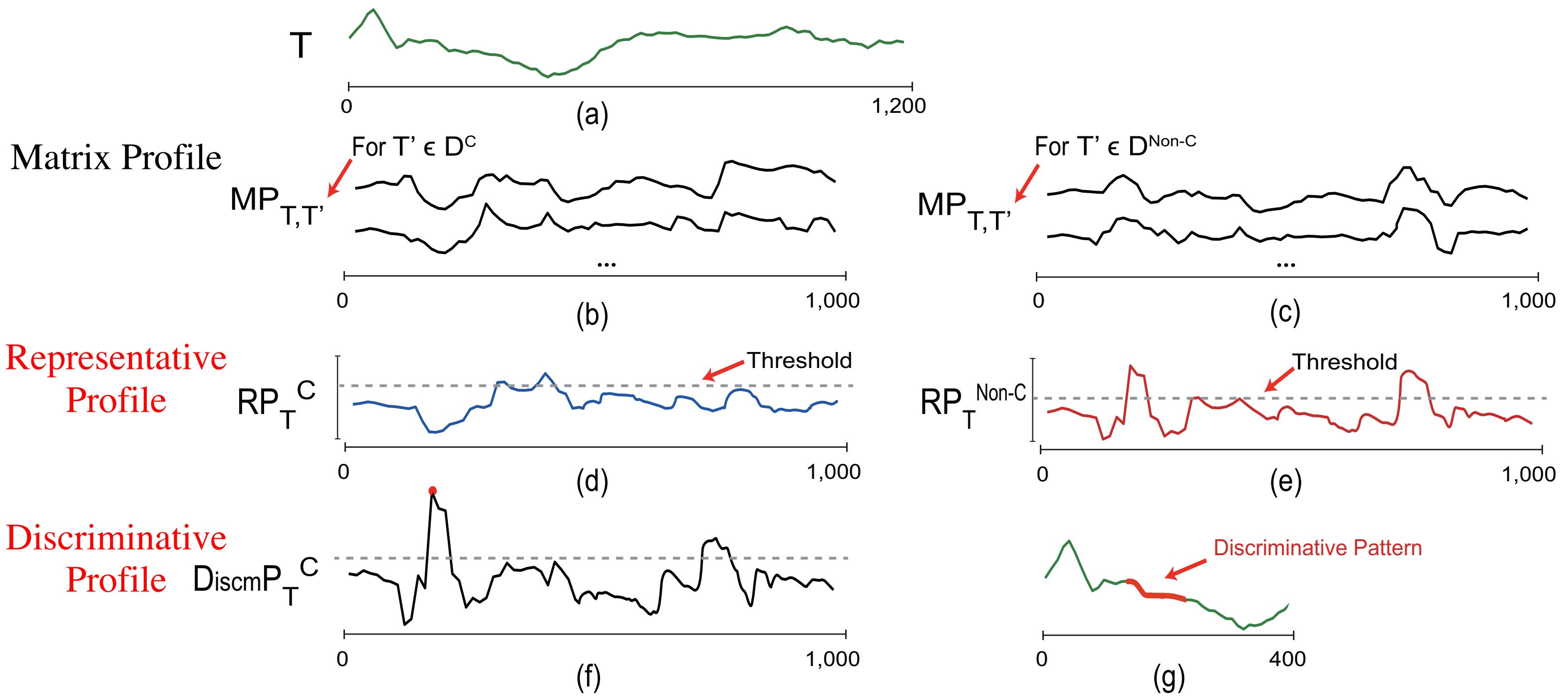
SMAP[1] (Shapelet Extraction on Matrix Profile)



1. J. Zuo, K. Zeitouni, and Y. Taher, "*Exploring interpretable features for large time series with SE4TeC.*" In Proc. EDBT 2019, Lisbon, Portugal.

2. Preliminaries - Our previous work

SMAP [1] (Shapelet Extraction on Matrix Profile)



Cache Dataset in HDFS.

- MapPartition (*Set of <ID, T>*)

  T. $dist_{Thresh} \leftarrow RepresentativeProfile(T, D^C)$

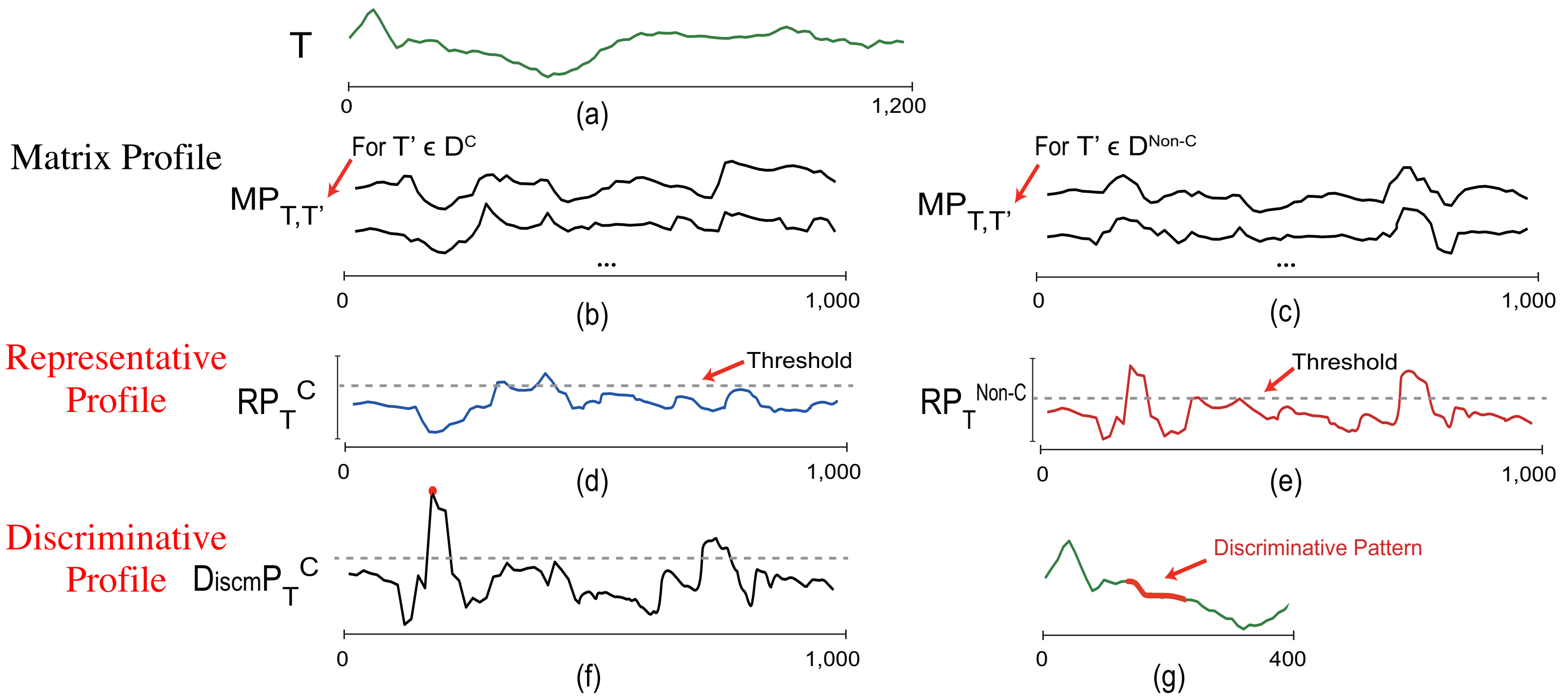
  T. $DiscmP \leftarrow ComputeDiscriminativeProfile(T, D)$

  emit *(ID, T)*

1. J. Zuo, K. Zeitouni, and Y. Taher, "*Exploring interpretable features for large time series with SE4TeC*." In Proc. EDBT 2019, Lisbon, Portugal.

