CS 876 - Streaming Data Systems

Concept Drift Detection in AIR

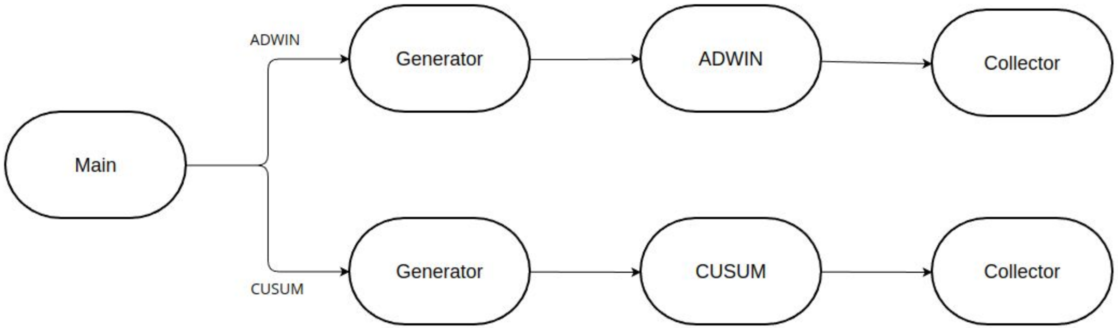
Yash Koushik (IMT2020033)

Problem Statement

- We aim to solve the problem of efficiently processing and analyzing the drift in concept of the streaming data under consideration by implementing concept drift detection algorithms in AIR .
- In real-world applications, data streams are dynamic and subject to changes over time. Concept drift, where the statistical properties of the data change, is a common challenge. There is a need to assess the performance of existing distributed concept drift detection algorithms in AIR systems.

Approach

- In this project, we make use of AIR dataflow. Concept drift detection data flow consists of the below nodes.



Generator Node

- Key role in generating a continuous event stream with concept drift.
- Two sets defined: trending (random subset) and remaining.
- Bags contain five randomly chosen trending items, ensuring consistency.
- Concept drift introduced at intervals, simulating item replacements.

ADWIN Node

- Utilizes buckets to store recent data. Each bucket contains sum and count of elements within a timeframe.
- Periodically calculates mean and variance based on bucket information.
- Mean estimates the current concept, while variance assesses the likelihood of change.
- Compares current variance to a threshold (delta-based). Triggers drift event if variance significantly exceeds the threshold.
- Resetting the buckets upon drift detection, ensures quick adaptation to new concept without the influence of the outdated data.

