STReam Insight and Drift Explanation (STRIDE): a Python Toolkit for Concept Drift Detection and Explanation

Deniz Aksoy, Kuba Czech, Wojciech Nagórka, Michał Redmer, Jerzy Stefanowski

Poznań University of Technology, Institute of Computing Sciences

Abstract. STRIDE is a new system implemented in Python for analyzing and explaining the causes of drift in data streams. In particular, its interactive dashboard and integration of signals from the data, model, and XAI layers allow analysts to perform a comprehensive diagnosis of different drifts. It also enables them to discover which specific regions of the feature space have undergone changes and how they affect classifier performance.

1 Introduction

Contemporary machine learning applications increasingly process data in the form of streams characterized by potentially infinite data volume and variability of their properties over time [2]. A key challenge in their analysis is the phenomenon of **concept drift**, a situation in which the data distribution changes over time, leading to deteriorating performance of predictive models trained on earlier data [3].

Most research focuses on drift detection and model adaptation rather than on attempting to characterize and understand the underlying causes of changes in the data and model predictions. Standard drift detectors typically do not indicate which specific regions of the feature space have undergone transformation, nor how these local changes affect classifier performance.

In STRIDE we follow a more comprehensive approach by providing a simultaneous analysis of changes before and after a detected drift across three distinct levels (layers): the Data Layer (statistical analysis, tests, and clustering), the Model Layer (various prediction drift detectors), and the Explanation Layer (appropriately adapted eXplainable AI methods for analyzing changes in model predictions). To support such comprehensive analyses, we have implemented an original software framework designed to assist both scientific researchers and data analysts working on real-world tasks.

2 System Overview

Unlike traditional tools that offer isolated detectors, STRIDE unifies the following approaches to ensure state-of-the-art monitoring capabilities:

– **Data Layer**: Employs descriptive statistics, selected unsupervised statistical tests (Kolmogorov-Smirnov or Anderson-Darling), probability divergences, and X-means clustering to detect shifts in the input distribution and visualize them.
– **Model Layer**: Utilizes performance-based triggers such as the Drift Detection Method (DDM) and its extensions to signal concept drifts.
– **Explanation Layer**: Bridges the gap between detection and understanding via feature importance monitoring, decision boundary analysis, and prototype-based explanations, revealing how the model's logic evolves.

Each method is designed to function autonomously, yet their true power is realized when utilized together, providing the user with a comprehensive view of different aspects of the drift's nature. STRIDE is accessible via two primary interfaces: a developer-friendly **Python API** for seamless integration into existing ML pipelines, and an **Interactive Dashboard** for real-time visual analytics.

The architectural flow of STRIDE follows a synchronous processing pipeline (Fig. 1). Rather than triggering explanations only upon drift detection, the framework concurrently processes each incoming data block (or window) across all three layers. This parallel execution ensures that statistical shifts in the Data Layer, performance metrics in the Model Layer, and interpretability insights in the Explanation Layer are continuously synchronized. Such a design allows the user to observe the evolution of model behavior and its explanations in real-time, even before a formal drift alert is issued by the detectors.

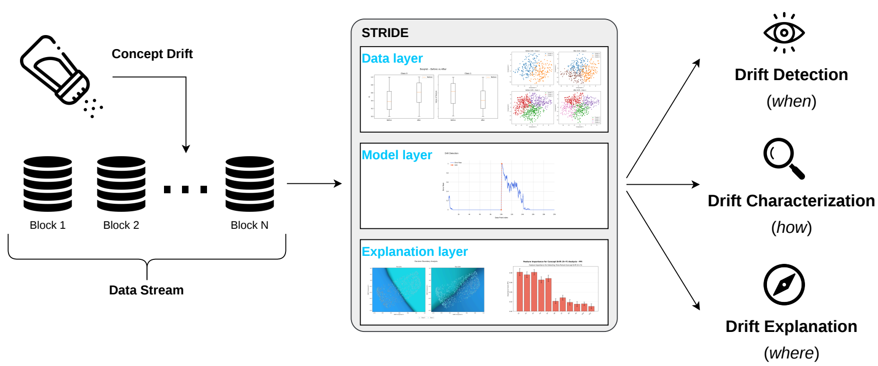


Fig. 1: The STRIDE synchronous processing pipeline. Data batches are processed concurrently across Data, Model, and Explanation layers to provide real-time, synchronized insights before and after drift detection.

The framework is implemented in Python, leveraging River [1] for stream processing and drift detection, while scikit-learn and pandas handle batch-based baseline calculations and data manipulation. Numerical efficiency is ensured by NumPy. The interactive dashboard is built using Streamlit, enabling a reactive web interface that handles real-time data flow with low latency. Data visualizations are rendered via Seaborn and Matplotlib, providing high-quality plots.