2. Preliminaries - Our previous work

SMAP [1] (Shapelet Extraction on Matrix Profile)



Cache Dataset in HDFS.

- MapPartition (*Set of <ID, T>*)

  $T.\,dist_{Thresh} \leftarrow RepresentativeProfile(T, D^C)$

  $T.\,DiscmP \leftarrow ComputeDiscriminativeProfile(T, D)$

  emit *(ID, T)*

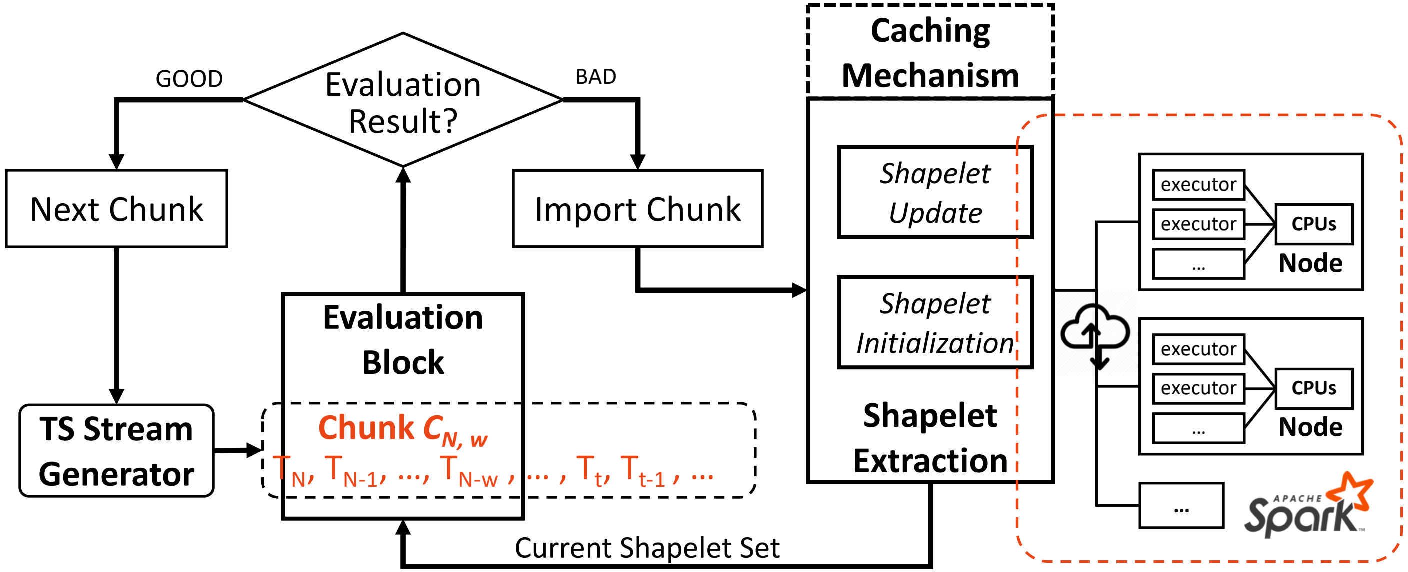
- MapAggregation (*class*, *(ID, T)*)

  $\hat{S} \leftarrow getTopK(aggregation(T.\,DiscmP))$

  return $\hat{S}$

1. J. Zuo, K. Zeitouni, and Y. Taher, "*Exploring interpretable features for large time series with SE4TeC*." In Proc. EDBT 2019, Lisbon, Portugal.

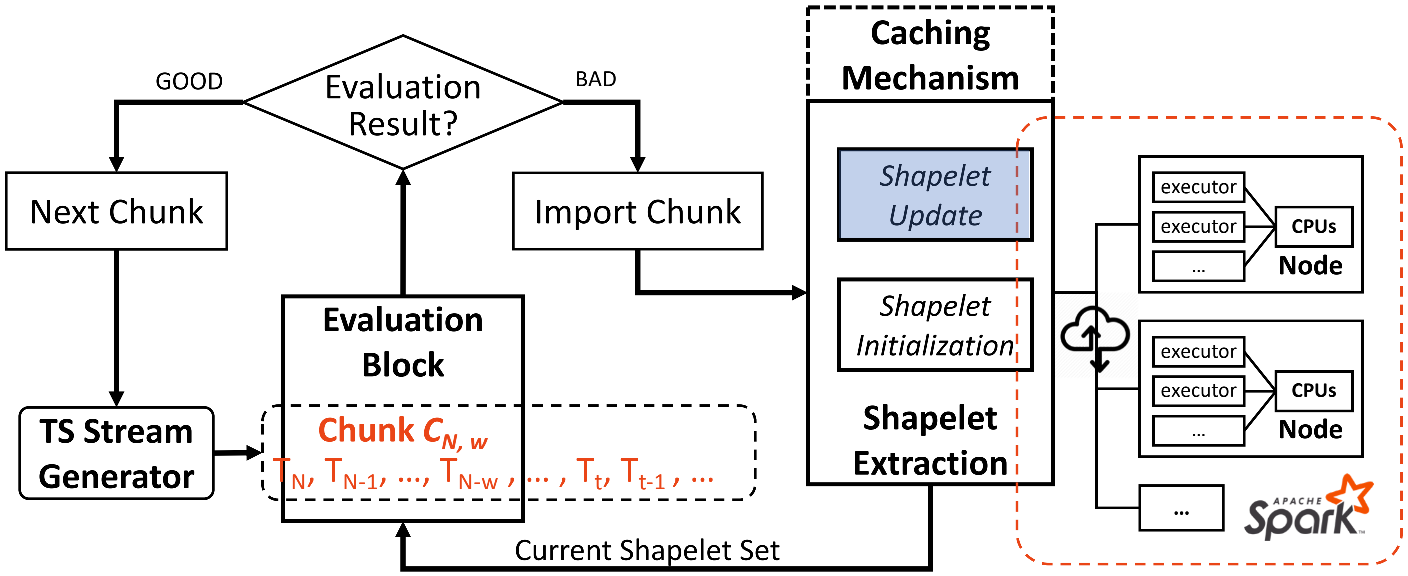
3. Our proposal [1,2]



Test-then-Train strategy

1. J. Zuo, K. Zeitouni, and Y. Taher, "*ISETS: Incremental Shapelet Extraction from Time Series Stream*", demo paper in ECML-PKDD'19 , Würzburg, Germany
2. J. Zuo, K. Zeitouni, and Y. Taher, "*Incremental and Adaptive Feature Exploration over Time Series Stream*", AALTD@ECML-PKDD'19, Würzburg, Germany