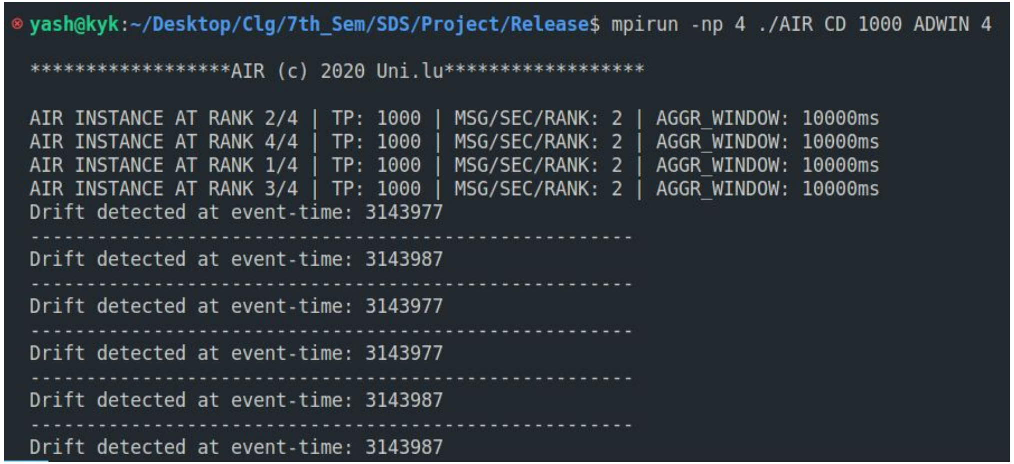
CUSUM Node

- Utilizes counters for positive and negative deviations to identify potential shifts in data distribution.
- Define reference value for deviation measurement. Declare sumup and sumdown counters and set thresholds for deviations triggering drift events.
- Deserialize data points and calculate the euclidean distance deviation from the reference value.
- Update sumup and sumdown. Trigger drift event if either of these exceed threshold.
- Reset sumup and sumdown to 0 (Allows adaptation, similar to ADWIN)

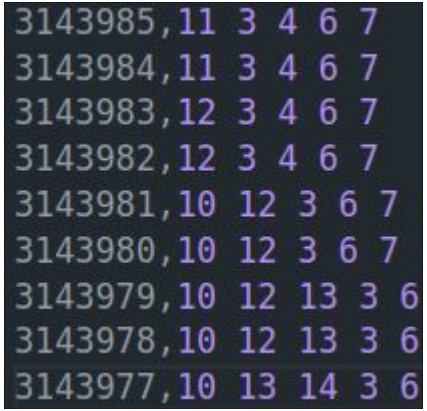
Collector Node

- This node marks the end of the data flow.
- Main responsibilities involve processing EventAdwin events and handling drift events.
- Upon receiving an EventAdwin structure, the code examines the drift flag.
- If a drift event is detected, the timestamp associated with the event is logged and printed in the terminal.

Evaluation - 1

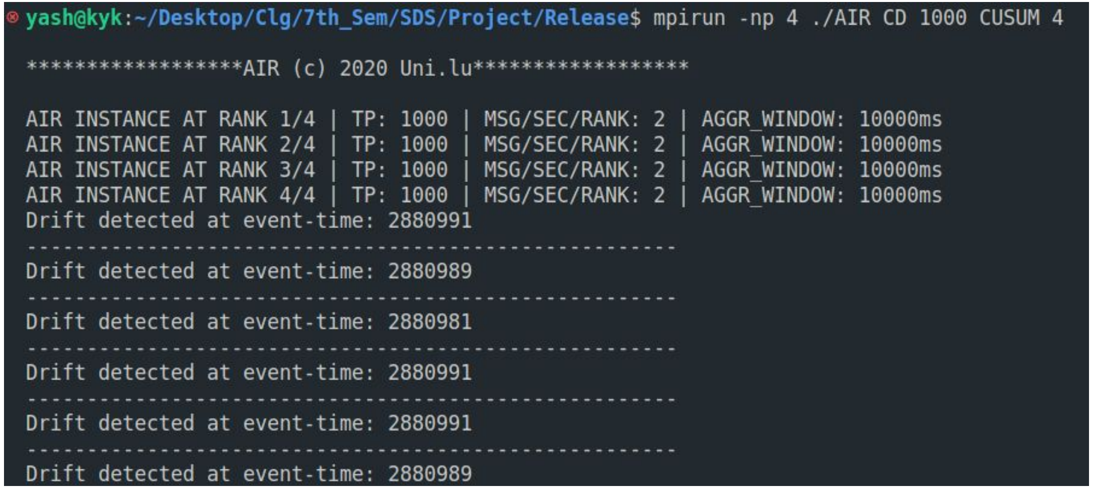


```
⊗ yash@kyk:~/Desktop/Clg/7th_Sem/SDS/Project/Release$ mpirun -np 4 ./AIR CD 1000 ADWIN 4

 ******************AIR (c) 2020 Uni.lu******************

 AIR INSTANCE AT RANK 2/4 | TP: 1000 | MSG/SEC/RANK: 2 | AGGR_WINDOW: 10000ms
 AIR INSTANCE AT RANK 4/4 | TP: 1000 | MSG/SEC/RANK: 2 | AGGR_WINDOW: 10000ms
 AIR INSTANCE AT RANK 1/4 | TP: 1000 | MSG/SEC/RANK: 2 | AGGR_WINDOW: 10000ms
 AIR INSTANCE AT RANK 3/4 | TP: 1000 | MSG/SEC/RANK: 2 | AGGR_WINDOW: 10000ms
 Drift detected at event-time: 3143977
 ------------------------------------------------------
 Drift detected at event-time: 3143987
 ------------------------------------------------------
 Drift detected at event-time: 3143977
 ------------------------------------------------------
 Drift detected at event-time: 3143977
 ------------------------------------------------------
 Drift detected at event-time: 3143987
 ------------------------------------------------------
 Drift detected at event-time: 3143987
```

