Enhancing Concept Drift Detection in Drifting and Imbalanced Data Streams through Meta-Learning

Authors: Mr. Gabriel J. Aguiar [1*]; Dr. Alberto Cano, Ph.D.[1]

[1] College of Engineering, Virginia Commonwealth University, Richmond, USA
* Presenter

Agenda

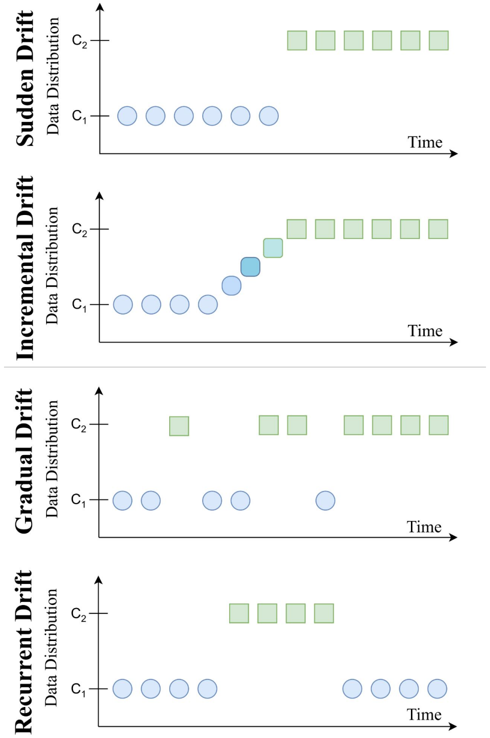
1. Introduction
2. Proposed framework
3. Experimental Setup
4. Results
5. Conclusion and future work

Introduction

- Modern data sources are characterized by producing continuous data in high volume and velocity
  - Scenario known as data streams
- New challenges emerge when dealing with data streams
  - Potentially unbounded, which creates memory constraints
  - Due to its evolving nature, classifier needs to adapt to new concepts that emerge over time (*Concept Drift*)
  - Class imbalance may appear, and ratios can oscillate over time
- Tackling these challenges by detecting changes in data is necessary

Introduction

- A concept drift occurs when the probabilistic distribution of a data stream changes over time.
- A plethora drift detectors have been proposed in recent years
- Selecting the most suitable one for a given data stream requires a priori information, which is not feasible in the online scenario
- There is no single algorithm suitable for every dataset

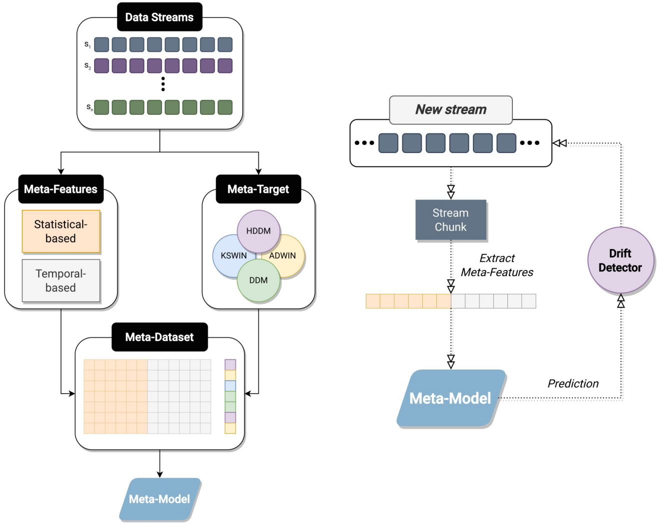


Introduction

- One potential solution is to dynamically recommend the best algorithm for each specific problem
    - The algorithm recommendation problem has been effectively addressed through Meta-Learning
- The core concept of MtL is to leverage knowledge gained from previous similar problems to recommend the most suitable algorithm for a new and unseen data
- Our hypothesis is that MtL can be effectively applied to drifting data streams to dynamically select the most suitable drift detector for unseen chunks of a given stream

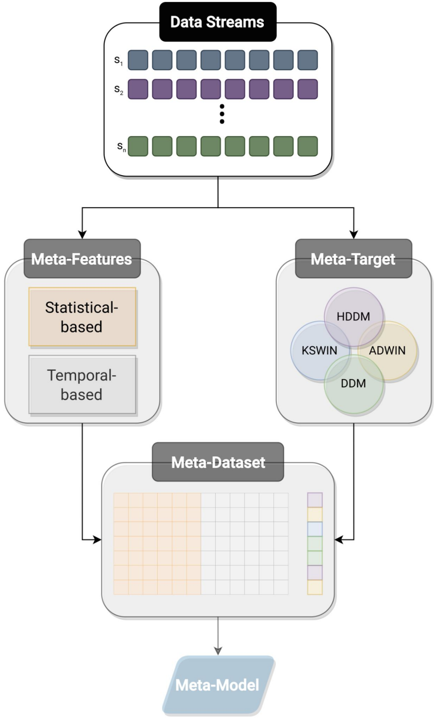
Proposed framework

- We propose an online framework to dynamically select concept drift detectors over data streams.
- The framework consists of two main tasks:
    - Modeling (left)
    - Recommendation (right)
- The modeling task is offline while the recommendation step is online.