3. Our new proposal - Incremental SMAP (ISMAP)



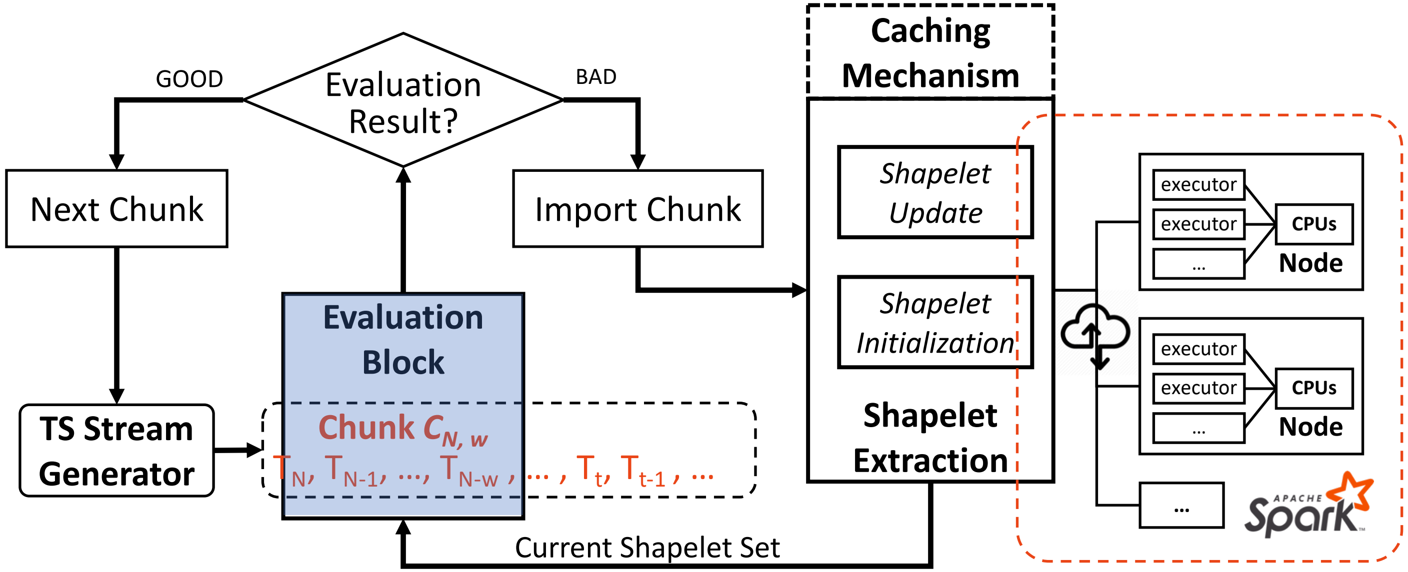
To achieve the Incrementality

When new TS chunk is imported:

1. Update the discriminative power of existing Shapelets
2. Introduce new candidate Shapelets, compute their discriminative power
3. Update the Shapelet Set

Step 1 and 2 share the same computation process

3. Our proposals - Evaluation Block



Evaluation from two aspects

1. Shapelet Evaluation

   ⇢ *Only when loss > threshold, import TS into extraction process*

   ⇢ Select the most informative TS chunks

2. Concept Drift Detection

   ⇢ Distinguish from Shapelet loss

Evaluation Block (Shapelet Evaluation + Concept Drift Detection)

Shapelet Evaluation

- 0-1 Loss Function

$$L(Y, h(T)) = \begin{cases} 0, & Y = h(T) \\ 1, & Y \neq h(T) \end{cases}$$

where

$$h(T) = \begin{cases} C, & if\ dist(T, \hat{s}) \leq \hat{s}.dist_{Thresh} \\ nonC, & otherwise \end{cases}$$

- Sigmoid Loss Function

$$L(Y, h(T)) = \frac{1}{1 + e^{-(x-\sigma)}}, \quad \sigma = \hat{s}.dist_{Thresh}$$

$$x = min(dist(T^C, \hat{s})), \quad \hat{s} \in \hat{S}^C$$

A Loss Threshold Δ can be set to import incrementally the valuable instances.

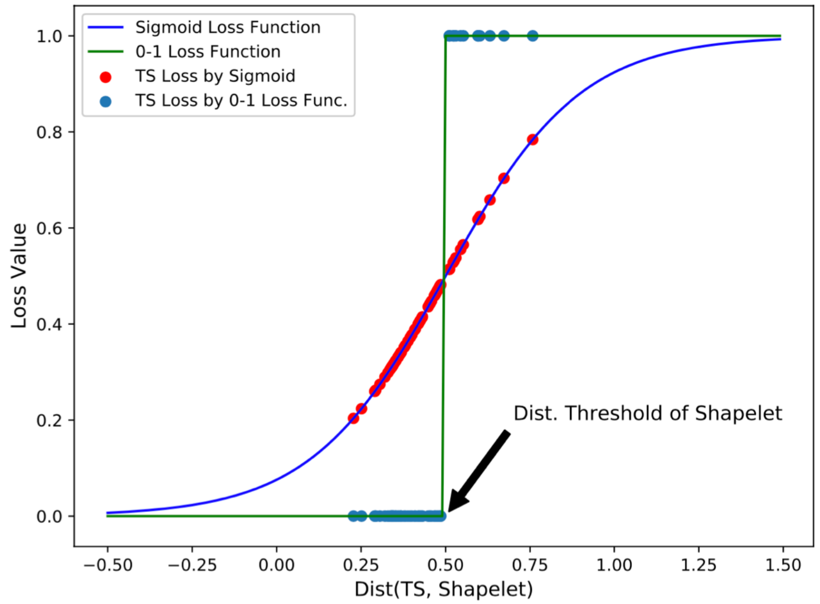


Figure 2: Shapelet Evaluation over newly input TS instances

Evaluation Block (Shapelet Evaluation + Concept Drift Detection)

Concept Drift detection

- Page-Hinkey (PH) Test: a typical technique for change detection in signal processing.

$$L_C(N) = \frac{1}{w}\sum_{k=1}^{w} L(Y_{N-w+k}, h(T_{N-w+k}))$$

$$m_N = \sum_{t=0}^{N}(L_C(t) - L_{avg}(t) - \delta)$$

$$M_N = min(m_t, t = 1...N)$$

$$PH_N = m_N - M_N$$

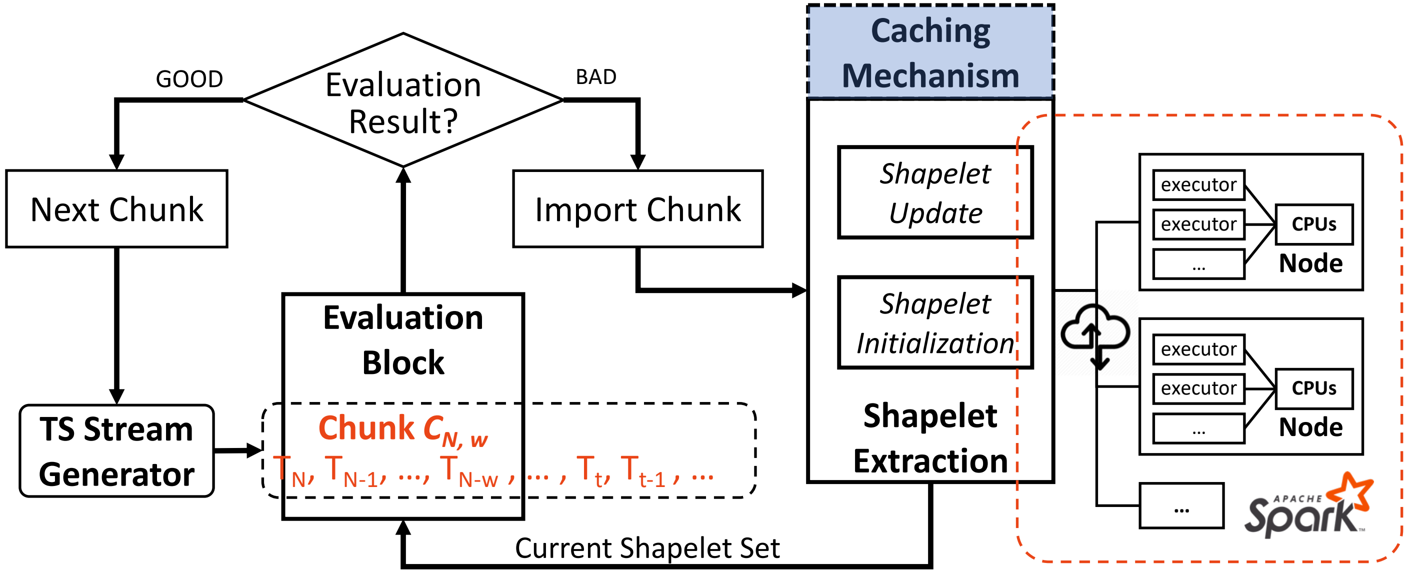
- $L_C(N)$: the average loss of newly input TS chunk
- $L_{avg}(t)$: the average loss of all historical TS chunk until t
- m_N: the cumulative difference between the chunk loss and average loss until the current time. δ: *Loss Tolerance*
- M_N: the minimal cumulative difference recorded

