Towards Robustness to Different Types of Concept Drift in Data Stream Learning

Leandro L. Minku

UNIVERSITY OF BIRMINGHAM

EPSRC

SPDISC Project

Outline

- Motivation and definitions.
    - Example applications.
    - Data stream.
    - Online learning.
    - Concept drift.
    - Types of concept drift.
- A diversity framework based on clustering in the model space.
    - Diversity.
    - Using diversity as a memory strategy.
    - Clustering in the model space to compose a more robust ensemble.
    - Evaluation.
- Other challenges in learning under concept drift.
    - Partially labelled data.
    - Verification latency.
    - Lack of data from target domain.
    - Novel regions of the input space.
    - Class evolution.
    - Class imbalance.

Arctic Ocean
Beaufort Sea
Alaska (US)
Greenland (Dk.)
Greenland Sea
Barents Sea
Norwegian Sea
Iceland
Sweden
Finland
Norway
Russia
Sea of Okhotsk
Canada
Ukraine
Kazakhstan
Mongolia
China
ATLANTIC
United States
Spain
Portugal
Turkey
Iran
Afghanistan
Pakistan
India
East China Sea
Mexico
OCEAN
Morocco
Algeria
Libya
Egypt
Saudi Arabia
Oman
Yemen
Arabian Sea
Mauritania
Mali
Niger
Chad
Sudan
Caribbean Sea
Cameroon
Ethiopia
Somalia
Colombia
Venezuela
Philippines
Democratic Rep. of the Congo
Congo
Kenya
Tanzania
Angola
Zambia
Namibia
Botswana
Mozambique
Madagascar
South Africa
Brazil
Peru
Bolivia
Paraguay
Argentina
Uruguay
Chile
INDIAN
OCEAN
ATLANTIC
OCEAN
PACIFIC
OCEAN
Australia

1.15 million inhabitants; 2nd largest city in the UK.

Founded in 1825.

My Lab

Software Engineering

- Software Defect Prediction
- Software Effort Estimation
- Software Project Scheduling

Artificial Intelligence

- Learning in Dynamic Environments
- Class Imbalance Learning
- Transfer Learning
- Ensemble Learning
- Evolutionary Algorithms

Data Stream Learning of Classification Problems

Data stream: ordered and potentially infinite sequence of examples $\mathcal{S} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \cdots\}$ where $(\mathbf{x}^{(t)}, y^{(t)}) \in \mathcal{X} \times \mathcal{Y}$, $(\mathbf{x}^{(t)}, y^{(t)}) \sim p^{(t)}(\mathbf{x}, y)$.

x = {demand, day of week, etc}, y =

x = {td-idf word 1, td-idf word 2, etc}, y = {topic}

x = {sa

default}

Online Supervised Learning of Classification Problems

- Given a model $f^{(t-1)} : \mathscr{X} \to \mathscr{Y}$ and a new example $(\mathbf{x}^{(t)}, y^{(t)})$ from a probability distribution $p^{(t)}(\mathbf{x}, y)$.

- Learn a model $f^{(t)} : \mathscr{X} \to \mathscr{Y}$ able to generalise to unseen examples of the distribution $p^{(t)}(\mathbf{x}, y)$.

- Strict online learning: only $f^{(t-1)}$ and $(\mathbf{x}^{(t)}, y^{(t)})$ are available for learning.

- Non-strict scenarios: $f^{(t-1)}$, $(\mathbf{x}^{(t)}, y^{(t)})$ and a limited number of past examples may be available.

Concept Drift

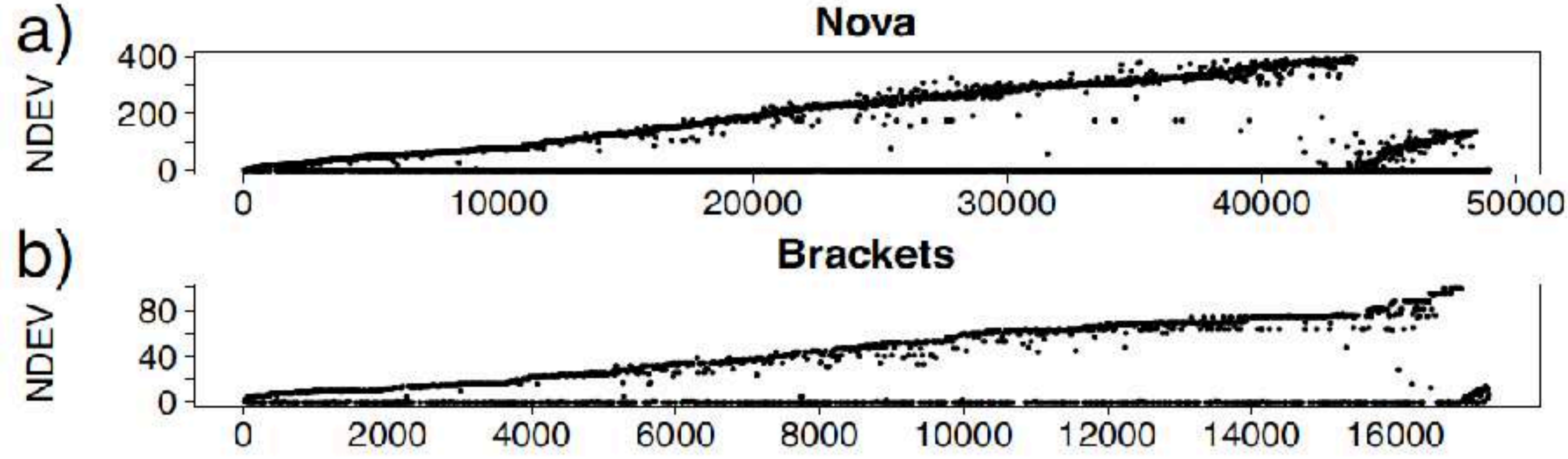
- Concept drift: a change in the joint probability distribution
    - $p^{(t)}(\mathbf{x}, y) = p^{(t)}(y \mid \mathbf{x}) p^{(t)}(\mathbf{x})$

E.g.: credit card approval being affected by economic crises.

Concept Drift

- Concept drift: a change in the joint probability distribution
  - $p^{(t)}(\mathbf{x}, y) = p^{(t)}(y \mid \mathbf{x}) p^{(t)}(\mathbf{x})$

Number of developers that changed the modified files (NDEV) over time



Characterisation of Concept Drifts

- Severity or magnitude: size of the changes caused by the concept drift.
  - Proportion of the input space that has its target class changed.

    $$\int I(p^{(t)}(y\,|\,\mathbf{x}) \neq_e p^{(t')}(y\,|\,\mathbf{x}))\; d\mathbf{x}$$

    L. Minku, A. White and X. Yao. The Impact of Diversity on On-line Ensemble Learning in the Presence of Concept Drift, IEEE Transactions on Knowledge and Data Engineering 22(5):730-742, 2010.

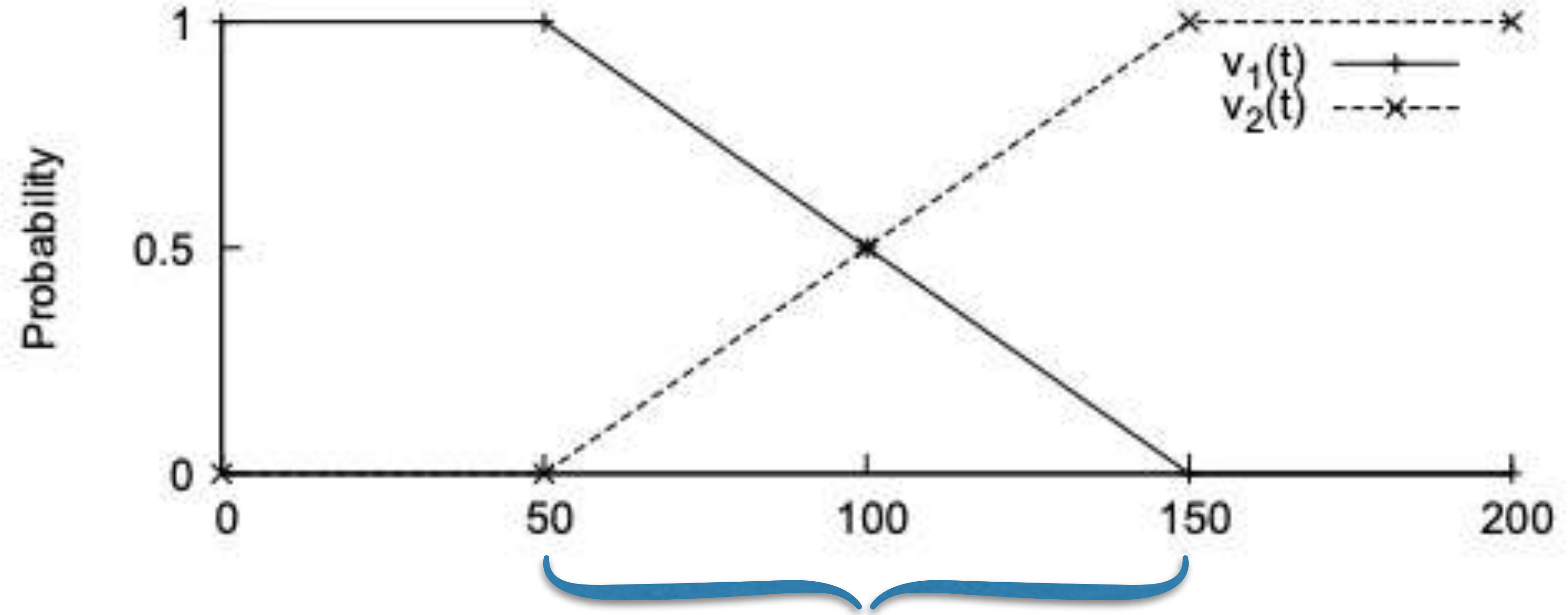
  - Distance between distributions, e.g., Hellinger distance.

    $$H^2(p^{(t)}, p^{(t')})$$

    G. Webb et al. Characterising Concept Drift. Data Mining and Knowledge Discovery 30:964-994, 2016.

Characterisation of Concept Drifts

- Speed: how fast the new probability distribution replaces the past one.
  - Abrupt drifts: replace current distribution suddenly.
  - Gradual: replace current distribution slowly.

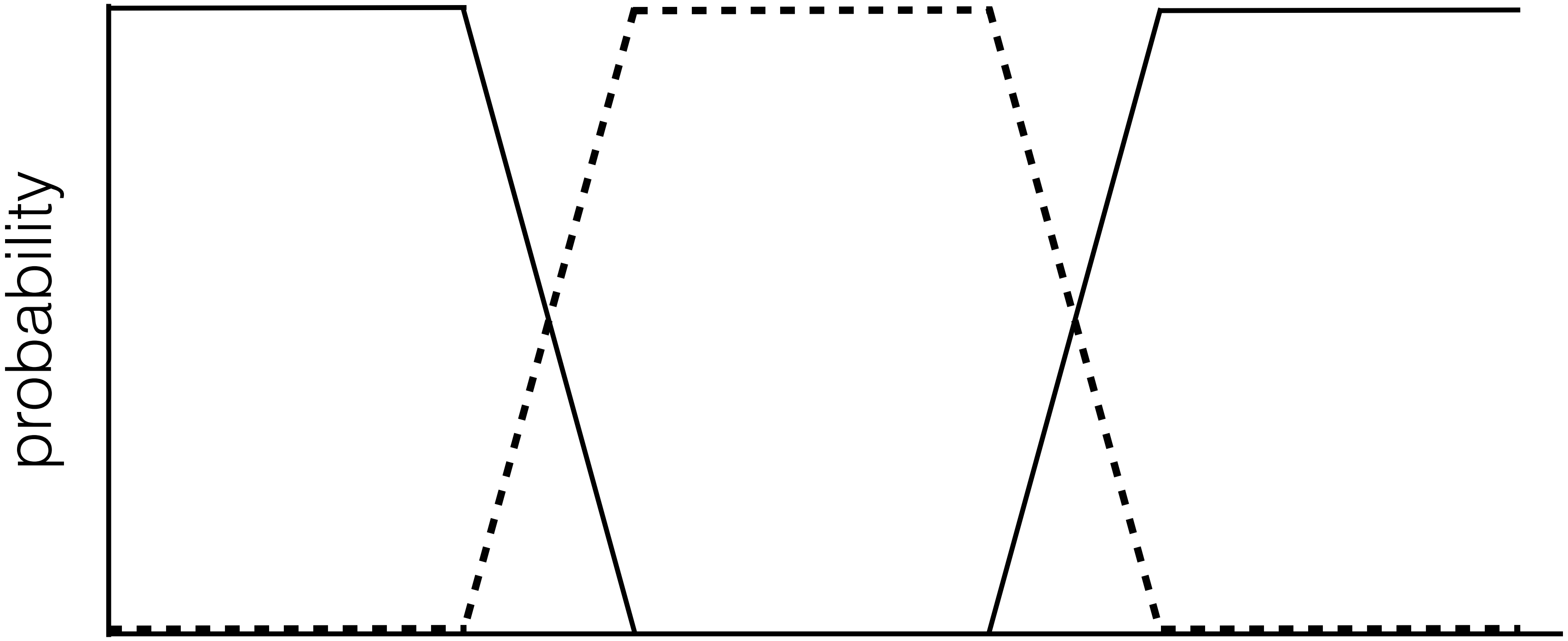


drift width or drifting time

L. Minku, A. White and X. Yao. The Impact of Diversity on On-line Ensemble Learning in the Presence of Concept Drift, IEEE Transactions on Knowledge and Data Engineering 22(5):730-742, 2010.

