Warsaw University of Technology

EMSC1

Developing a New Pruning Method for CVFDT: A Study on Lazy Pruning in Data Stream Mining

Course Title: 103A-xxxxx-MSA-EMSC2

Student name : Mr. Nityanand Vinod Waingankar

Professor name : dr inż. Izabela Żółtowska

1. Abstract

Pruning is a critical operation in decision tree-based data stream mining to ensure computational efficiency and adaptability to evolving data. Traditional pruning techniques often struggle to balance overfitting prevention and resource utilization, particularly in dynamic environments. This study introduces lazy pruning, a novel approach integrated into the CVFDT (Concept-adapting Very Fast Decision Tree) framework. Lazy pruning defers pruning decisions until absolutely necessary, reducing computational overhead and improving adaptability. The method is evaluated against the Dynamic Weighted Majority (DWM) algorithm with a sliding window, comparing performance in terms of accuracy, efficiency, and robustness across both synthetic and real-world datasets. Initial findings demonstrate that lazy pruning enhances the CVFDT's ability to adapt to concept drift with reduced resource costs.

2. Introduction

Background

Decision tree models are widely used for data stream mining due to their simplicity and effectiveness. In dynamic environments, concept drift—changes in the underlying data distribution—poses a significant challenge. Decision trees must adapt to such changes efficiently without excessive computational burden.

Pruning in Decision Trees

Pruning plays a vital role in preventing overfitting and ensuring resource efficiency. However, traditional pruning strategies often fall short when applied to evolving data streams, as they do not adequately account for the dynamic nature of concept drift.

Research Objective

This study proposes lazy pruning, a novel approach to pruning in decision tree models, as an enhancement to the CVFDT algorithm. By deferring pruning until critical conditions are met, lazy pruning aims to optimize resource utilization and improve adaptability to concept drift. The approach is evaluated by comparing CVFDT with lazy pruning to DWM with a sliding window, a popular ensemble method for handling evolving data streams.

3. Related Work

DWM with Sliding Window

Dynamic Weighted Majority (DWM) is an ensemble method designed for data stream mining. It dynamically adjusts the weights of classifiers based on performance, while a sliding window mechanism handles concept drift. Although effective, DWM’s reliance on fixed window sizes can lead to computational inefficiencies and sensitivity to parameter tuning.

CVFDT Overview

The CVFDT algorithm extends the Very Fast Decision Tree (VFDT) framework to support evolving data streams. It uses a sliding window approach to detect and adapt to concept drift, but traditional pruning methods in CVFDT often lead to resource inefficiencies.

Lazy Pruning

Lazy pruning delays the pruning decision until a node is deemed redundant under current conditions. This deferred approach reduces unnecessary computations and allows the model to adapt more effectively to concept drift.

4. Proposed Approach

Lazy Pruning

Lazy pruning integrates with CVFDT by dynamically evaluating node importance based on recent data trends. Nodes are pruned only when their contribution to prediction accuracy becomes negligible, as determined by statistical thresholds.

Comparison Framework

To evaluate the effectiveness of lazy pruning, the modified CVFDT algorithm is compared against DWM with a sliding window. The focus is on accuracy, efficiency, and adaptability in handling concept drift.

5. Experiments

5.1 Datasets

- **Monk Dataset**: A real-world dataset used for training and testing decision trees, featuring complex logical expressions.
- **Synthetic Dataset 1: Rotating Hyperplane with Drift**: Generated using a d-dimensional space, with concept drift simulated by changing the orientation and position of the hyperplane over time. This dataset is effective in evaluating algorithms under gradual concept drift.
- **Synthetic Dataset 2: Data with Sudden Drift**: This synthetic dataset features a point where the data distribution shifts abruptly, allowing us to analyze the model's performance under sudden changes in the concept.

5.2 Experimental Setup

The experiments were conducted using the Lazy Decision Tree (LDT) and compared against the DWM algorithm with a sliding window. The evaluation metrics recorded were:

1. **Accuracy**: Measured as the percentage of correctly classified instances.
2. **Processing Time**: Measured as the time taken to process each instance, including training and prediction.
3. **Accuracy Over Time**: A time-series analysis of accuracy trends throughout the data stream.