- λ: *PH threshold* to detect a Concept Drift
- $Concept\ Drift = \begin{cases} True, & PH_N \geq \lambda \\ False, & otherwise \end{cases}$

Loss -> Signal

Change point detection

3. Our proposals - Caching Mechanism

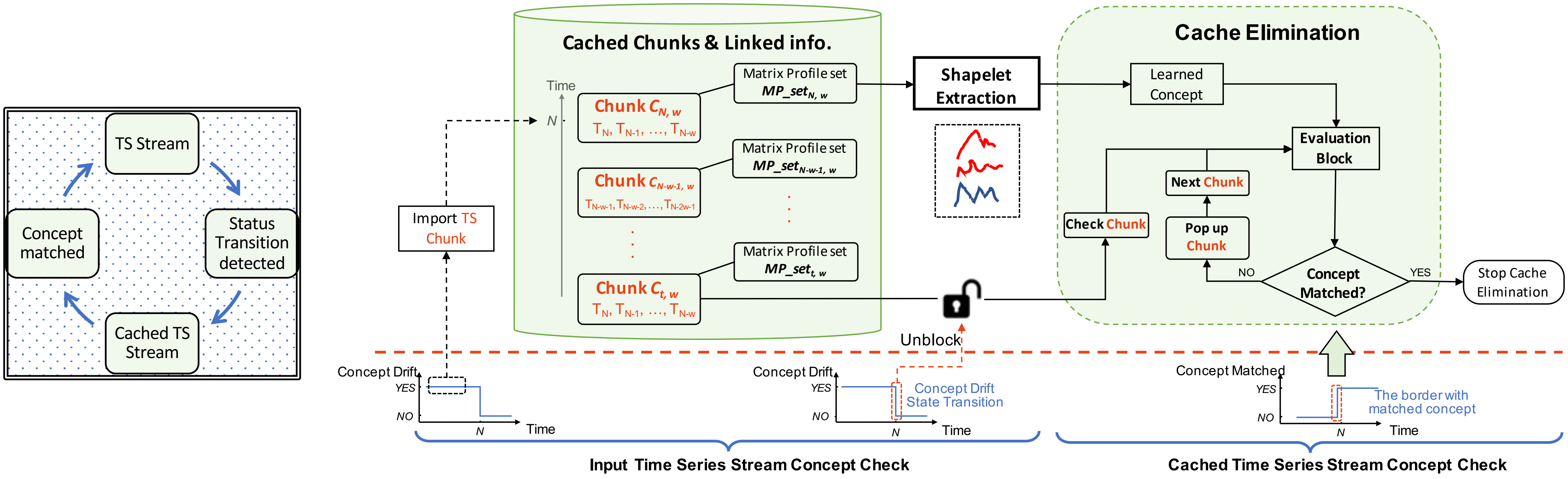


Dependence on cached data

Shapelet Extraction relies on a set of TS instance

- Current Learned concept
- Out-of-date concept

3. System Structure - Elastic Caching Mechanism



Intuition: Fresh learnt concept is inapplicable for the out-of-date instances in the cache

4. Experimental Results

- Incremental test under stable concept (14 Shapelet datasets)

Baseline: Shapelet Tree classifiers

- Information Gain (IG)[1]
- Kruskall-Wallis (KW)[2]
- Mood's Median (MM)[2]

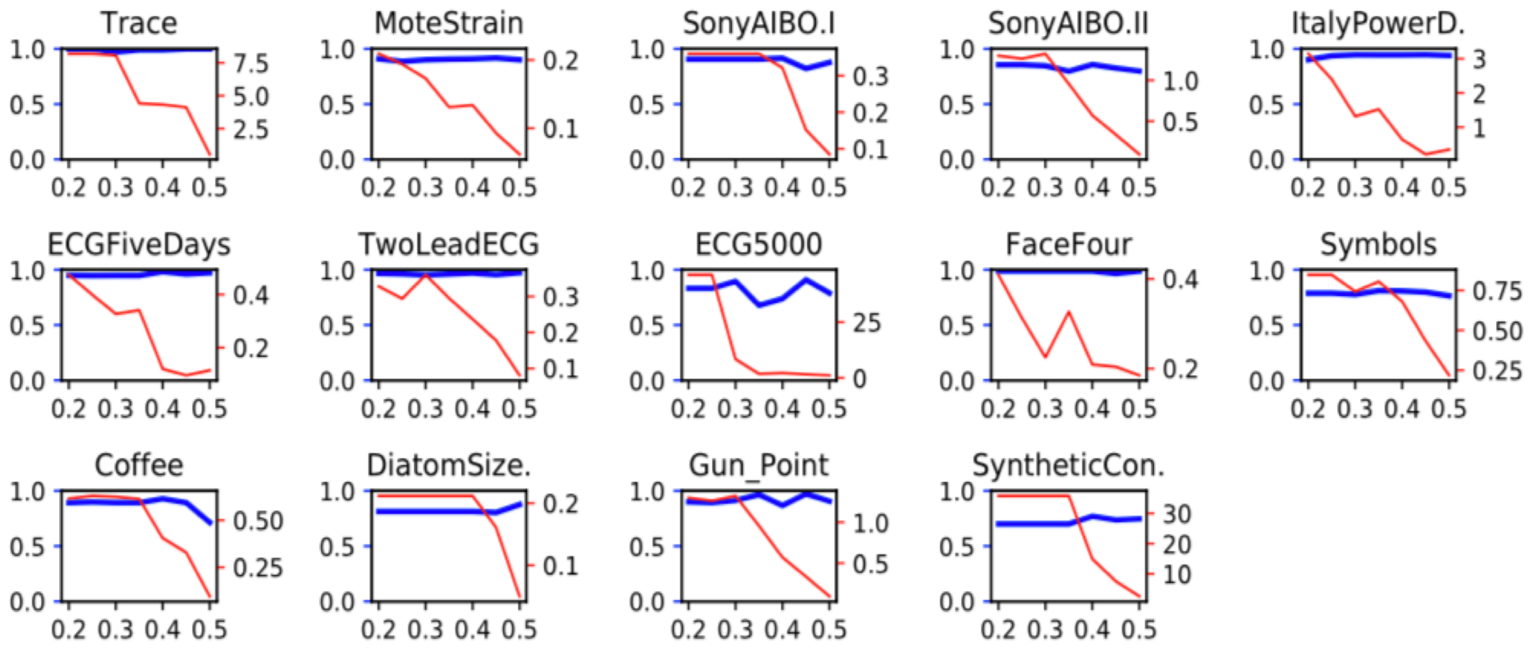
$$Comp.Ratio = \frac{nbr.instance_{imported}}{nbr.instance_{training}}$$