Characterisation of Concept Drifts

- Recurrence: whether the joint probability distribution reoccurs over time.
- A new joint probability distribution may also be similar, but not the same as a past distribution.



L. Minku, A. White and X. Yao. The Impact of Diversity on On-line Ensemble Learning in the Presence of Concept Drift, IEEE Transactions on Knowledge and Data Engineering 22(5):730-742, 2010.

Online Approaches for Concept Drift Handling

- Abrupt drifts:
  - Typically best dealt with based on explicit (active) approaches:
    - Make use of an explicit concept drift detection method.
    - Special adaptation mechanisms are triggered upon concept drift detection.
- Gradual drifts:
  - Typically best dealt with based on implicit (passive) approaches:
    - Do not make use of an explicit concept drift detection method.
    - Continuously attempt to adapt to the current distribution.
    - Most of such approaches are ensemble approaches.

G. Ditzler et al. Learning in Nonstationary Environments: a survey, IEEE Computational Intelligence Magazine 10(4):12-25, 2015.

B. Krawczyk et al. Ensemble Learning for Data Stream Analysis: a survey, Information Fusion 37:132-156, 2017.

Online Approaches for Concept Drift Handling

- Recurrent drifts:
    - Memory-based approaches can help to deal with recurrent drifts.
        - Single learners retrieved from the memory.
        - Ensembles.
- Severe / not severe drifts:
    - Approaches typically depend on hyperparameters to control the trade-off between sensitivity to noise and ability to deal with drifts of different severities.

How to improve robustness to different types of concept drift?

C. Chiu, L. Minku. A Diversity Framework for Dealing with Multiple Types of Concept Drift Based on Clustering in the Model Space. IEEE Transactions on Neural Networks and Learning Systems, 2020 (in press).

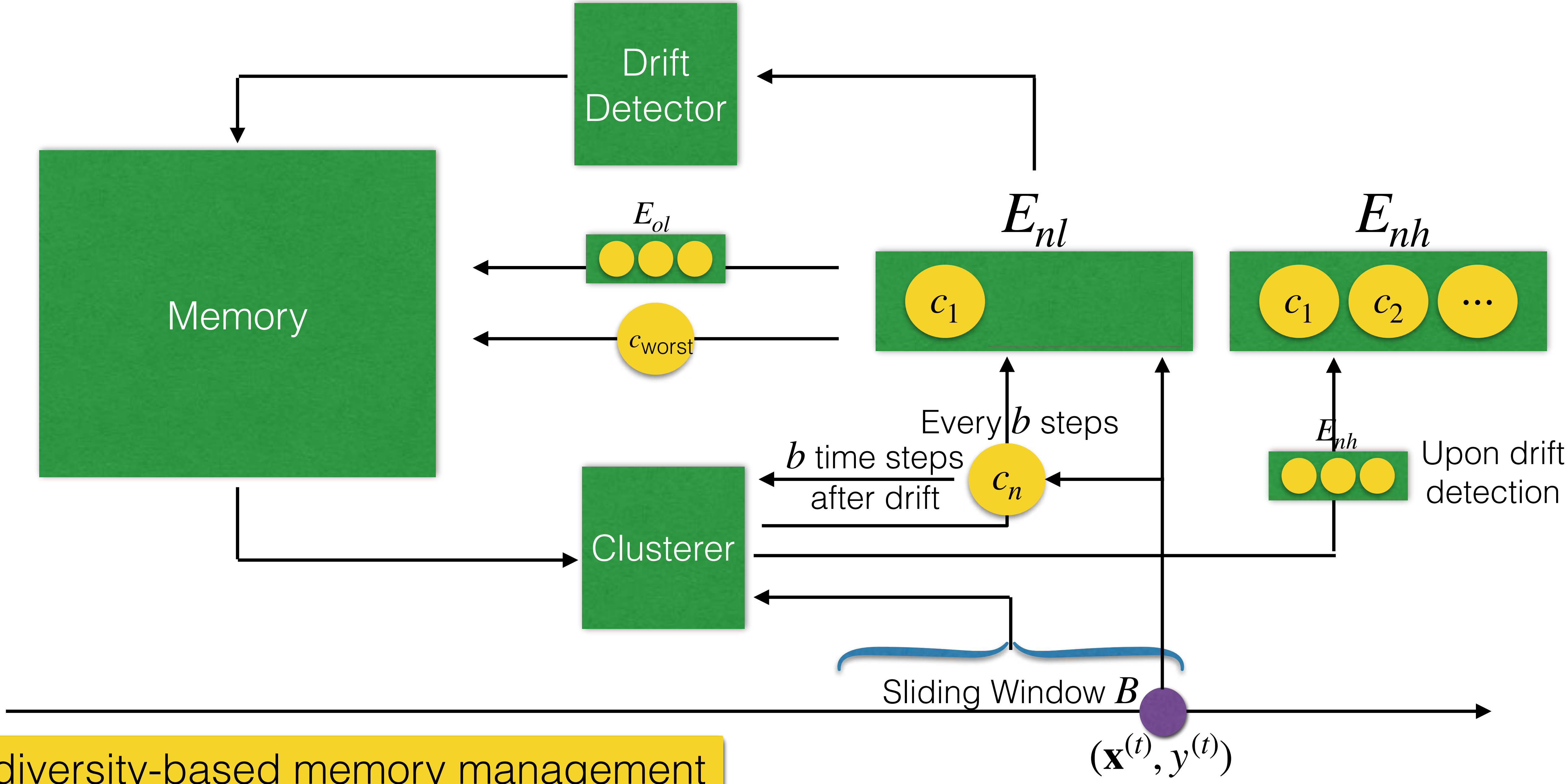
Diversity and Clustering in the Model Space

- Diversity is the level of disagreement among different models. E.g., Q-statistic:

$$Q_{i,k} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}}$$

- Diversity can potentially work as a memory management strategy to maintain a diverse memory of previous models, increasing the chances that relevant past models are kept.
- Clustering the memory models can help to decide which models to retrieve from the memory to deal with various types of drift.
- No previous work had exploited the full potential of diversity as a memory strategy to increase robustness to different types of drift.

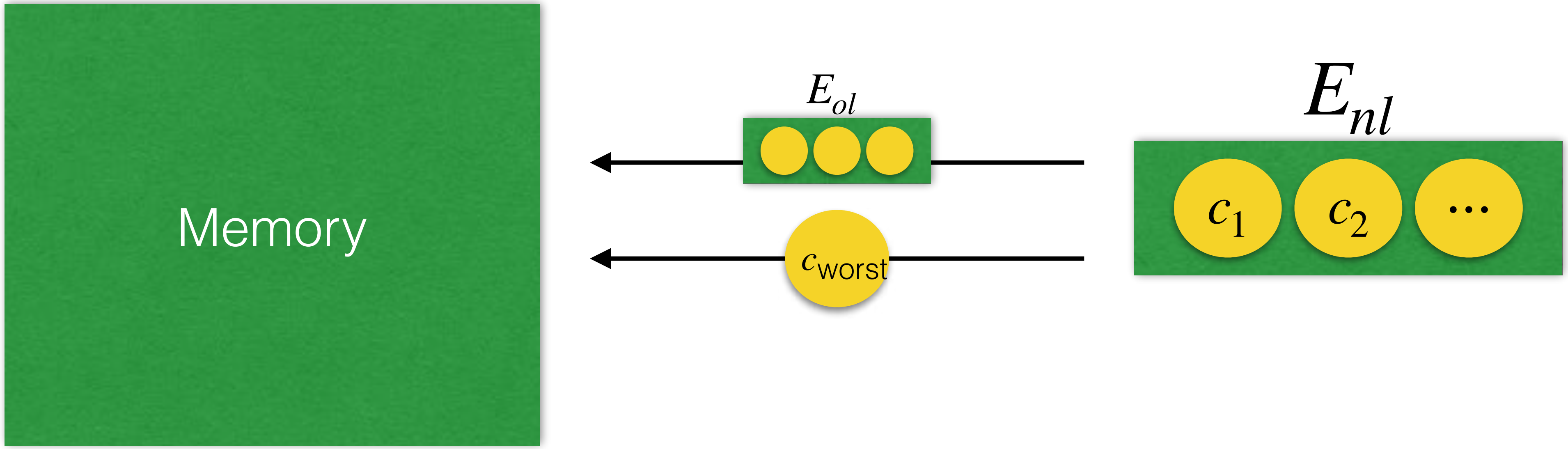
Concept Drift Handling Based on Clustering in the Model Space (CDCMS)



A diversity-based memory management strategy is used to decide which model to keep in the memory.