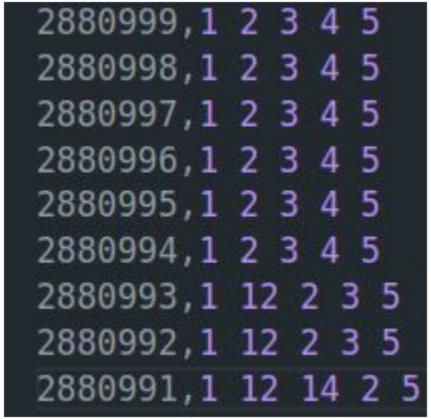


```
3143985,11 3 4 6 7
3143984,11 3 4 6 7
3143983,12 3 4 6 7
3143982,12 3 4 6 7
3143981,10 12 3 6 7
3143980,10 12 3 6 7
3143979,10 12 13 3 6
3143978,10 12 13 3 6
3143977,10 13 14 3 6
```

Evaluation - 2



```
yash@kyk:~/Desktop/Clg/7th_Sem/SDS/Project/Release$ mpirun -np 4 ./AIR CD 1000 CUSUM 4

 *****************AIR (c) 2020 Uni.lu*****************

 AIR INSTANCE AT RANK 1/4 | TP: 1000 | MSG/SEC/RANK: 2 | AGGR_WINDOW: 10000ms
 AIR INSTANCE AT RANK 2/4 | TP: 1000 | MSG/SEC/RANK: 2 | AGGR_WINDOW: 10000ms
 AIR INSTANCE AT RANK 3/4 | TP: 1000 | MSG/SEC/RANK: 2 | AGGR_WINDOW: 10000ms
 AIR INSTANCE AT RANK 4/4 | TP: 1000 | MSG/SEC/RANK: 2 | AGGR_WINDOW: 10000ms
 Drift detected at event-time: 2880991
 -----------------------------------------------------
 Drift detected at event-time: 2880989
 -----------------------------------------------------
 Drift detected at event-time: 2880981
 -----------------------------------------------------
 Drift detected at event-time: 2880991
 -----------------------------------------------------
 Drift detected at event-time: 2880991
 -----------------------------------------------------
 Drift detected at event-time: 2880989
```



```
2880999,1 2 3 4 5
2880998,1 2 3 4 5
2880997,1 2 3 4 5
2880996,1 2 3 4 5
2880995,1 2 3 4 5
2880994,1 2 3 4 5
2880993,1 12 2 3 5
2880992,1 12 2 3 5
2880991,1 12 14 2 5
```

Conclusion

- Detecting concept drift in streaming data is crucial for real-time applications like fraud detection, stock analysis, and network monitoring.
- By addressing this challenge of concept drift, we can significantly improve the performance and effectiveness of streaming data systems.
- The absence of online resources for implementing drift detection algorithms in C++ underscores a substantial knowledge gap, indicating a wide scope for further development in this domain.

Project Category

- I nominate my project under the Excellent category.
- I have done adequate study about the state of the art drift detection algorithms.
- I have laid out the basis for drift detection in AIR by implementing ADWIN and CUSUM drift detection algorithms.
- I have also provided the future aspects and various challenges faced in the report.

Links

- Github link: https://github.com/yashk0311/Drift-Detection-in-AIR
- Demo link: SDS Project

Thank You