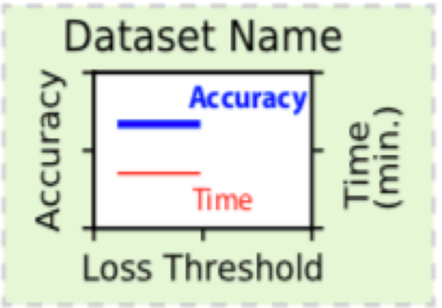
Type	Name	Train/Test	Class	Length	IG	KW	MM	ISMAP(best)	Para. (Δ)	Comp. Ratio
Simulated	SyntheticControl	300/300	6	60	**0.9433**	0.9000	0.8133	0.7007	0.35	46.7%
Sensor	Trace	100/100	4	275	0.9800	0.9400	0.9200	**1**	0.5, 0.45	26.0%
	MoteStrain	20/1252	2	84	0.8251	0.8395	0.8395	**0.9169**	0.45	60.0%
	SonyAIBO.I	20/601	2	70	0.8453	0.7281	0.7521	**0.9151**	0.4	95.0%
	SonyAIBO.II	27/953	2	65	0.8457	-	-	**0.8583**	0.4	63.0%
	ItalyPower.	67/1029	2	24	0.8921	0.9096	0.8678	**0.9466**	0.45	25.4%
ECG	ECG5000	500/4500	5	140	0.7852	-	-	**0.9109**	0.4	9.4%
	ECGFiveDays	23/861	2	136	0.7747	0.8721	0.8432	**0.9826**	0.4	51.2%
	TwoLeadECG	23/1189	2	82	0.8507	0.7538	7657	**0.9337**	0.5	47.8%
Images	Symbols	25/995	6	398	0.7799	0.5568	0.5799	**0.8113**	0.35	96.0%
	Coffee	28/28	2	286	**0.9643**	0.8571	0.8671	0.9286	0.4	78.6%
	FaceFour	24/88	4	350	0.8409	0.4432	0.4205	**0.9886**	except 0.45	62.5%
	DiatomSize.	16/306	4	345	0.7222	0.6111	0.4608	**0.8758**	0.5	50.0%
Motion	GunPoint	50/150	2	150	0.8933	0.9400	0.9000	**0.9733**	0.45	42.0%

1. Lexiang Ye and Eamonn Keogh, "*Time series shapelets: A New Primitive for Data Mining*" In Proc. SIGKDD 2009
2. Jason Lines, and Anthony Bagnall, "*Alternative Quality Measures for Time Series Shapelets*", IDEAL 2012

4. Experimental Results

- Incremental test under stable concept (14 Shapelet datasets)





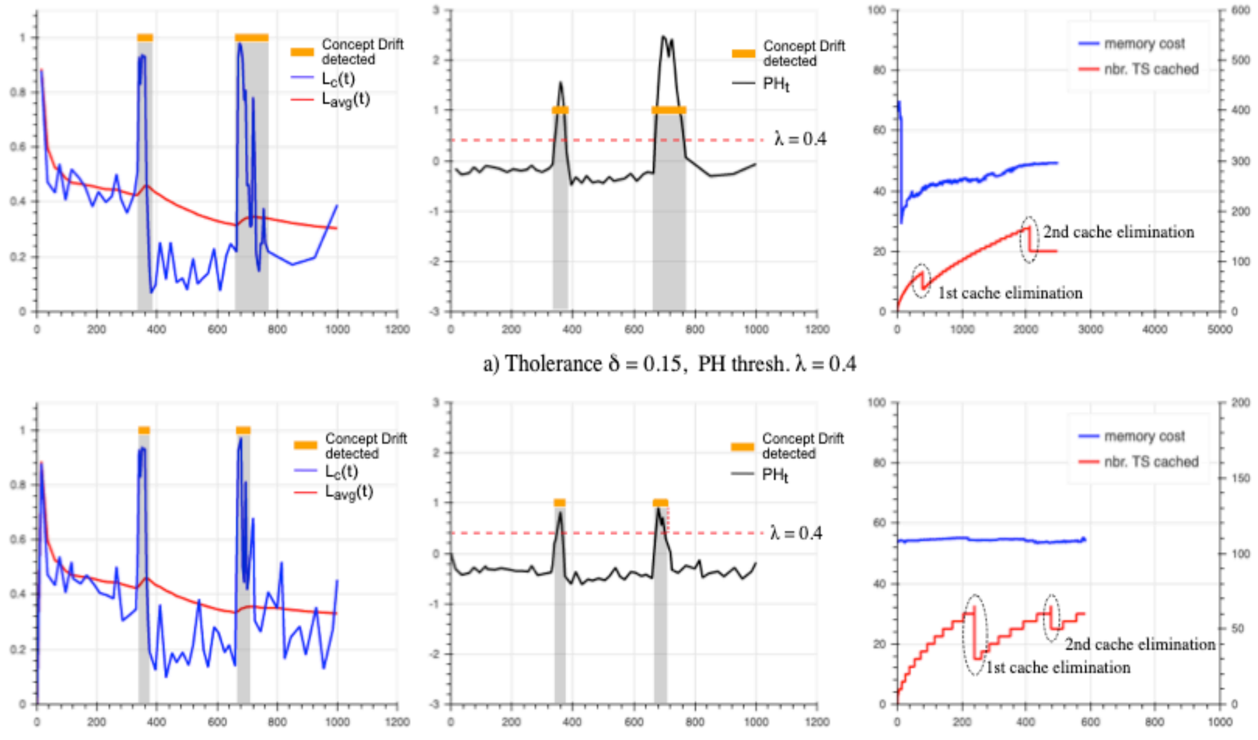
Trade-off between Accu. and Δ

- **In theory**, the higher the loss threshold Δ, the higher the efficiency, the lower the accuracy
- **In practice**, the highest accuracy falls in the range $\Delta \in [0.35, 0.45]$. Nevertheless, efficiency can be greatly increased with an exchange of a negligible decrease of accuracy.

Small uncertainty for the number of instances to be imported into the system

4. Experimental Results

- Concept Drift detection & adaptive features



a) Tholerance $\delta = 0.15$, PH thresh. $\lambda = 0.4$

b) Tolerance $\delta = 0.3$, PH thresh. $\lambda = 0.4$

Synthetic *Trace* dataset:

- Randomly put noise for Data Augmentation
- 1000/1000 training/testing instances
- Two drifts are inserted at time *333* and *667*

Concept Drift detection:

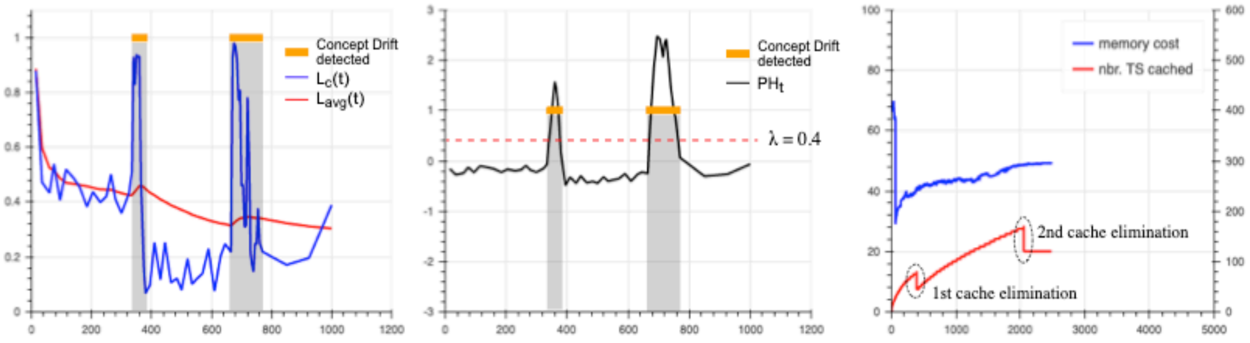
- 345/330, 670/667(δ=0.15); 350/330, 675/667(δ=0.30)