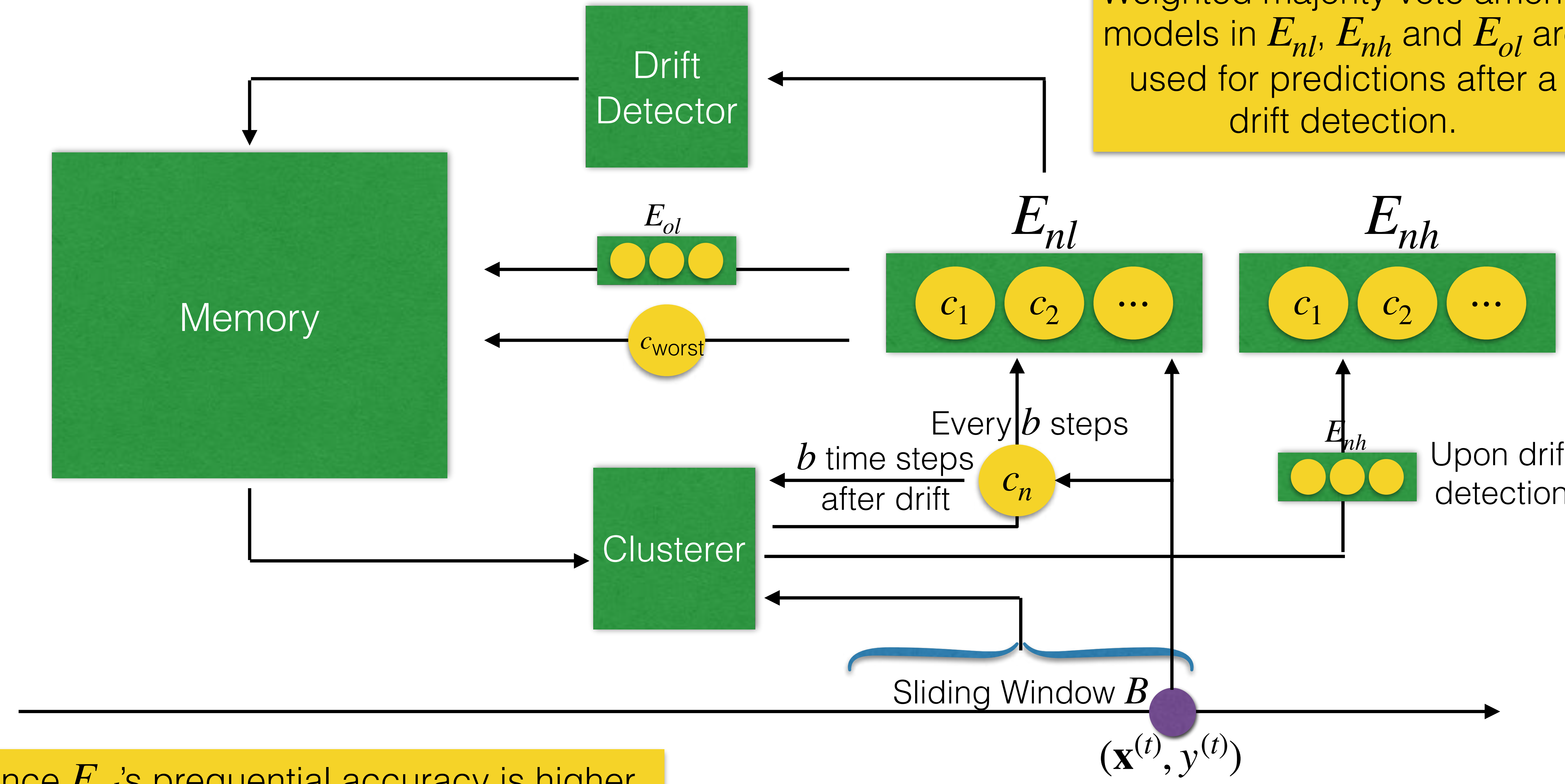
Diversity-Based Memory Maintenance

- If the memory is full, a model will be deleted from it.
- The model to be deleted is the most similar one to the model being added, according to the diversity metric Q.
- Maintaining a diverse memory is key to deal with recurrent drifts or any drifts that may benefit from past knowledge.



Concept Drift Handling Based on Clustering in the Model Space (CDCMS)

Weighted majority vote among models in E_{nl}, E_{nh} and E_{ol} are used for predictions after a drift detection.



Once E_{nl}'s prequential accuracy is higher than that of the others, the others are deactivated.

Synthetic Datasets

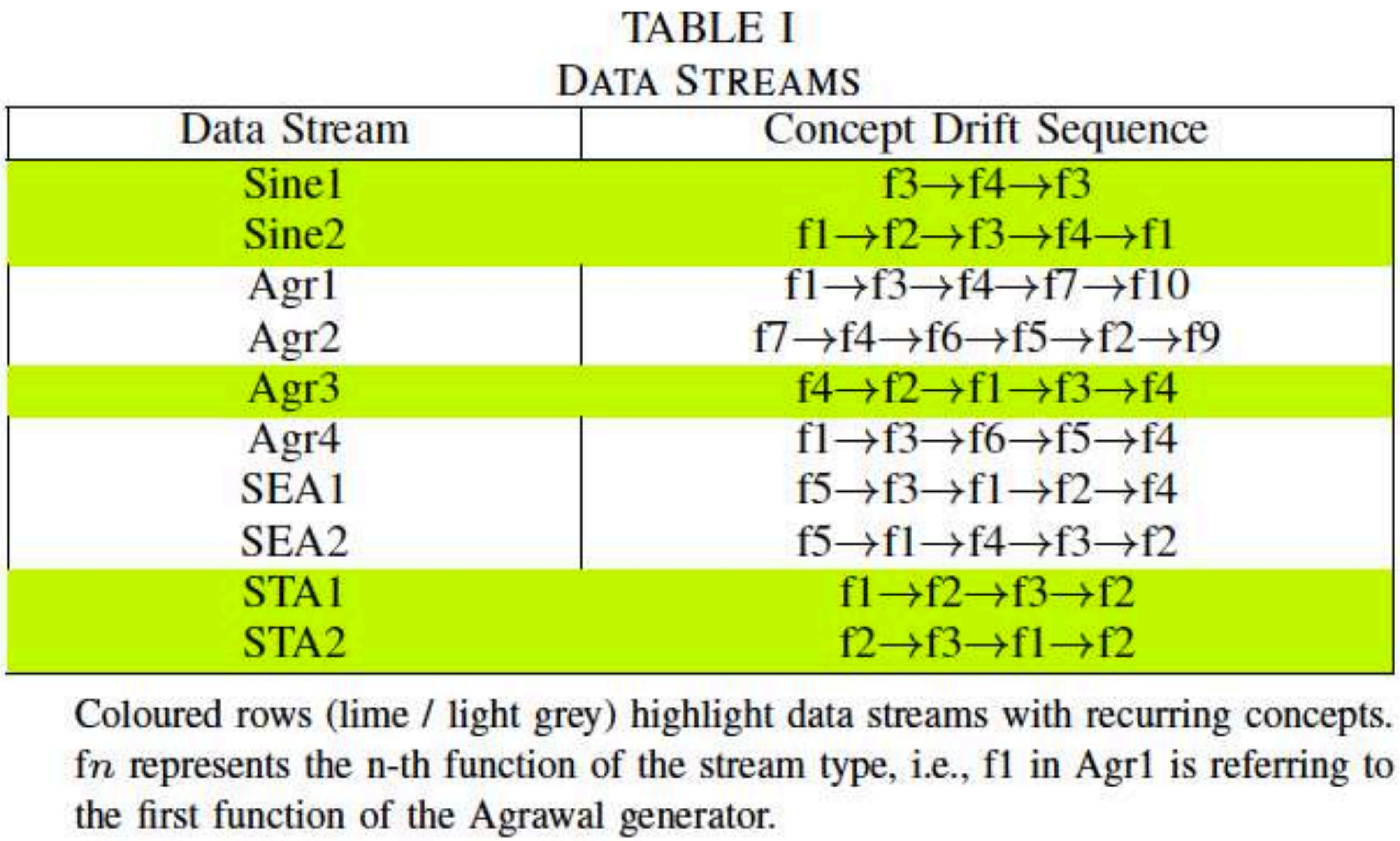


TABLE I
DATA STREAMS

Data Stream	Concept Drift Sequence
Sine1	f3→f4→f3
Sine2	f1→f2→f3→f4→f1
Agr1	f1→f3→f4→f7→f10
Agr2	f7→f4→f6→f5→f2→f9
Agr3	f4→f2→f1→f3→f4
Agr4	f1→f3→f6→f5→f4
SEA1	f5→f3→f1→f2→f4
SEA2	f5→f1→f4→f3→f2
STA1	f1→f2→f3→f2
STA2	f2→f3→f1→f2

Coloured rows (lime / light grey) highlight data streams with recurring concepts. fn represents the n-th function of the stream type, i.e., f1 in Agr1 is referring to the first function of the Agrawal generator.

Abrupt and gradual with width of 2000 time steps.

Synthetic Datasets

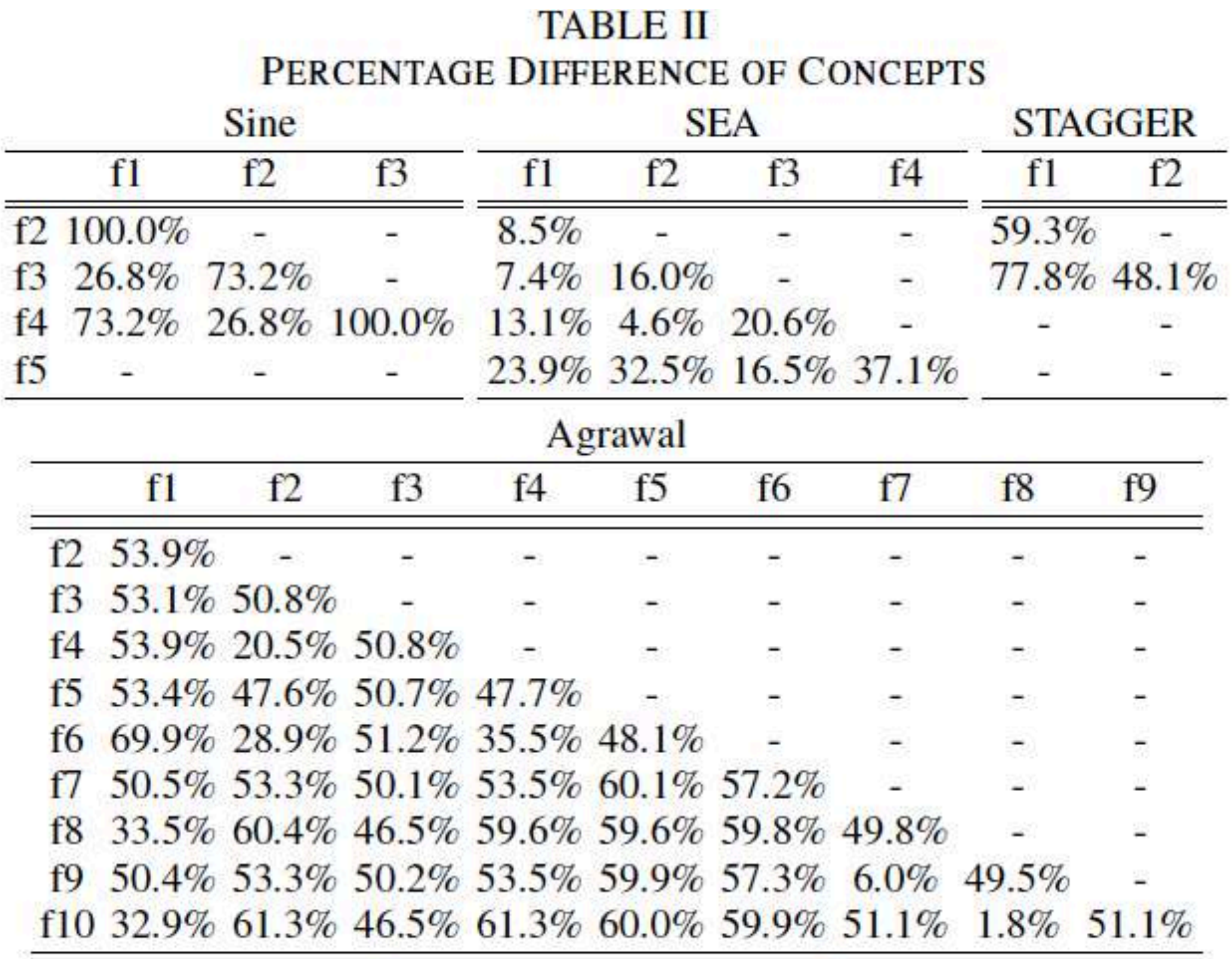


TABLE II
PERCENTAGE DIFFERENCE OF CONCEPTS

	Sine			SEA				STAGGER	
	f1	f2	f3	f1	f2	f3	f4	f1	f2
f2	100.0%	-	-	8.5%	-	-	-	59.3%	-
f3	26.8%	73.2%	-	7.4%	16.0%	-	-	77.8%	48.1%
f4	73.2%	26.8%	100.0%	13.1%	4.6%	20.6%	-	-	-
f5	-	-	-	23.9%	32.5%	16.5%	37.1%	-	-

	Agrawal								
	f1	f2	f3	f4	f5	f6	f7	f8	f9
f2	53.9%	-	-	-	-	-	-	-	-
f3	53.1%	50.8%	-	-	-	-	-	-	-
f4	53.9%	20.5%	50.8%	-	-	-	-	-	-
f5	53.4%	47.6%	50.7%	47.7%	-	-	-	-	-
f6	69.9%	28.9%	51.2%	35.5%	48.1%	-	-	-	-
f7	50.5%	53.3%	50.1%	53.5%	60.1%	57.2%	-	-	-
f8	33.5%	60.4%	46.5%	59.6%	59.6%	59.8%	49.8%	-	-
f9	50.4%	53.3%	50.2%	53.5%	59.9%	57.3%	6.0%	49.5%	-
f10	32.9%	61.3%	46.5%	61.3%	60.0%	59.9%	51.1%	1.8%	51.1%

$$diff(f_a, f_b) = \sum_{i=1}^{n} \frac{|\, y_{fa}^{(i)} - y_{fb}^{(i)} \,|}{n}$$

where $n = 1m$

Real World Datasets

- KDD Cup 99 — network intrusion detection. Ps: the drifts themselves are not real.
- Power supply — predict which hour the power supply comes from.
- Sensor — predict which sensor the temperature, humidity, light and voltage comes from.