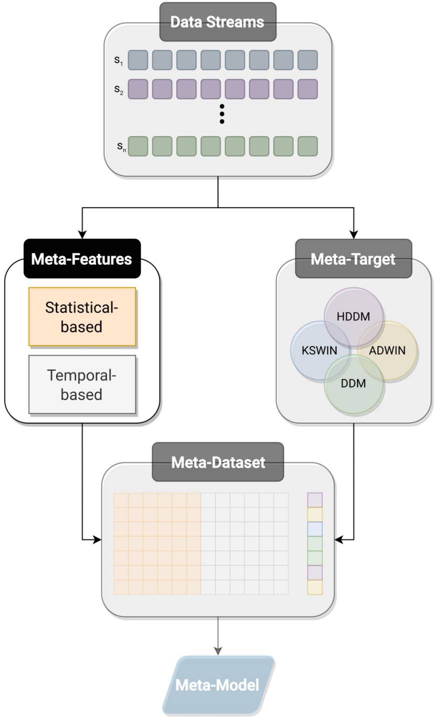
Proposed framework

- We need to define from which streams the model is going to leverage knowledge from
- We synthetically generated 1,334 data streams (meta-examples), using 5 different data stream generators
  - 4 different sizes (# of instances), with concept drift after each quarter of instances and different imbalance scenarios
- The extensive variations of meta-examples in our meta-dataset were chosen to increase the diversity and representativeness of scenarios



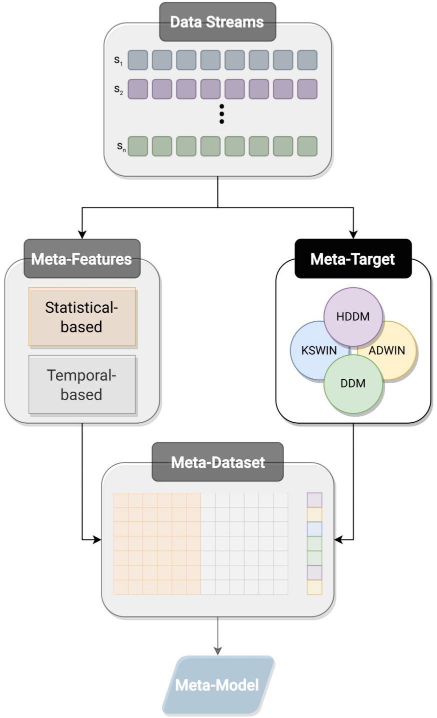
Proposed framework

- To characterize each data stream, we extract meta-features from the selected meta-examples.
- The meta-features can be divided into two groups: Statistical and Temporal.
  - Statistical features describe data without considering temporal relationships between instances, such as Mean, Skewness, Kurtosis, etc.
  - Temporal features focuses on time-sensitive features, e.g., Autocorrelation, Signal distance, etc.



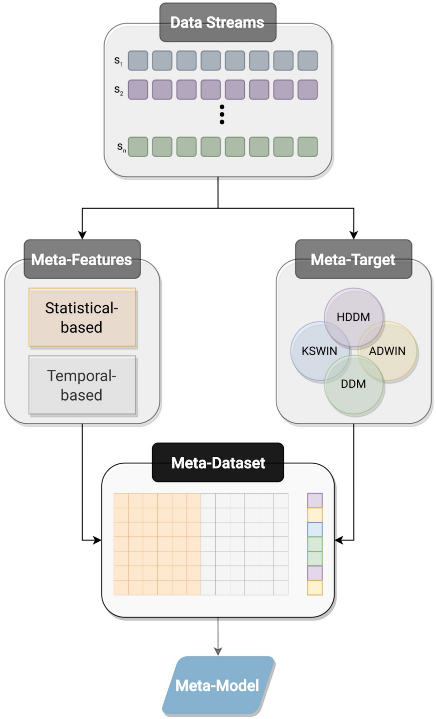
Proposed framework

- As possible meta-targets, we used four widely explored drift detectors:
  - ADWIN, KSWIN, HDDM, and DDM.
- To determine the performance of these drift detectors, we used the Hoeffding Adaptive Tree.
- The meta-label of each meta-example is the relative ranking of each drift detector considering the G-Mean metric