Caching cost:

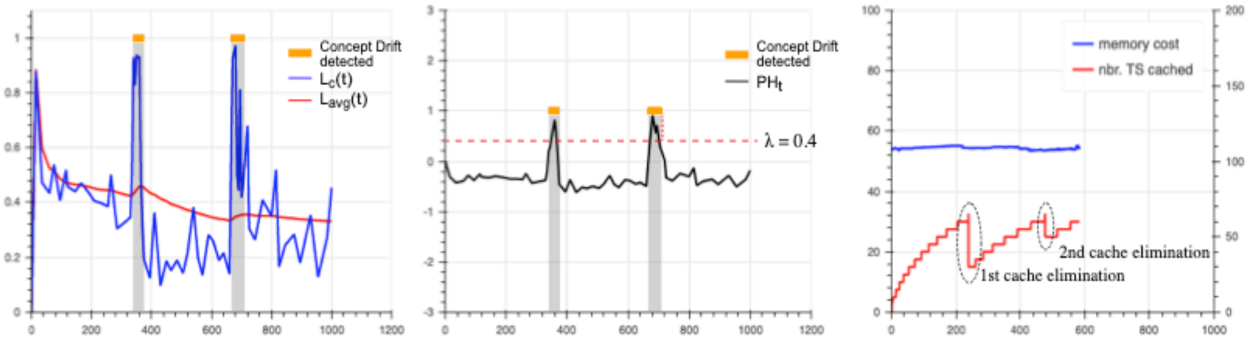
- 100 of 1000 (δ=0.15), 50 of 1000 (δ=0.30)
- Cache is eliminated at the end of drift period

4. Experimental Results

- Concept Drift detection & adaptive features



a) Tholerance $\delta = 0.15$, PH thresh. $\lambda = 0.4$



b) Tolerance $\delta = 0.3$, PH thresh. $\lambda = 0.4$

Synthetic *Trace* dataset:

- Randomly put noise for Data Augmentation
- 1000/1000 training/testing instances
- Two drifts are inserted at time *333* and *667*

TABLE I: Reliability of Extracted Shapelets on 4 time ticks at the beginning/end of each drift area

Dataset	-	**i**(Con. 1)	**ii**(Con. 2)	**iii**(Con. 2)	**iv**(Con. 3)
$Aug.Trace(\delta = 0.15)$	Time tick	*345*	*380*	*670*	*790*
	Test Accu.	0.9600	0.9900	0.9900	0.9800
$Aug.Trace(\delta = 0.30)$	Time tick	*350*	*365*	*675*	*700*
	Test Accu.	0.9600	0.9800	0.9800	0.9700

5. Conclusion

- ✓ First attempt to explore incremental and adaptive features in Time Series Stream.
- ✓ We propose a novel Shapelet Evaluation approach which allows the transition from Time Series to Data Stream analysis.
- ✓ We propose an elastic caching mechanism which is capable of eliminating out-of-date concepts/data proactively in the Time Series Stream model.
- ✓ The system is applicable in the scenario where:
  - New TS instances enrich the learned concept
  - New TS instances bring Concept Drift
- ➢ **Future work:**
  - On both TS Stream and Streaming TS context
  - With a focus on weak-labelled data and active learning

Project page in Github
(Demo video included)

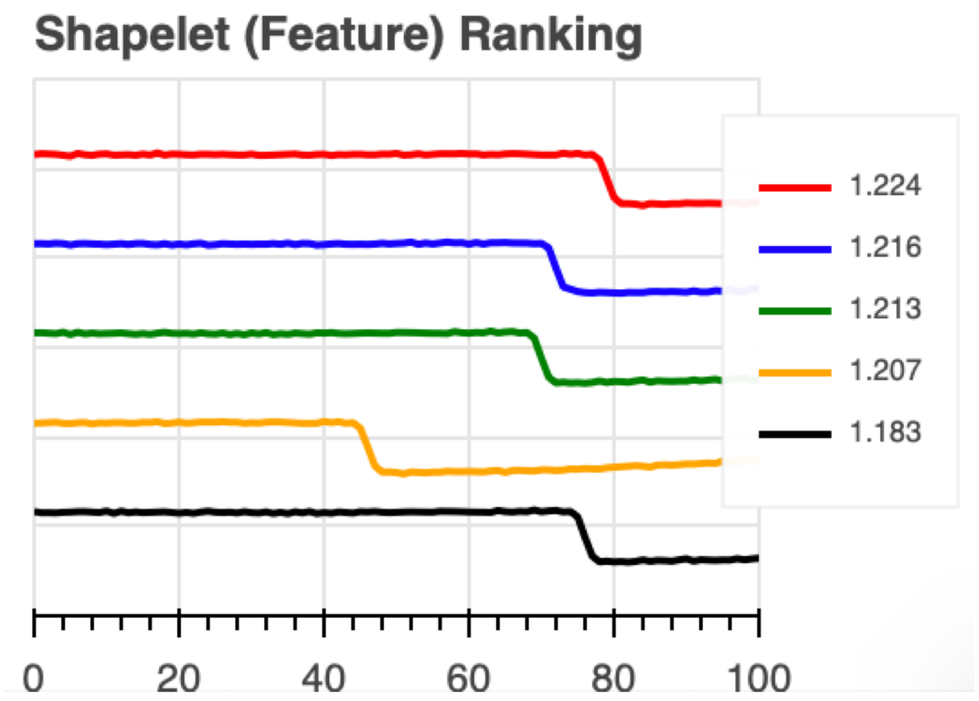
Questions?

Annex 1. Problem Statement

- Low Scalability and Incrementality of Time Series approaches
- Classical Shapelet Evaluation is not suitable in streaming context
- Memory cost of infinite TS instances (Shapelet Extraction relies on a set of instances cached in the memory)
- Concept Drift detection should be adapted in TS Stream model

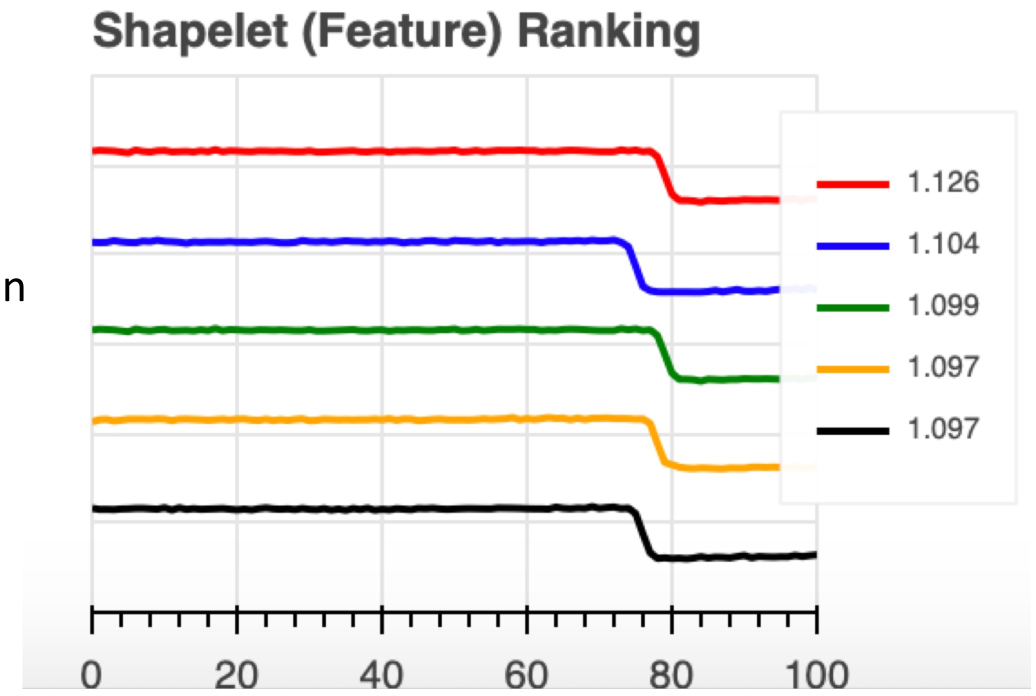
Annex 2. Feature Evolution

Dataset *Trace* (class 2)