Data streams available at: https://www.cse.fau.edu/~xqzhu/stream.html

Baseline Approaches

- HTNB: Hoeffding tree with naive Bayes in the leaves.

  P. Domingos and G. Hulten, “Mining High-Speed Data Streams,” ACM SIGKDD, 11 2002.

- DDD: diversity-based explicit ensemble approach (not prepared for recurrent drifts).

  L. L. Minku and X. Yao, DDD: A New Ensemble Approach for Dealing with Concept Drift, IEEE TKDE 24: 619–633, 2012.

- DP: diverse memory explicit approach for recurrent drifts (not prepared for gradual drifts).

  C. W. Chiu and L. L. Minku, Diversity-Based Pool of Models for Dealing with Recurring Concepts, IJCNN, pp. 2759–2766, 2018.

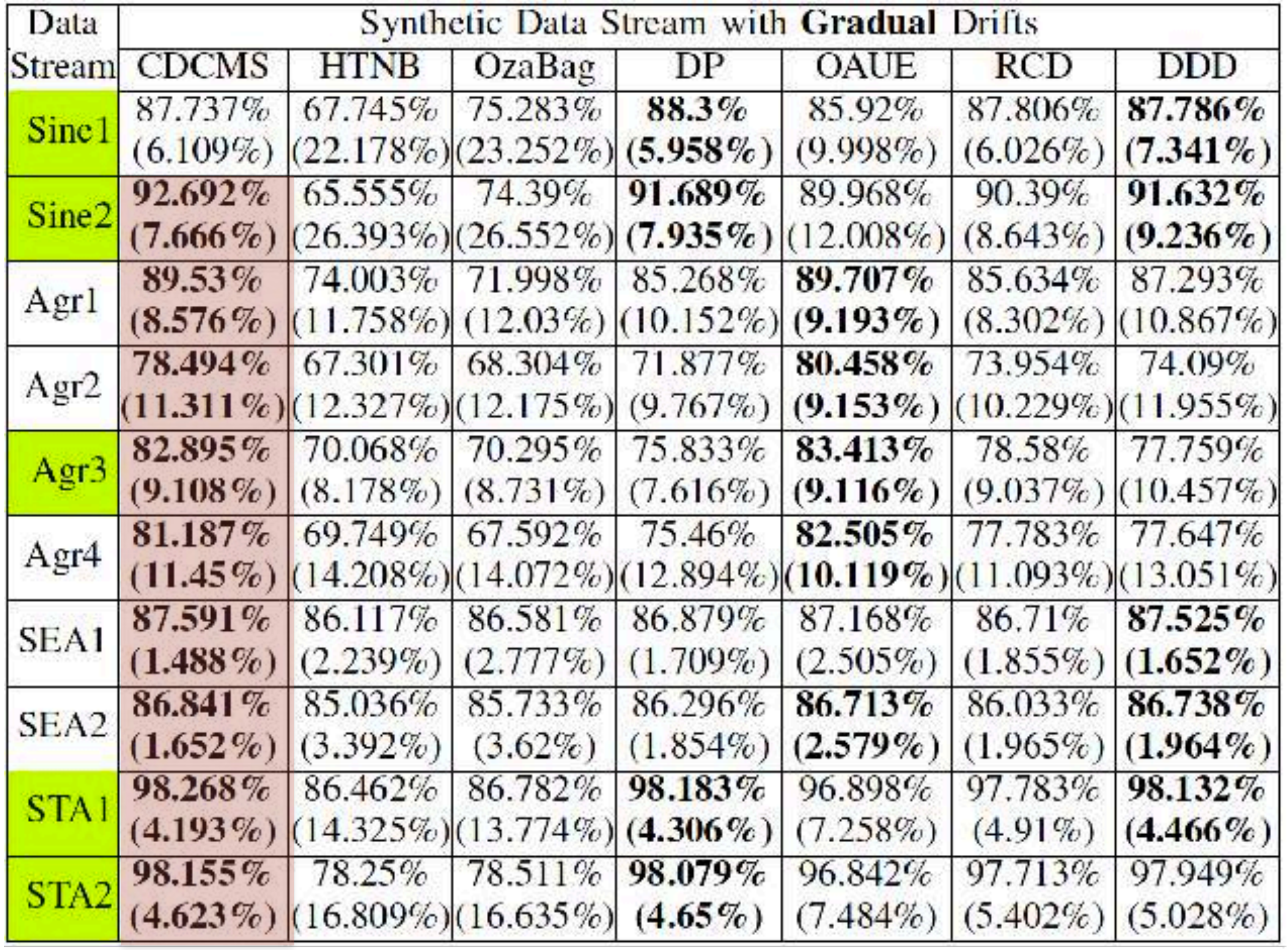
- RCD: memory-based explicit approach for recurrent drifts (based on FIFO management strategy).

  J. Goncalves Jr and R. Barros, RCD: A Recurring Concept Drift Framework, Patt. Recognit. Lett., 34:1018–1025, 2013.

- OAUE: implicit ensemble approach (delete worst as memory management strategy).

  D. Brzezinski and J. Stefanowski, Combining Block-based and Online Methods in Learning Ensembles from Concept Drifting Data Streams, Information Sciences, 265:50–67, 2014

Accuracy on Synthetic Data Streams with Gradual Drifts



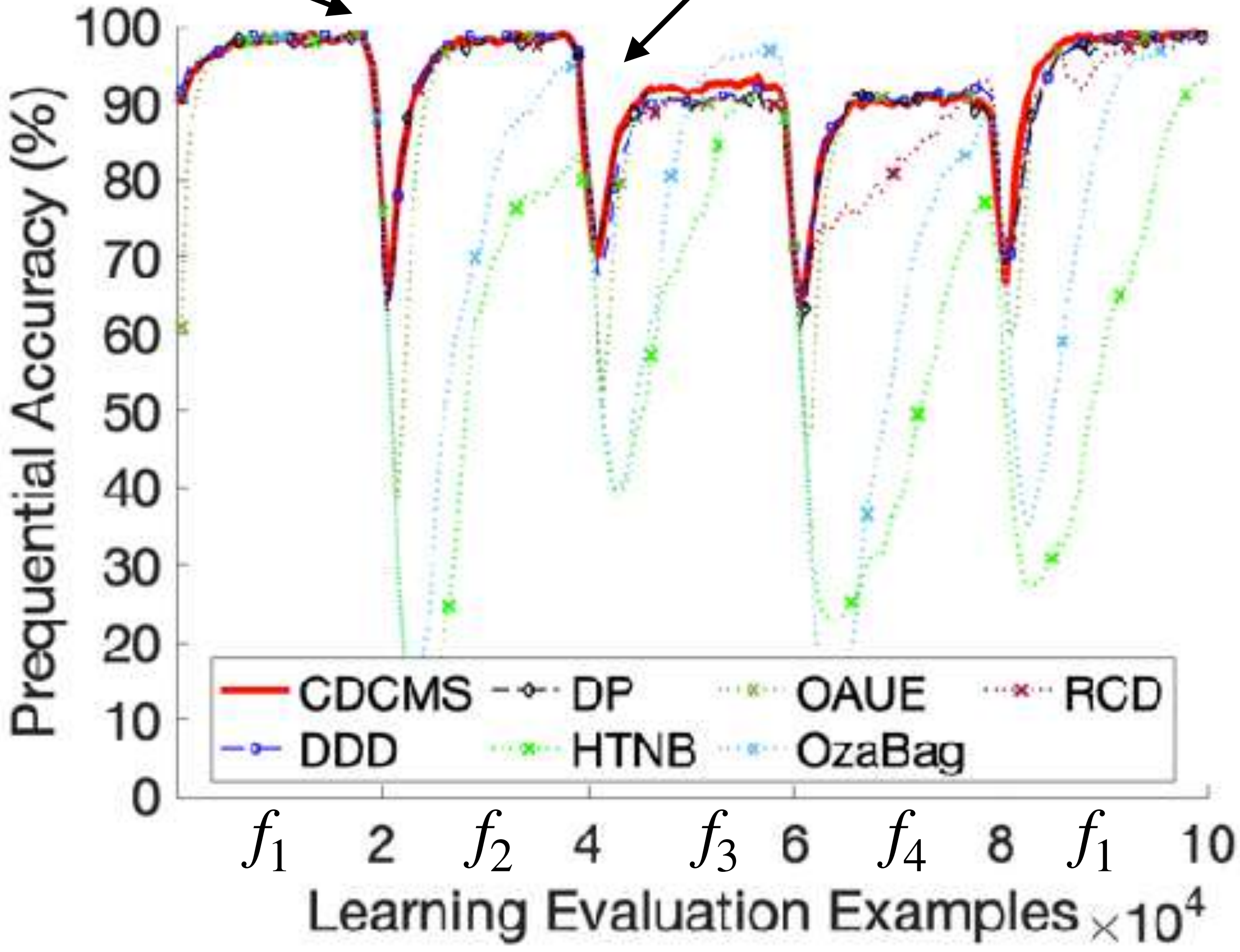
Data	Synthetic Data Stream with **Gradual** Drifts						
Stream	CDCMS	HTNB	OzaBag	DP	OAUE	RCD	DDD
Sine1	87.737% (6.109%)	67.745% (22.178%)	75.283% (23.252%)	**88.3%** **(5.958%)**	85.92% (9.998%)	87.806% (6.026%)	**87.786%** **(7.341%)**
Sine2	**92.692%** **(7.666%)**	65.555% (26.393%)	74.39% (26.552%)	**91.689%** **(7.935%)**	89.968% (12.008%)	90.39% (8.643%)	**91.632%** **(9.236%)**
Agr1	**89.53%** **(8.576%)**	74.003% (11.758%)	71.998% (12.03%)	85.268% (10.152%)	**89.707%** **(9.193%)**	85.634% (8.302%)	87.293% (10.867%)
Agr2	**78.494%** **(11.311%)**	67.301% (12.327%)	68.304% (12.175%)	71.877% (9.767%)	**80.458%** **(9.153%)**	73.954% (10.229%)	74.09% (11.955%)
Agr3	**82.895%** **(9.108%)**	70.068% (8.178%)	70.295% (8.731%)	75.833% (7.616%)	**83.413%** **(9.116%)**	78.58% (9.037%)	77.759% (10.457%)
Agr4	**81.187%** **(11.45%)**	69.749% (14.208%)	67.592% (14.072%)	75.46% (12.894%)	**82.505%** **(10.119%)**	77.783% (11.093%)	77.647% (13.051%)
SEA1	**87.591%** **(1.488%)**	86.117% (2.239%)	86.581% (2.777%)	86.879% (1.709%)	87.168% (2.505%)	86.71% (1.855%)	**87.525%** **(1.652%)**
SEA2	**86.841%** **(1.652%)**	85.036% (3.392%)	85.733% (3.62%)	86.296% (1.854%)	**86.713%** **(2.579%)**	86.033% (1.965%)	**86.738%** **(1.964%)**
STA1	**98.268%** **(4.193%)**	86.462% (14.325%)	86.782% (13.774%)	**98.183%** **(4.306%)**	96.898% (7.258%)	97.783% (4.91%)	**98.132%** **(4.466%)**
STA2	**98.155%** **(4.623%)**	78.25% (16.809%)	78.511% (16.635%)	**98.079%** **(4.65%)**	96.842% (7.484%)	97.713% (5.402%)	97.949% (5.028%)

Bold font shows top ranked methods based on Friedman and Nemenyi tests at level of significance of 0.05.

