# [TDATAAI962] Data Stream Mining

# Final projects

The goal of this project is to build teams of 3 people to work together on an open-source implementation of streaming algorithms that can be used with river in Python.

https://riverml.xyz/latest/

The deliveries of the project are:

1. Source code following scikit-learn guidelines:
    a. http://scikit-learn.org/stable/developers/contributing.html#coding-guidelines
2. Documentation following scikit-learn guidelines:
    a. http://scikit-learn.org/stable/developers/contributing.html#documentation
3. Presentation slides with experimental results

Refer here for additional guidelines specific to scikit-multiflow

**Note:** Teams whose implementation displays high-quality may be invited to contribute their code to the project on GitHub. This contribution is voluntary and unrelated to the grading of your project.

---

## [1] Ensemble of Restricted Hoeffding Trees

An ensemble method that combines the predictions of multiple models using stacking. The base-level models are Hoeffding Trees trained on different attribute-subsets and the meta-level learners are perceptrons (one per class value) trained on class probabilities from the base-level models.

[Paper]

---

## [2] On-line Regression/Model Trees with Options

Uses the idea of options nodes to speed up the slow splitting process in the original FIMT-DD algorithm. Option trees build upon regular trees by adding splitting options in the internal nodes. As such they are known to improve accuracy, stability and reduce ambiguity.

**Implementation note:** scikit-multiflow currently does not have an implementation of the FIMT-DD but includes two closely related methods: Hoeffding Tree Regressor and Hoeffding Adaptive Tree Regressor.
**Extra points:** FIMT-DD implementation (extended from existing code).

[Paper]

---

## [3] Hoeffding Option Tree

Hoeffding Option Trees are regular Hoeffding Trees containing additional option nodes that allow several tests to be applied, leading to multiple Hoeffding trees as separate paths.

[Paper]

---

## [4] Unsupervised anomaly/outlier detection

Like drift detection, except anomalies are a single point. Stream-based clustering methods could be used, by adapting k-means or KNN classes. If a data point/test instance falls far from a known cluster, it could be considered as an anomaly. Requires data set with known anomalies, or modification of an existing dataset or a generator to insert them (so as to evaluate detection accuracy). Visualization could be an interesting addition.

[Paper, (esp. Section 6)]

---

### [5] Time series forecasting

Build a sliding window to create a stream of pseudo-instances. Use these instances as input to predict future values in the stream. Could be implemented as a wrapper/transformation method to which any off-the-shelf data stream learner can be applied, e.g., Hoeffding Tree. Can also use an echo state network instead of a sliding window to produce the meta stream. Apply it to any real world data (e.g., Electricity dataset). How far into the future can you accurately predict?

[Paper]

---

# Team Proposal (deadline 18/5/2021)

Write below the name of team members (one team per line) and your bid for at least 3 projects (highest interest first).

Project assignments will be announced after all teams have submitted their bids.

| **Team members** | **Project bid** |
|---|---|
| Gurami Keretchashvili, Temur Malishava, Guillaume Barthe | 4,5,3 |
| Moïne , sini suresh, Akshaya | 2, 5, 3 |
| | |
| | |
| | |
| | |

# Team Assignments

| **Team members** | **Project bid** |
|---|---|
| Gurami Keretchashvili, Temur Malishava, Guillaume Barthe | 4 |
| Moïne , sini suresh, Akshaya, Yi | 2 |
| | |
| | |
| | |