Time stamp = 20

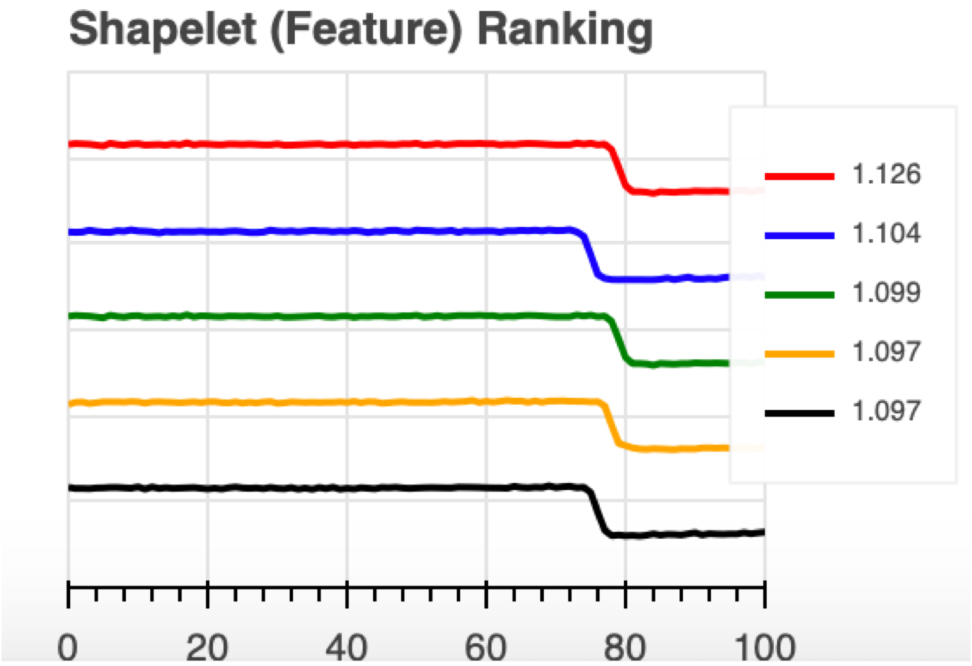
Feature Evolution



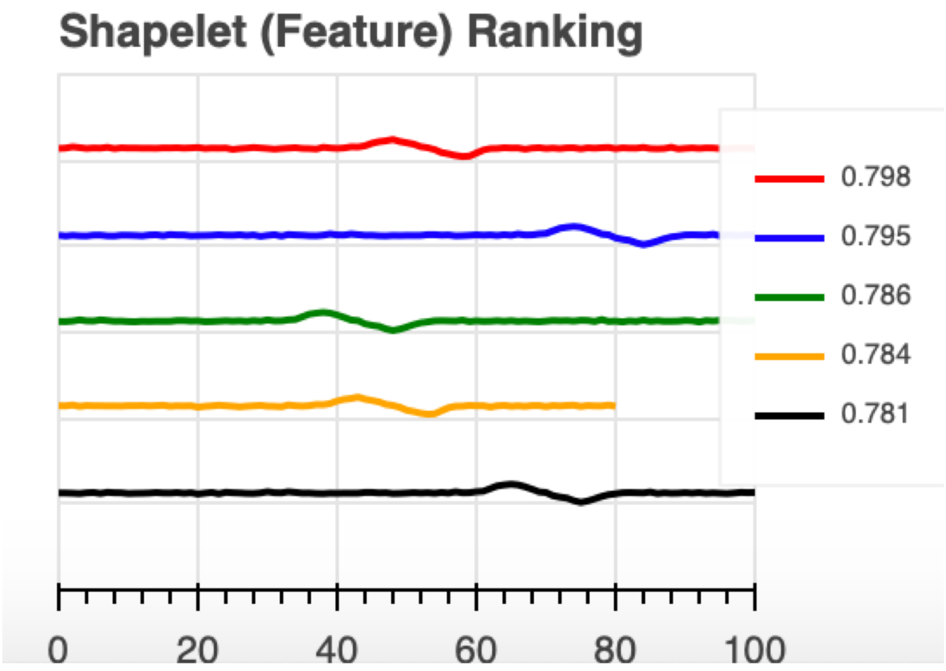
Time stamp = 100

Annex 3– Concept Drift

Dataset *Trace* (class 2)



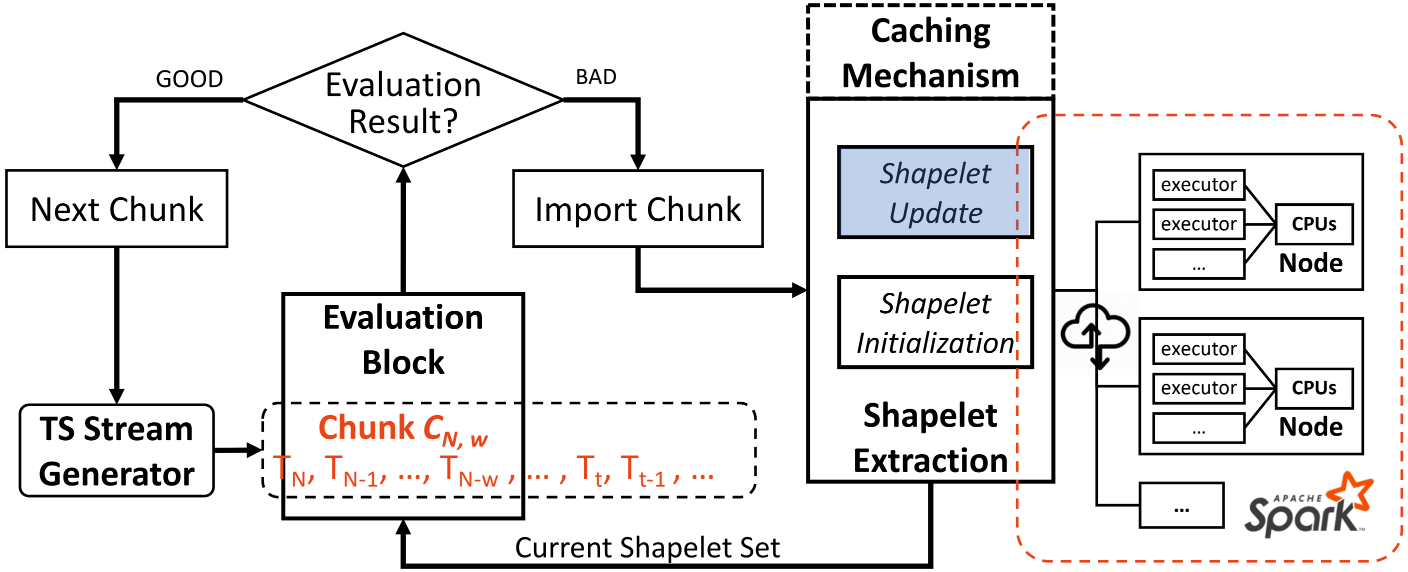
Concept Drift



Time stamp = 100

Time stamp = 200

Annex 4- Incrementality



Cache TS Chunk / T_N in HDFS

1. MapPartition (*Set of (ID, T)*)

computeMP(T, T_N)