Sample Gradual Drift Results

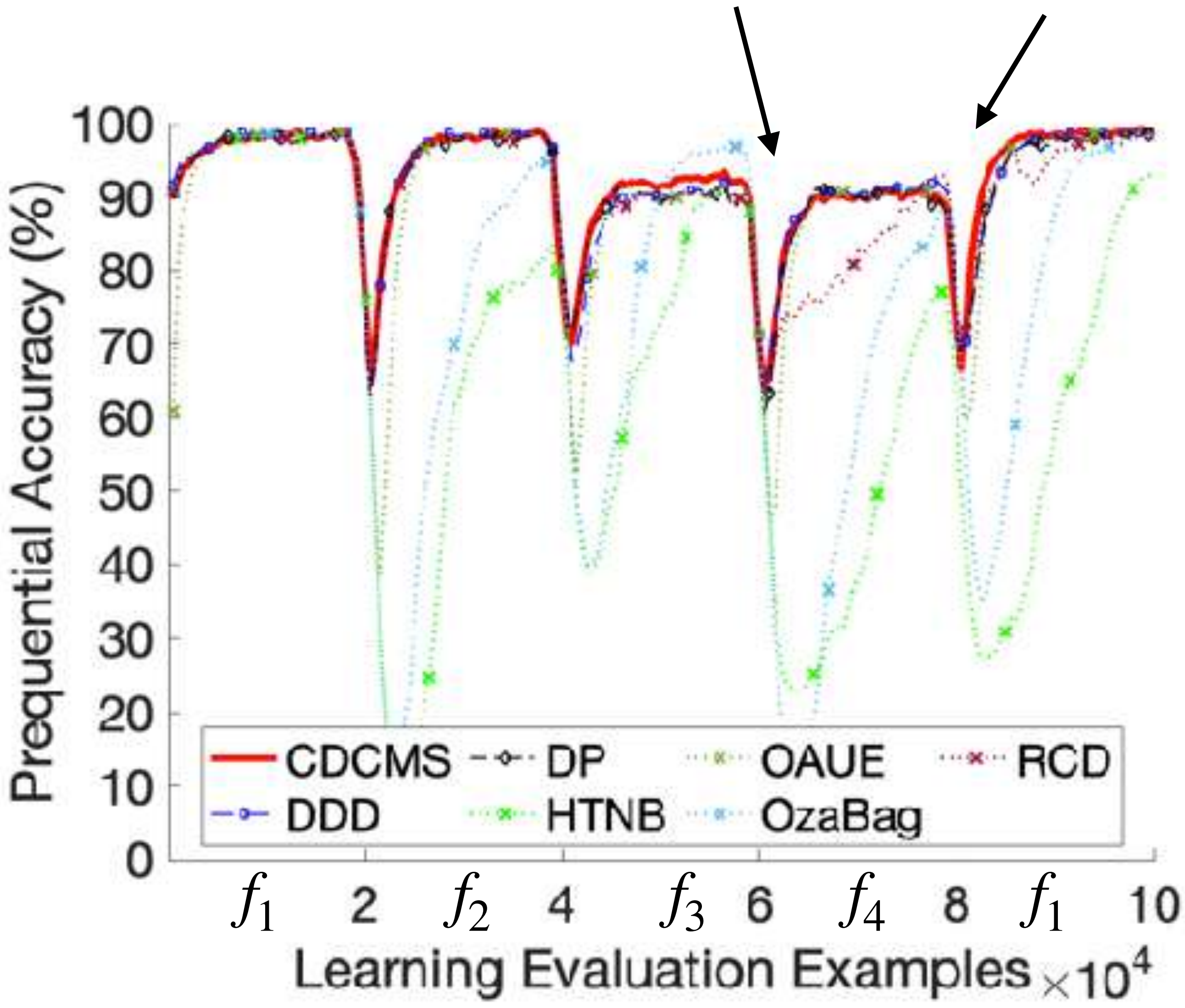
Drift detection leading to new unseen distribution (severity 100%)

Drift detection recovering f_1 (difference 26.8%)



Sin2 - Gradual

Sample Gradual Drift Results

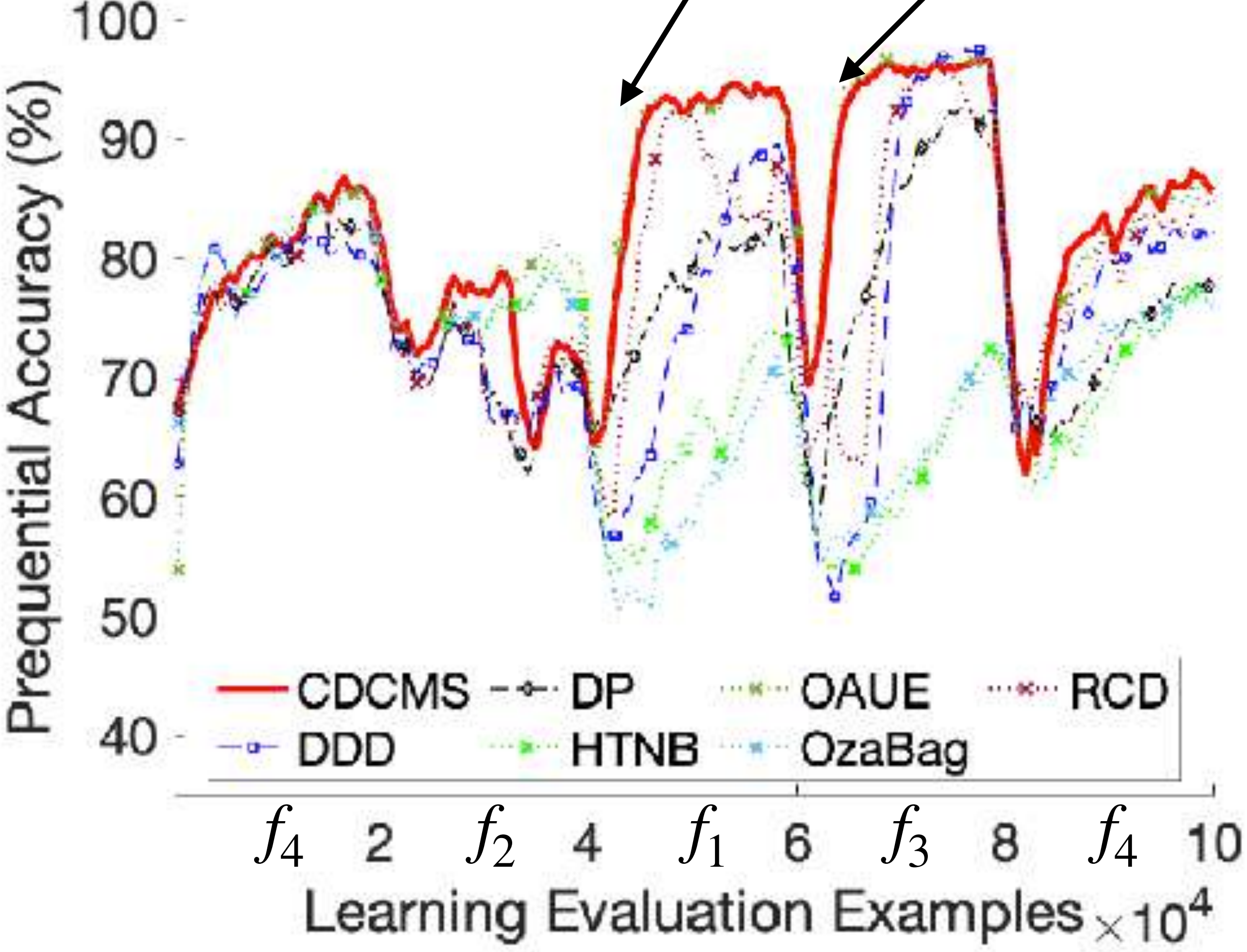


Drift detection recovering f_1 and f_2 (difference 73.2% and 26.8%)

