In21-S5-CS3501 Data Science and Engineering Project
Department of Computer Science and Engineering
University of Moratuwa

Comparative Evaluation of Data Drift Detection Algorithms: A Benchmarking Study

Team Members:
Nadawa R.M.N. 210401T
Nanayakkara A.H.M. 210404F
Mendis P.H.M.N. 210386A

Group ID: 17
Project ID: 06
Mentor: Dr. T. Uthayasanker
Teaching Assistant: Mr. Dileesha Kannangara

1. Executive Summary:

This project aims to develop a comprehensive benchmarking system to evaluate data drift detection algorithms, involving the selection of key algorithms and the creation of a robust experimental design using both real and synthetic datasets. Here we find the machine learning algorithms and train machine learning models to detect the concept drift of datasets and make accurate predictions.

2. Problem Statement:

The project aims to solve the challenge of evaluating data drift detection algorithms, which is crucial for maintaining the accuracy and reliability of machine learning models in dynamic environments. Without effective drift detection, models can degrade over time, leading to poor decision-making and reduced trust in AI systems. This problem affects data scientists, machine learning engineers, and organizations relying on predictive models for critical business decisions. Developing a comprehensive benchmarking system will provide clear insights into algorithm performance, helping stakeholders choose the best methods to ensure model robustness and trustworthiness.

3. Data Description:

(1). Synthetic Datasets

The FriedmanDrift dataset generator creates a synthetic dataset with concept drifts for testing and evaluating machine learning algorithms. This dataset is based on the Friedman synthetic dataset and introduces various types of concept drift to simulate changes in the data distribution over time. Following are some details of

Features:

- Number of Features: Each observation consists of 10 features.
- Feature Values: Each feature value is uniformly sampled from the interval [0, 1].
- Relevant Features: Only the first 5 features are relevant for predicting the target variable.

Target Variable:
The target variable is defined by different functions, depending on the type of drift, which determines how the relationship between the input features and the target changes over time.

Modes of Operation:
The dataset generator supports three modes of operation, each corresponding to a different type of concept drift:

1. Local Expanding Abrupt Drift (lea)
    - Concept drift appears in two distinct regions of the instance space. The remaining regions are left unaltered. There are three points of abrupt change in the training dataset. With each consecutive change, the regions of drift expand.

2. Global Recurring Abrupt Drift (gra)
    - Concept drift affects the entire instance space. There are two points of concept drift. At the second point, the old concept reoccurs.

3. Global and Slow Gradual Drift (gsg)
    - Concept drift affects the entire instance space gradually. There are two change points. During a transition window of length transition_window, examples from both old and new concepts are generated with equal probability. After the transition period, only examples from the new concept are generated.

(2). Real-world datasets

New York Taxi Trip Data - The Dataset consists of NYC taxi trip data. This dataset provides a comprehensive record of urban mobility, detailing each taxi trip with precision. It includes temporal information such as pickup

and drop-off times, and spatial data including specific locations. The dataset offers an in-depth view of urban transit patterns, featuring quantitative metrics like trip distance, fare breakdown, payment methods, and passenger counts.
A full description of the dataset is available here:
https://www.kaggle.com/datasets/microize/newyork-yellow-taxi-trip-data-2020-2019?select=taxi%2B_zone_lookup.csv

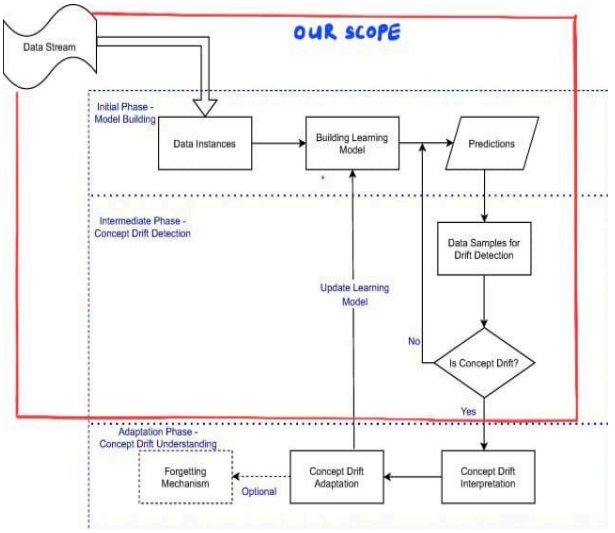
4. Methods:

1. Data Cleaning

- **Handling Missing Values:** Use Pandas to identify and fill or remove missing values.
- **Outlier Detection:** Identify and handle outliers using statistical methods or visualisation tools like Matplotlib and Seaborn.
- **Data Normalization:** Normalize the dataset to ensure uniform scaling using Scikit-learn's preprocessing utilities.