$T.dist_{Thresh} \leftarrow UpdateRepresentativeProfile(T, T_N)$

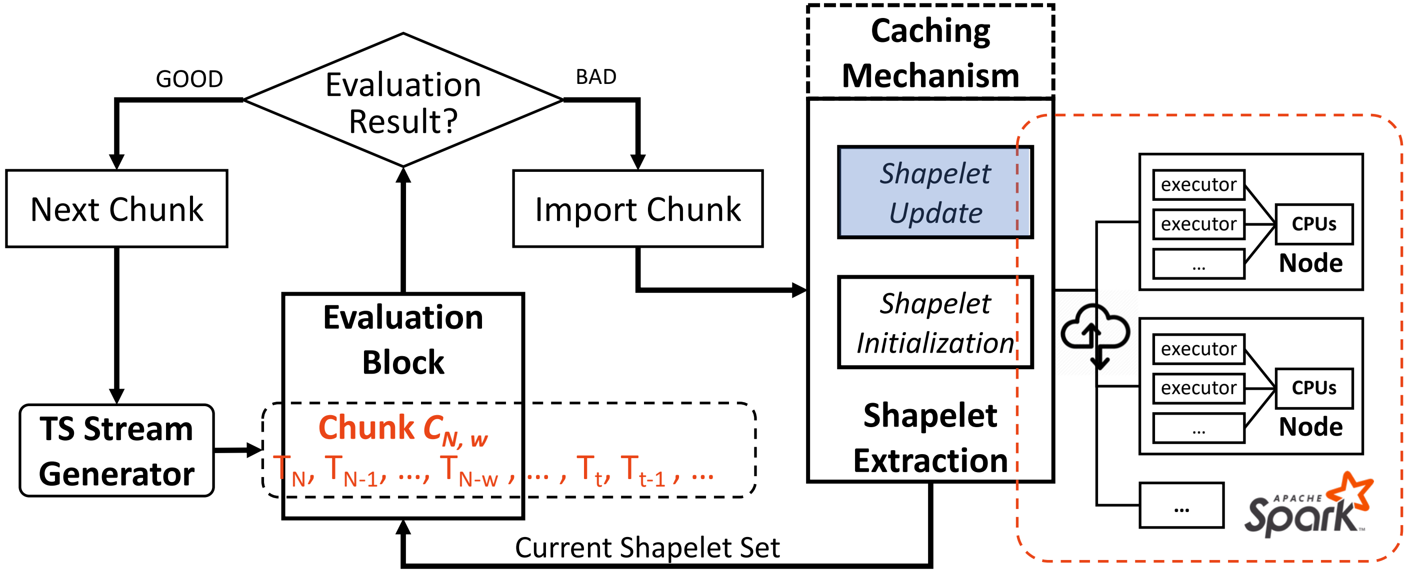
$T.DiscmP \leftarrow UpdateDiscriminativeProfile(T, T_N)$

$MP_{T_N} \leftarrow computeMP(T_N, T)$

emit (ID, T, MP_{T_N})

Update the discriminative power of existing Shapelets

Annex 4 - Incrementality



Cache TS Chunk / T_N in HDFS

1. MapPartition (*Set of (ID, T)*)

   $T.\,dist_{Thresh} \leftarrow UpdateRepresentativeProfile(T, T_N)$

   $T.\,DiscmP \leftarrow UpdateDiscriminativeProfile(T, T_N)$

   $\boxed{MP_{T_N} \leftarrow computeMP(T_N, T)}$

   emit (ID, T, MP_{T_N})

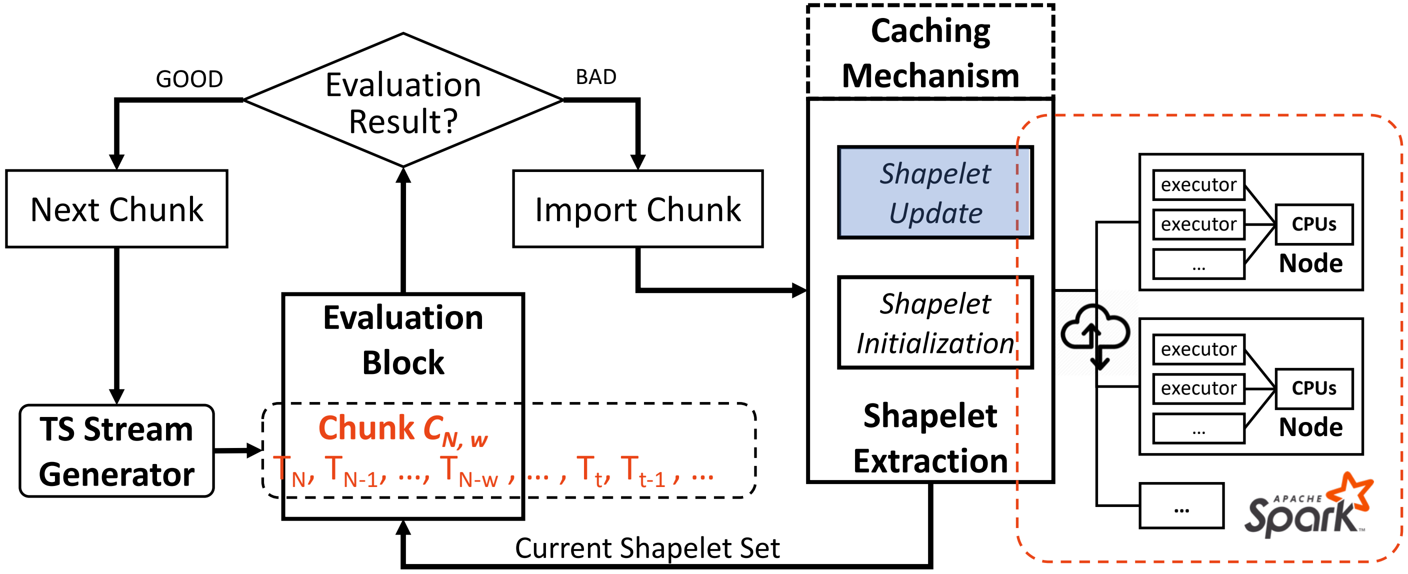
2. MapAggregation $(*, (ID, T, MP_{T_N}))$

   $\boxed{\begin{array}{l} T_N.\,dist_{Thresh} \leftarrow RepresentativeProfile(agg(MP_{T_N})) \\ T_N.\,DiscmP \leftarrow DiscriminativeProfile(agg(MP_{T_N})) \end{array}}$

   return(ID, T_N)

Introduce new candidate Shapelets,
compute their discriminative power

Annex 4 - Incrementality



Cache TS Chunk / T_N in HDFS

1. MapPartition (*Set of (ID, T)*)

   $T.dist_{Thresh} \leftarrow UpdateRepresentativeProfile(T, T_N)$

   $T.DiscmP \leftarrow UpdateDiscriminativeProfile(T, T_N)$

   $MP_{T_N} \leftarrow computeMP(T_N, T)$

   emit (ID, T, MP_{T_N})

2. MapAggregation $(*, (ID, T, MP_{T_N}))$

   $T_N.dist_{Thresh} \leftarrow RepresentativeProfile(agg(MP_{T_N}))$

   $T_N.DiscmP \leftarrow DiscriminativeProfile(agg(MP_{T_N}))$

   return(ID, T_N)

3. MapAggregation $(class, (ID, T))$

   $\hat{S} \leftarrow getTopK(aggregation(T.DiscmP))$

   return $\hat{S}$

Update the Shapelet Set