Drift detection recovering f_1

Sin2 - Gradual

Sample Gradual Drift Results



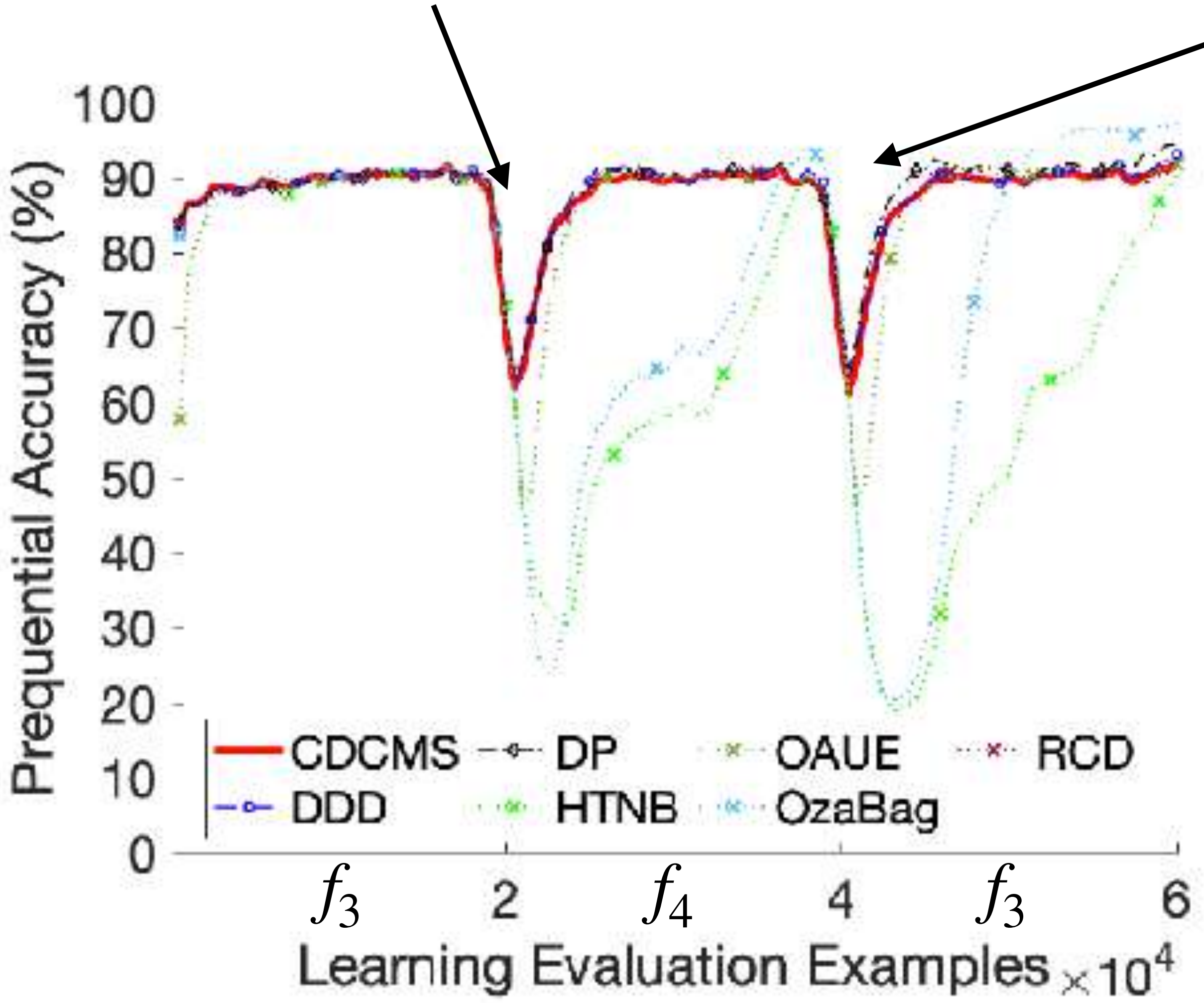
No drift detection

Agr3 - Gradual

Sample Gradual Drift Results

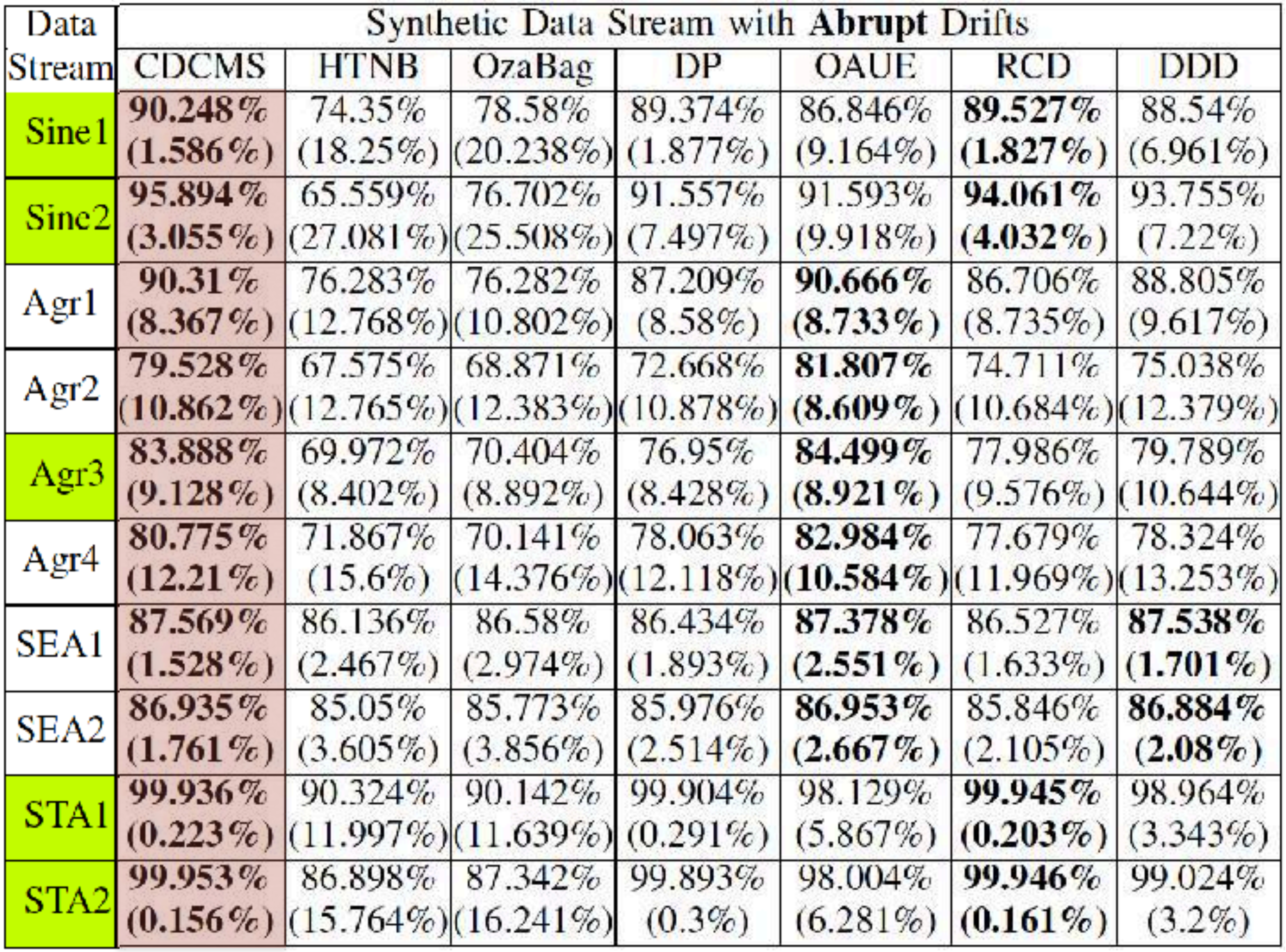
False positive drift detection learning an intermediate distribution

Recovering intermediate model, instead of pure model



Sin1 - Gradual

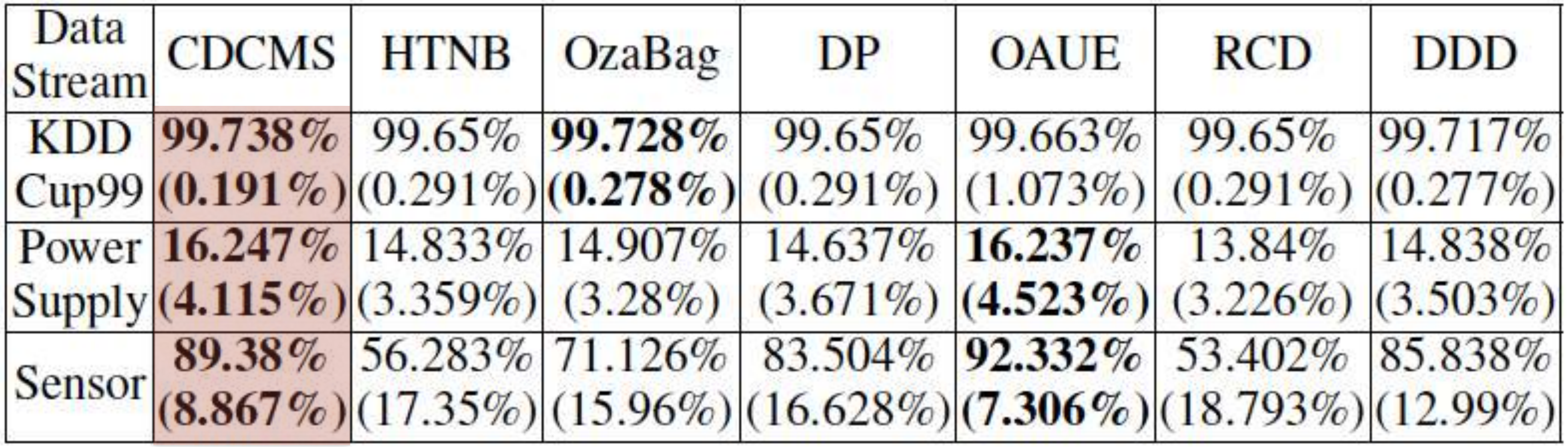
Accuracy on Synthetic Data Streams with Abrupt Drifts



Data Stream	Synthetic Data Stream with **Abrupt** Drifts						
	CDCMS	HTNB	OzaBag	DP	OAUE	RCD	DDD
Sine1	**90.248%** **(1.586%)**	74.35% (18.25%)	78.58% (20.238%)	89.374% (1.877%)	86.846% (9.164%)	**89.527%** **(1.827%)**	88.54% (6.961%)
Sine2	**95.894%** **(3.055%)**	65.559% (27.081%)	76.702% (25.508%)	91.557% (7.497%)	91.593% (9.918%)	**94.061%** **(4.032%)**	93.755% (7.22%)
Agr1	**90.31%** **(8.367%)**	76.283% (12.768%)	76.282% (10.802%)	87.209% (8.58%)	**90.666%** **(8.733%)**	86.706% (8.735%)	88.805% (9.617%)
Agr2	**79.528%** **(10.862%)**	67.575% (12.765%)	68.871% (12.383%)	72.668% (10.878%)	**81.807%** **(8.609%)**	74.711% (10.684%)	75.038% (12.379%)
Agr3	**83.888%** **(9.128%)**	69.972% (8.402%)	70.404% (8.892%)	76.95% (8.428%)	**84.499%** **(8.921%)**	77.986% (9.576%)	79.789% (10.644%)
Agr4	**80.775%** **(12.21%)**	71.867% (15.6%)	70.141% (14.376%)	78.063% (12.118%)	**82.984%** **(10.584%)**	77.679% (11.969%)	78.324% (13.253%)
SEA1	**87.569%** **(1.528%)**	86.136% (2.467%)	86.58% (2.974%)	86.434% (1.893%)	**87.378%** **(2.551%)**	86.527% (1.633%)	**87.538%** **(1.701%)**
SEA2	**86.935%** **(1.761%)**	85.05% (3.605%)	85.773% (3.856%)	85.976% (2.514%)	**86.953%** **(2.667%)**	85.846% (2.105%)	**86.884%** **(2.08%)**
STA1	**99.936%** **(0.223%)**	90.324% (11.997%)	90.142% (11.639%)	99.904% (0.291%)	98.129% (5.867%)	**99.945%** **(0.203%)**	98.964% (3.343%)
STA2	**99.953%** **(0.156%)**	86.898% (15.764%)	87.342% (16.241%)	99.893% (0.3%)	98.004% (6.281%)	**99.946%** **(0.161%)**	99.024% (3.2%)

Bold font shows top ranked methods based on Friedman and Nemenyi tests at level of significance of 0.05.

Accuracy on Real World Data Streams



Data Stream	CDCMS	HTNB	OzaBag	DP	OAUE	RCD	DDD
KDD Cup99	**99.738%** **(0.191%)**	99.65% (0.291%)	**99.728%** **(0.278%)**	99.65% (0.291%)	99.663% (1.073%)	99.65% (0.291%)	99.717% (0.277%)
Power Supply	**16.247%** **(4.115%)**	14.833% (3.359%)	14.907% (3.28%)	14.637% (3.671%)	**16.237%** **(4.523%)**	13.84% (3.226%)	14.838% (3.503%)
Sensor	**89.38%** **(8.867%)**	56.283% (17.35%)	71.126% (15.96%)	83.504% (16.628%)	**92.332%** **(7.306%)**	53.402% (18.793%)	85.838% (12.99%)

Bold font shows top ranked methods based on Friedman and Nemenyi tests at level of significance of 0.05.

Time Complexity Analysis

- Prediction:

$O(eM_p)$ where e is the ensemble size and M_p is the base learner prediction time complexity.

- Training:

$O(eM_t)$ where e is the ensemble size and M_t is the base learner training time complexity.

Time Complexity Analysis

- Clustering in the model space (triggered only when there are concept drift detections):

$O(HbM_p)$ where H is the memory size, b is the number of examples in the sliding window, and M_p is the base learner prediction time complexity.

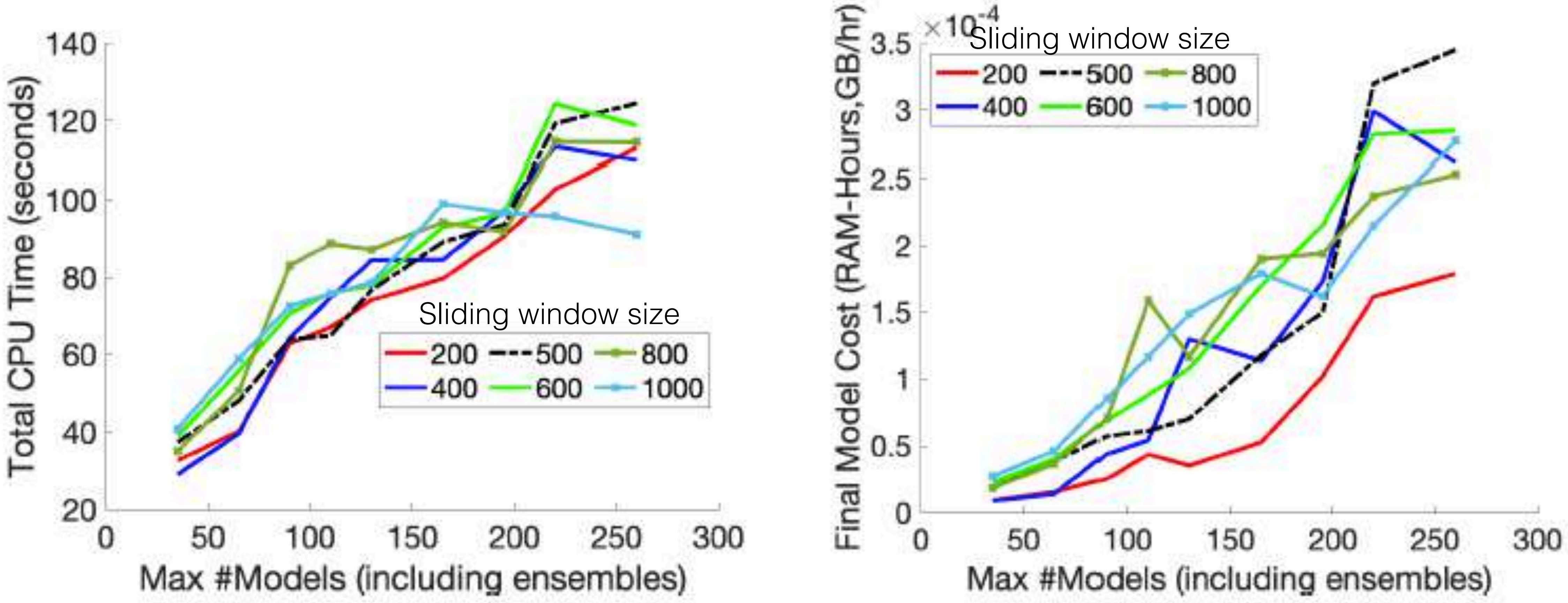
$O(G)$ where G is the clustering method time complexity.

$$O(HbM_p + G)$$

- Memory management (triggered at every b time steps and upon drift detection):

$O(LbM_p)$ where L is the number of models being compared for diversity, b is the number of examples in the sliding window and M_p is the base learner prediction time complexity.