All methods come with default configurations, experimentally tuned for optimal performance and explanations. Advanced users can override these settings via the API to adjust sensitivity or certainty levels. The object-oriented design and open architecture ensure that STRIDE is easily extensible with new algorithms, metrics, or visualization modules.

3 Case Study

We demonstrate STRIDE's capabilities on the popular hyperplane synthetic dataset, featuring a rotating decision boundary and utilizing a neural network as a black-box classifier. The generated data stream is partitioned into 20 windows of 1000 instances each. First, it is necessary to identify the moment of the drift. The DDM detector approximates it at instance no. 10030, which is also confirmed by comparing changes in the distribution of prototype explanations. Therefore, for subsequent analyses, we compare window no. 9 (just before the drift) with window no. 10 (just after the drift).

Using descriptive statistics from the Data Layer, the class-conditional feature distributions reveal a shift in feature values, particularly in the x_1 feature. This is confirmed by larger values of the Wasserstein distance (0.334 for the x_1 feature, compared to 0.104 for x_2). Detailed results of the cluster analysis also show that both Class 0 and Class 1 evolved from 2 to 3-4 clusters each, indicating that the rotating decision boundary affected both classes approximately symmetrically.

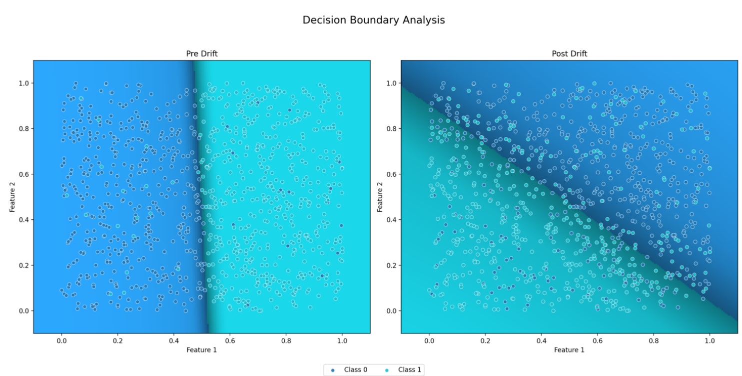
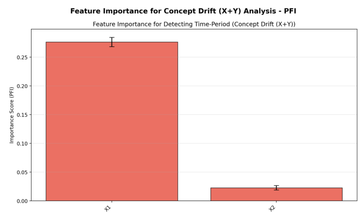
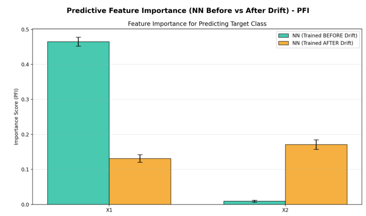


Fig. 2: Pre and Post Drift Decision Boundaries (Window 9 vs 10)

Examining the XAI Supervised Decision Boundary Map for model predictions before and after the drift (see Figure 2), one can clearly notice a change in this boundary between classes. This corresponds to the rotation of the hyperplane in the original feature space.



(a) Visualization of the Model-Centric approach.



(b) Visualization of the Data-Centric approach.

Fig. 3: Visualization of Feature Importance analysis for the detected drift.

Finally, a feature importance analysis of the model's predictions (see Figure 3b) shows that at first, the model did not consider x_2 valuable, as x_1 could better separate the decision classes. However, as the decision boundary rotated, x_1 became less valuable while the importance of x_2 grew significantly. This aligns with the takeaways from the decision boundary analysis. Figure 3a further suggests that the drift affects x_1 more strongly than x_2, reflecting our takeaways from the Data Layer.

4 Conclusions

We presented STRIDE, an integrated environment that simultaneously orchestrates established analytical methods across three functional layers to provide a novel, multi-perspective diagnostic of streaming data. To the best of our knowledge, STRIDE is a unique software solution capable of delivering such a comprehensive analysis of the underlying causes of concept drift. The framework is open-source, inviting further community-driven extensions.

Additional information

Webpage and documentation: https://michalredm.github.io/stride-website/, GitHub: https://github.com/KubaCzech/STRIDE

References

1. Montiel, J., Halford, M., Mastelini, S.M., Caballero, G., Bifet, A.: River: machine learning for streaming data in python. Journal of Machine Learning Research **22**(110), 1–5 (2021)
2. Read, J., Zliobaite, I.: Supervised learning from data streams: an overview and update. ACM Computing Surveys **57**(12), 1–31 (2025)
3. Webb, G.I., Hyde, R., Cao, H., Nguyen, H.L., Petitjean, F.: Characterizing concept drift. Data Mining and Knowledge Discovery **30**(4), 964–994 (2016)