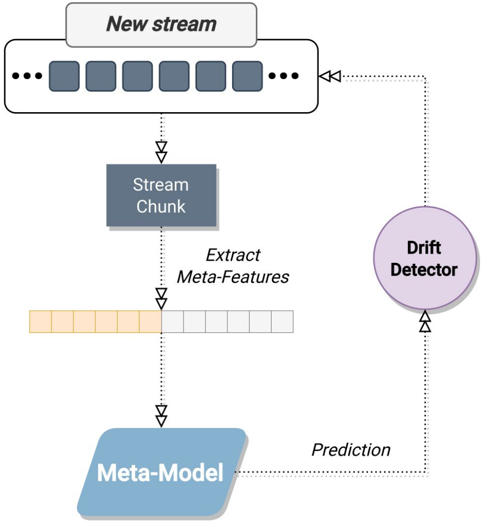
Proposed framework

- By combining the meta-features and the meta-label we would have a meta-dataset
- As our meta-learner, we opted for the Random Forest
- We build four distinct meta-models, one for each drift detector, to predict their relative rankings within the respective data streams.



Proposed framework

- Finally, the meta-model can be applied to a new data stream.
- We extract consecutive chunks of size *w* from the data stream to recommend the most suitable drift detector for a specific time period



Experimental Setup

- The experimental setup was designed to answer the following research questions
    - **RQ1:** Is Meta-Learning able to recommend the best drift detector for an unseen chunk of data?
    - **RQ2:** Is the proposed Meta-Learning framework able to handle imbalanced data streams?
    - **RQ3:** Is the proposed Meta-Learning framework able to handle real-world data stream difficulties?

Experimental Setup

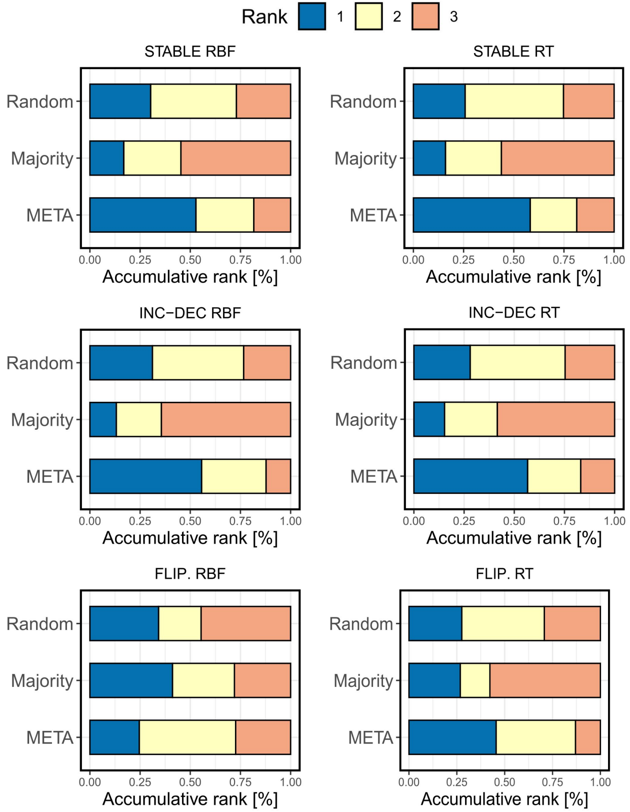
- In order to evaluate the proposed framework we selected a diverse set of benchmark data streams
  - 54 streams synthetic generated
  - 9 streams from real-world domains
- We used two synthetic generators (Random Tree and RBF) and generated streams with different sizes, drift speeds and imbalance ratios
  - Number of instances: 20*k*, 30*k* and 50*k* instances
  - Three imbalance scenarios: STABLE, FLIPPING, INCREASE AND DECREASE.

Experimental Setup

- As comparison methods we selected two baseline recommendation strategies: *Random* and *Majority* (KSWIN)
- We assessed the performance of each recommendation method using Adaptive Hoeffding Tree (AHT) combined with the predicted drift detector, employing the G-Mean metric.
- The framework and respective classifiers were implemented using Python 3.8 and the river package and it is publicly available for future research