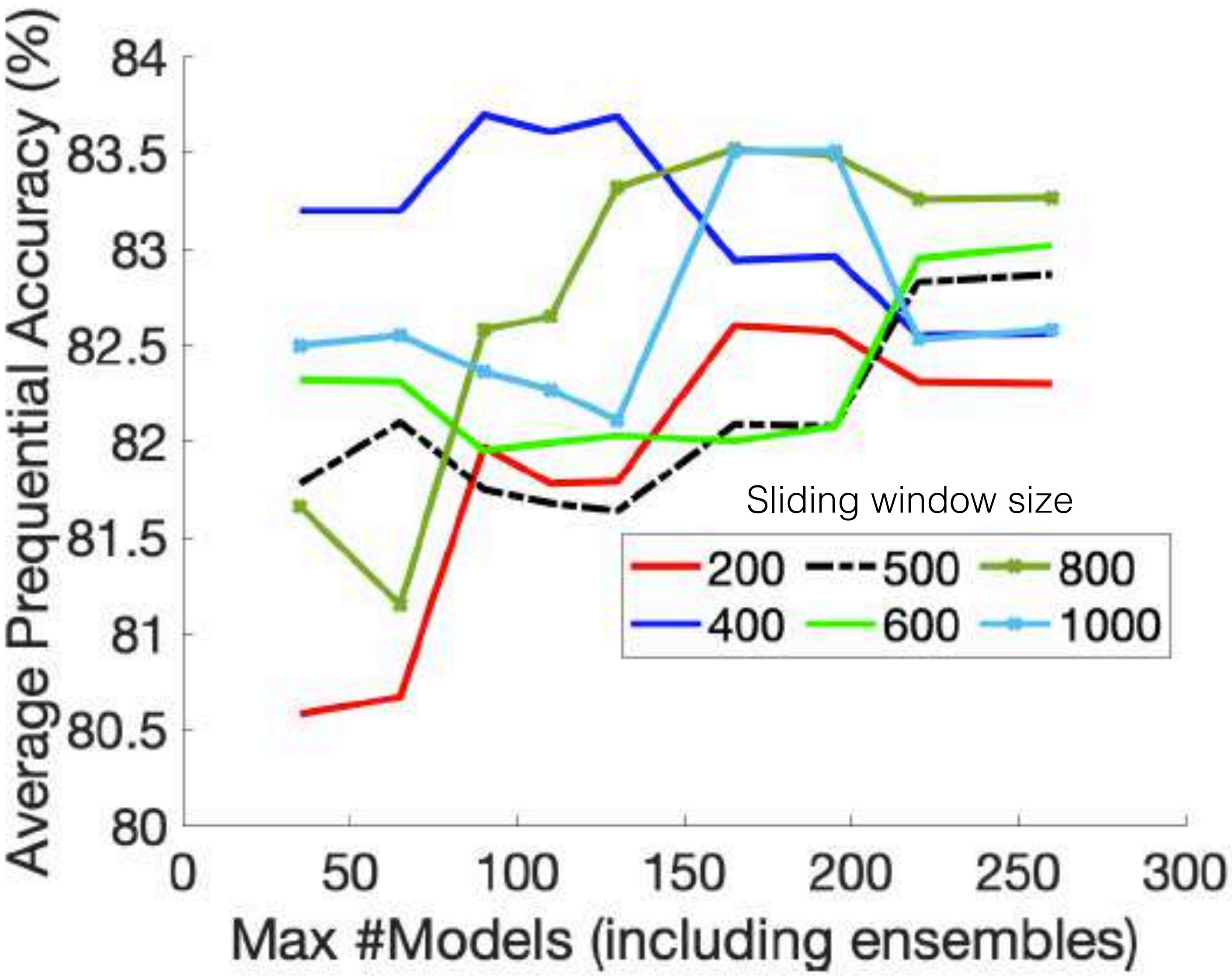
Runtime and Memory



(a) Runtime

(b) Memory

Accuracy for Different Sliding Window and Maximum Number of Models



(c) Accuracy

Conclusions

- Multiple types of concept drift may co-occur in real world problems.

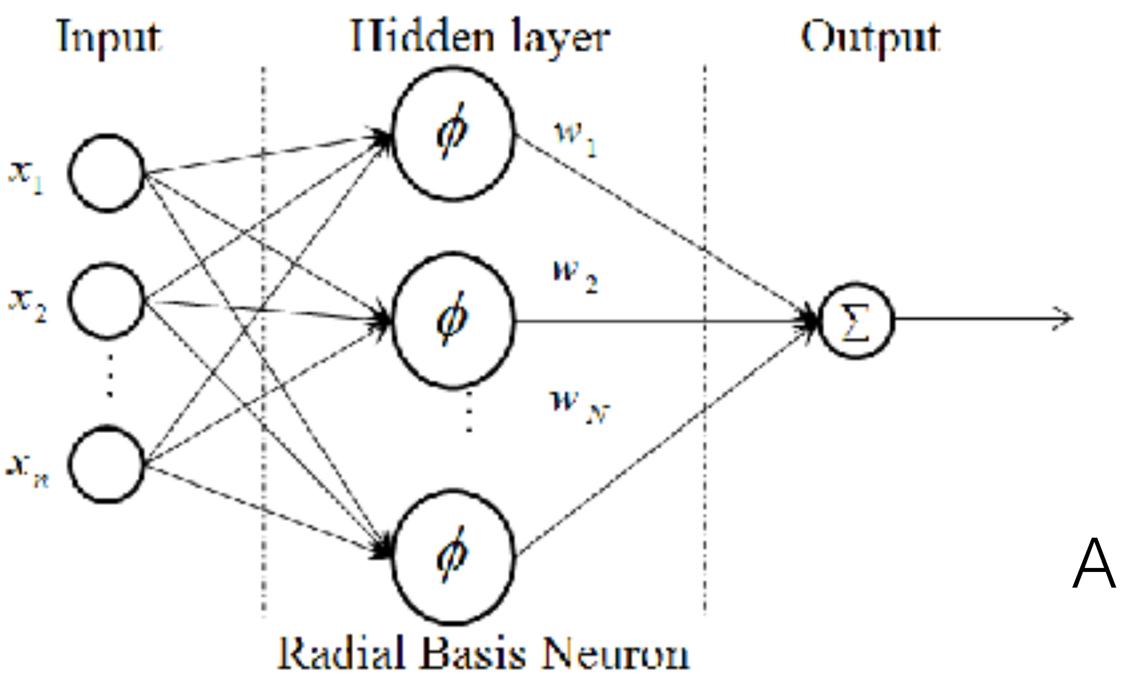
- CDCMS makes use of diversity and clustering in the model space to improve robustness to several different types of drift.

- Most of the time, it successfully managed to recover models from the memory when they are useful to deal with various types of concept drift.

- It successfully created new learners to tackle sudden abrupt drifts that did not involve reoccurrence.

Other Data Stream Challenges - Partially Labelled Data

- Semi-supervised data stream learning.



$$\begin{aligned}\mathcal{L}(B^{(t)}, \mathbf{w}) = &-\frac{1}{L}\sum_{i\in B_l^{(t)}} [y_i \ln(f_i) + (1-y_i)\ln(1-f_i)]\\ &-\frac{\lambda}{U}\sum_{i\in B_u^{(t)}} [u_i \ln(f_i) + (1-u_i)\ln(1-f_i)]\\ &+\frac{\alpha}{2N}||\mathbf{w}||^2,\end{aligned}$$

Automatic adjustment of learning rate to cope with abrupt and gradual drifts.

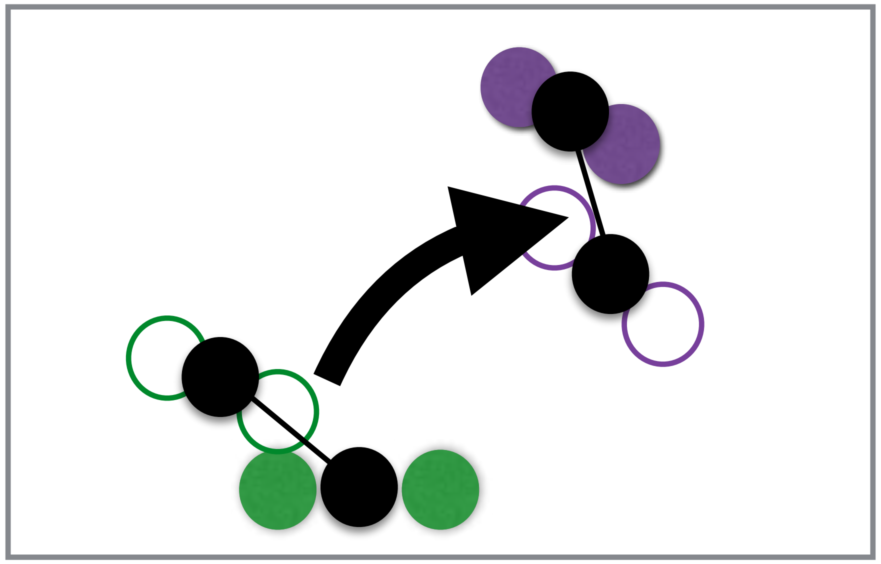
Closed form solution to learn centres.

R. Soares and L. Minku. OSNN: An Online Semisupervised Neural Network for Nonstationary Data Streams, IEEE Transactions on Neural Networks and Learning Systems, 2021 (in press).

Other Data Stream Challenges - Small Target Data

- Transfer learning for data streams.

Target data is projected into the source space.



Source models are then used as part of an ensemble to predict the target.

H. Du, L. Minku, H. Zhou. MARLINE: Multi-Source Mapping Transfer Learning for Non-Stationary Environments", IEEE International Conference on Data Mining, 2020.

Other Data Stream Challenges - Verification Latency

- Waiting time strategy with resampling to cope with mislabeled instances.

Implementing a change

We don’t know the true label of the change at this stage — that’s why we need to predict it.

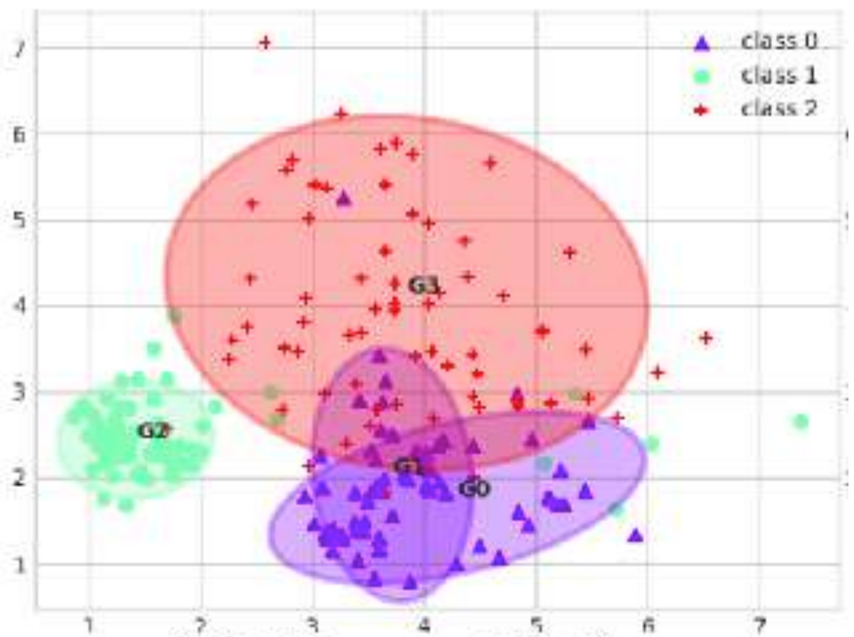
Versioning Repository

Cabral, G.; Minku, L.; Shihab, E.; Mujahid, S. Class Imbalance Evolution and Verification Latency in Just-in-Time Software Defect Prediction. International Conference on Software Engineering, 2019.