2. Data Exploration

- **Descriptive Statistics:** Use Pandas to compute mean, median, standard deviation, etc.
- **Data Visualization:** Utilize Matplotlib and Seaborn to create plots (e.g., histograms, box plots, scatter plots) to visualize data distributions and relationships.
- **Correlation Analysis:** Use heatmaps to examine correlations between features and target variables.

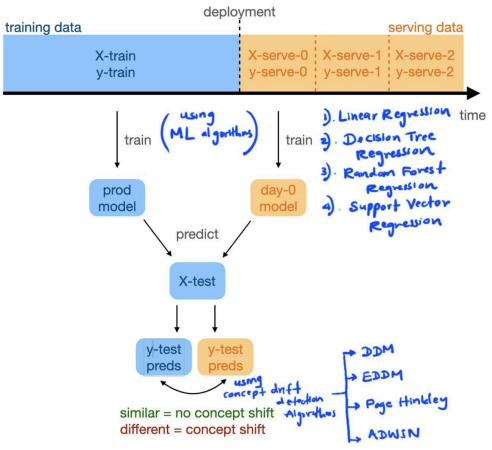


3. Data Analysis

- **Feature Engineering:** Create new features or transform existing ones to improve model performance using Pandas.
- **Dimensionality Reduction:** Apply techniques like PCA (Principal Component Analysis) using Scikit-learn to reduce the number of features while retaining important information.
- **Statistical Analysis:** Conduct hypothesis testing using Statsmodels to understand the significance of features.

4. Building Model

- **Model Training:** Use Scikit-learn to train various regression models:
  - Linear Regression
  - Decision Tree Regression
  - Random Forest Regression
  - Support Vector Regression
- **Model Evaluation:** Evaluate models using metrics such as RMSE (Root Mean Square Error), MAE (Mean Absolute Error), and R^2 score.
- **Hyperparameter Tuning:** Optimise model performance using techniques like Grid Search or Random Search provided by Scikit-learn.



5. Concept Drift Detection

- **Data Stream Simulation:** Generate a continuous data stream using Pandas, where the concept drift occurs at a specific point.
- **Drift Detection Methods:** Implement and evaluate the following methods using the frouros library:
    - DDM (Drift Detection Method)
    - EDDM (Early Drift Detection Method)
    - Page Hinkley
    - ADWIN (Adaptive Windowing)
- **Performance Evaluation:** Assess the performance of each drift detection method in terms of detection accuracy.

5. Expected Outcomes and Success Criteria:

Expected Outcomes

1. **Evaluation of Drift Detection Methods:**
    - Performance metrics for each method (DDM, EDDM, Page Hinkley, ADWIN) with different machine learning models.
    - Insights into strengths and weaknesses under various conditions.
2. **Comparison of Machine Learning Models:**
    - Performance comparison of regression models (Linear, Decision Tree, Random Forest, Support Vector) in the presence of concept drift.
    - Identification of models that are more robust to drift.
3. **Deliverables**
    - Functional Python scripts which can replicate the experiments carried out
    - Fully comprehensive report explaining the findings (comparative studies/ discussions/ conclusions)
    - Dashboard that showcases the results and insights visually

Success Criteria

1. **Performance Metrics:**
    - Successful identification of the best-performing drift detection methods and machine learning models under various conditions.
    - Clear, quantitative performance metrics (e.g., detection accuracy, false positive rate, detection delay) for each combination of drift detection method and machine learning model.

6. Preliminary Bibliography:

1. Learning under Concept Drift: A Review By Jie Lu, Fellow, IEEE, Anjin Liu, Member, IEEE, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang

2. Frouros: Sample of similar tools: https://github.com/IFCA-Advanced-Computing/frouros

3. DeepChecks: A package for comprehensively validating machine learning models and data. It includes functionalities for checking data integrity, data drift, and model performance. (URL: https://github.com/deepchecks/deepchecks)

4. Handling Concept Drifts in Regression Problems – the Error Intersection Approach By Lucas Baier1, Marcel Hofmann2, Niklas Kühl1, Marisa Mohr2,3 and Gerhard Satzger