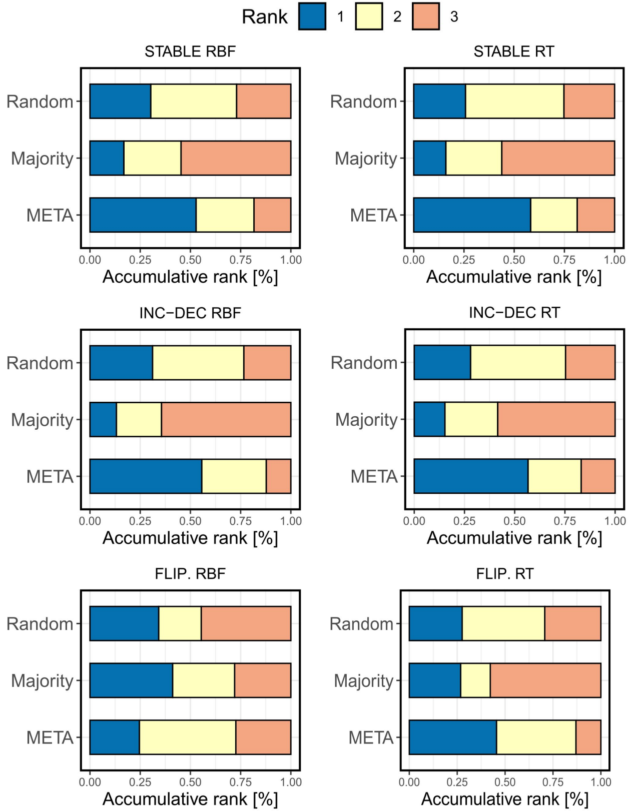
Results

- In the STABLE and INCREASE-DECREASE scenarios, META achieved the highest frequency as the best method for both generators.
- Random recommendation outperformed the Majority approach.



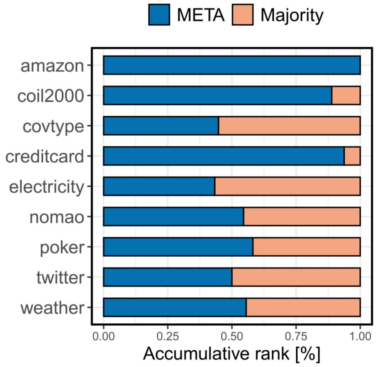
Results

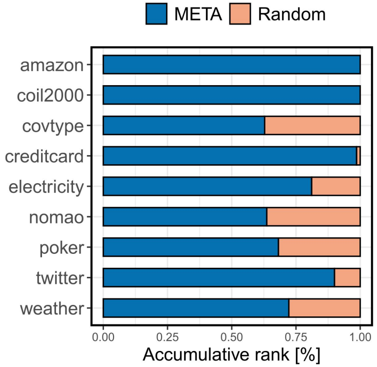
- Regarding the most unstable scenario (FLIPPING), META displayed its worse performance
  - Outperformed by Random and Majority
- The proposed framework effectively recommends the best drift detector most part of the evaluated scenarios.



Results

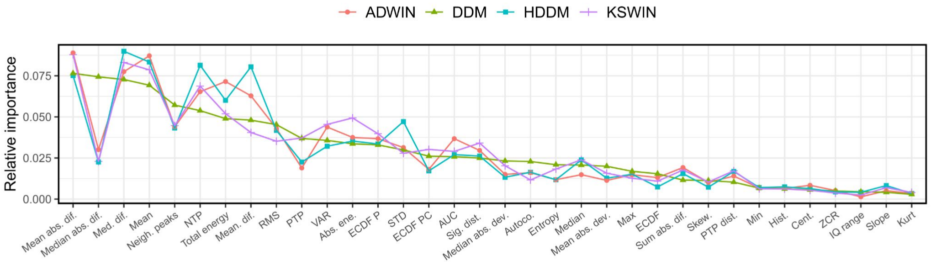
- When compared to Majority, META achieves the best G-Mean for at least 50% of the time in 6 of 9 evaluated datasets
- In comparison to Random, META is outperformed only in the tripadvisor dataset.
- These results demonstrate how META outperforms the baselines in almost every evaluated dataset.





Results

- We employed the RF Feature Importance to examine the contribution of each feature in selecting the most suitable drift detector
- The analysis reveals that the meta-models share a set of highly important features
    - Including Histogram, Centroid, Median, Mean Difference, and Skewness
- Conversely, features such as Autocorrelation, Absolute Energy, Peak to Peak distance, and ECDF are among the less important meta-features



Results

- The results showed that our proposed method effectively recommends the best drift detector for an unseen data chunk (**RQ1**)
- We can confidently state that META showcases the capability to handle certain imbalance scenarios without requiring a specific mechanism for skewed data (**RQ2**)
- The results in real-world datasets demonstrate how META outperforms the baselines in almost every evaluated dataset (**RQ3**).

Conclusion and future work

- We addressed the challenges of learning from imbalanced data streams under concept drift while incorporating algorithm recommendations
- We proposed an online Meta-Learning framework that dynamically recommends the most suitable drift detector for an unseen chunk of data
- Through extensive experiments, we demonstrated the effectiveness of our framework compared to the baselines with and without class imbalance.
  - Supporting our hypothesis that dynamically changing the drift detector can significantly improve predictive performance
- We plan to explore the application of Meta-Learning for hyperparameter tuning, ensemble classifier selection, and adjusting Active Learning constraints.

Conclusion and future work

- If you are interested in this work, you should check out more about imbalance data streams
  - Aguiar G.; Krawczyk B.; Cano. A. A survey on learning from imbalanced data streams: taxonomy, challenges, empirical study, and reproducible experimental framework.