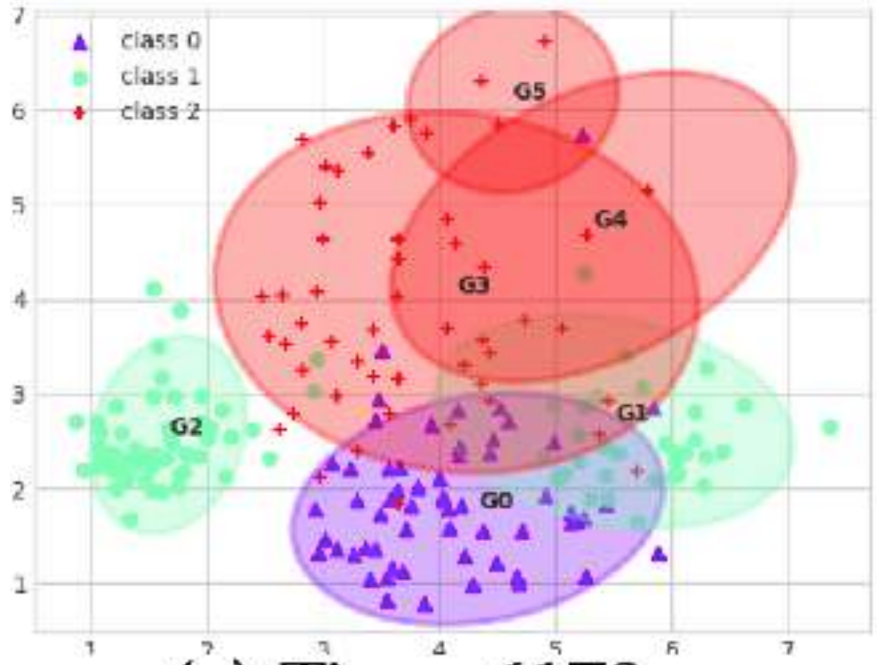
Tabassum, S.; Minku, L.; Feng, D.; Cabral, G.; Song, L. An Investigation of Cross-Project Learning in Online Just-In-Time Software Defect Prediction. International Conference on Software Engineering, 2020.

Other Data Stream Challenges

- Dealing with concept drifts in $p^{(t)}(\mathbf{x}, y) = p^{(t)}(y \mid \mathbf{x})p^{(t)}(\mathbf{x})$



(a) Time: 6050



(e) Time: 6170

Update gaussians for non-severe drifts in $p(y \mid \mathbf{x})$

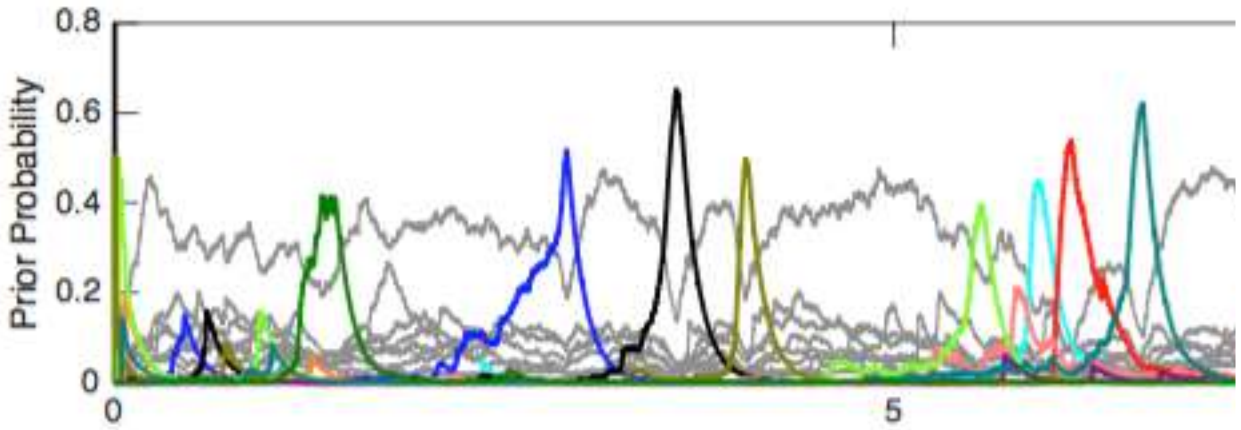
Create gaussians for drifts in $p(\mathbf{x})$

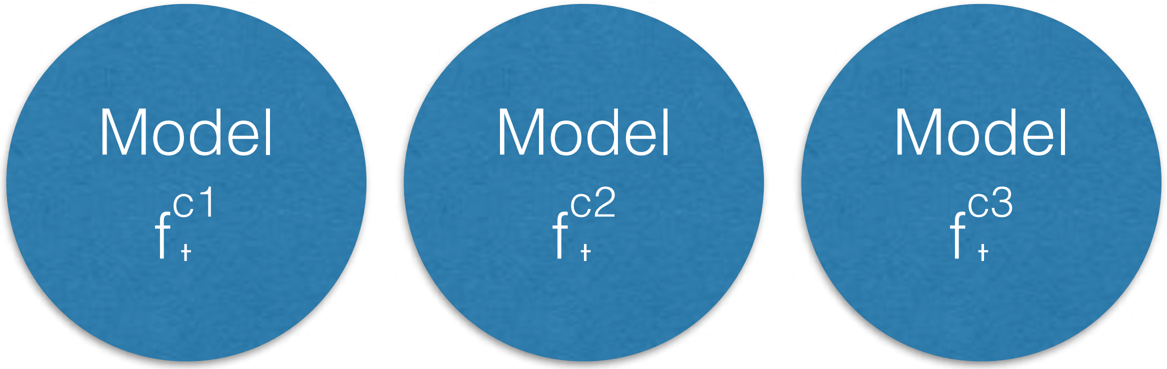
Reset system for severe drifts in $p(y \mid \mathbf{x})$

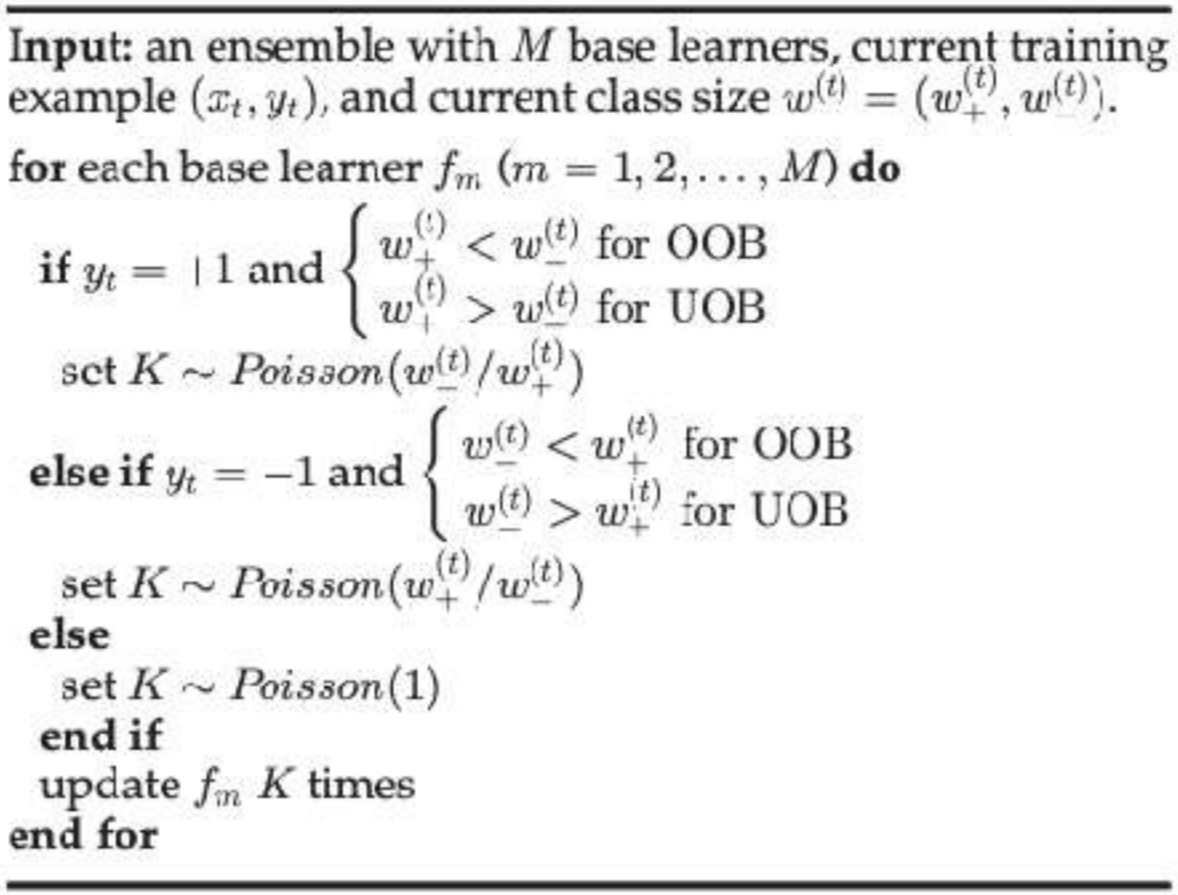
G. Oliveira, L. Minku, A. Oliveira. Tackling Virtual and Real Concept Drifts: An Adaptive Gaussian Mixture Model Approach. IEEE Transactions on Knowledge and Data Engineering, 2021 (in press).

Other Data Stream Challenges

- Gradual class evolution and class imbalance evolution:





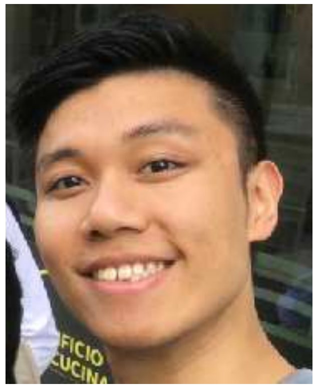


```
Input: an ensemble with $M$ base learners, current training example $(x_t, y_t)$, and current class size $w^{(t)} = (w_+^{(t)}, w_-^{(t)})$.
for each base learner $f_m$ $(m = 1, 2, \ldots, M)$ do
  if $y_t = +1$ and $\begin{cases} w_+^{(t)} < w_-^{(t)} \text{ for OOB} \\ w_+^{(t)} > w_-^{(t)} \text{ for UOB} \end{cases}$
    set $K \sim Poisson(w_-^{(t)}/w_+^{(t)})$
  else if $y_t = -1$ and $\begin{cases} w_-^{(t)} < w_+^{(t)} \text{ for OOB} \\ w_-^{(t)} > w_+^{(t)} \text{ for UOB} \end{cases}$
    set $K \sim Poisson(w_+^{(t)}/w_-^{(t)})$
  else
    set $K \sim Poisson(1)$
  end if
  update $f_m$ $K$ times
end for
```

Sun et al. Online Ensemble Learning of Data Streams with Gradually Evolved Classes, IEEE Transactions on Knowledge and Data Engineering 28(6):1532—1545, 2016.

S. Wang, L. Minku, X. Yao. Resampling-Based Ensemble Methods for Online Class Imbalance Learning, IEEE Transactions on Knowledge and Data Engineering 27(5): 1356-1368, 2015.

With Thanks



Chun Wai Chiu

Gustavo Oliveira

Sadia Tabassum

Honghui Du

Rodrigo Soares

George Cabral

Adriano Oliveira

Shuo Wang