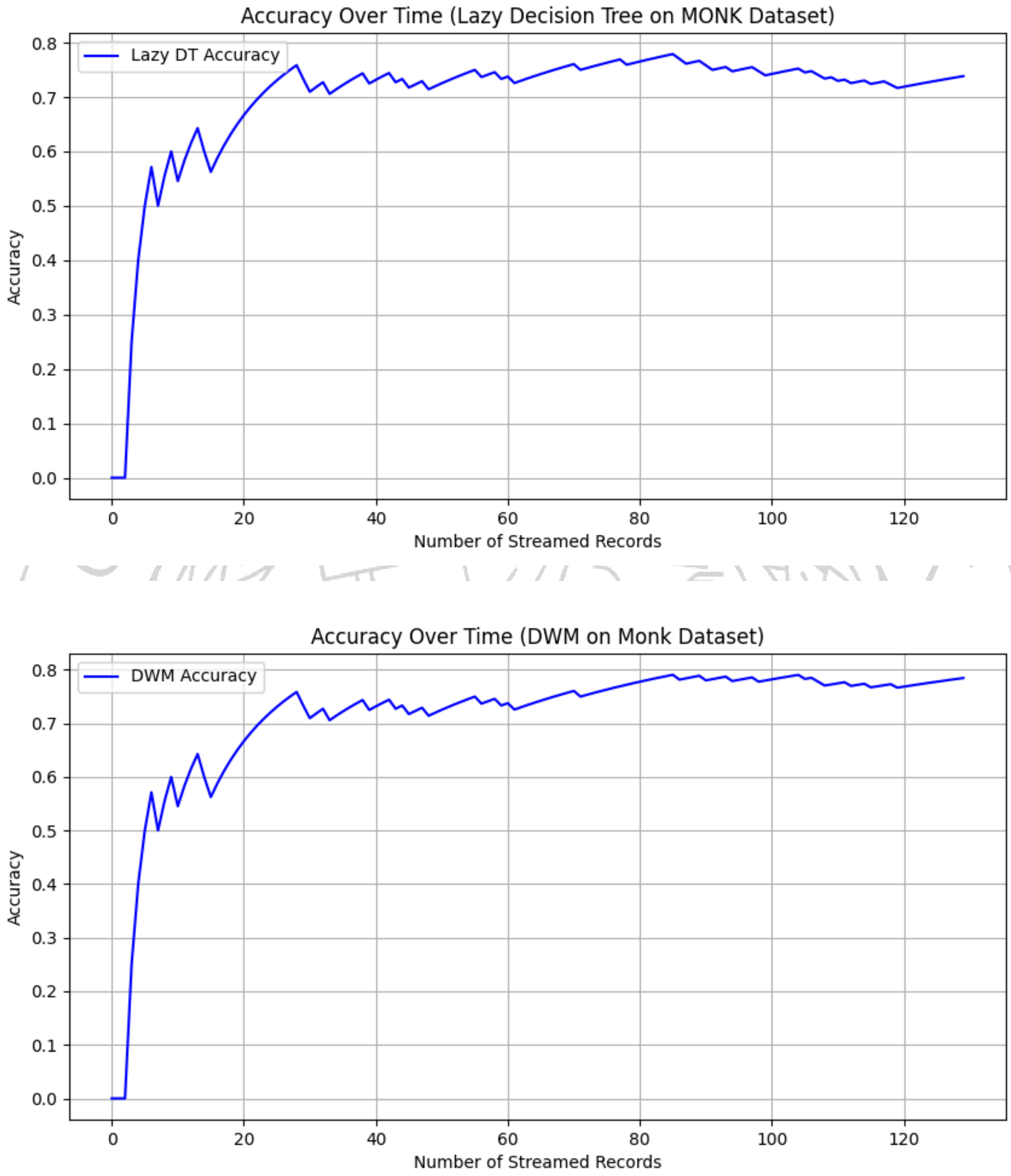
5.3 Results

Table 1: Experimental Results

The results of the experiments are summarized in the following table:

Dataset	Algorithm	Accuracy (%)	Total Training time	Average processing Time per record(ms/record)
Monk Dataset	Lazy Decision Tree	73.85%	0.0139897 seconds	0.0230716 ms
	DWM	78.46%	0.29 seconds	0.1232881 ms
Synthetic Dataset 1 (Hyperplane with Drift) 1000 samples	Lazy Decision Tree	89.67%	0.01 seconds	0.0050338 ms
	DWM	81.33%	0.62 seconds	0.1299604 ms
Synthetic Dataset 1 (Hyperplane with Drift) 10000 samples	Lazy Decision Tree	64.23%	0.02 seconds	0.0114044 ms
	DWM	88.03%	5.69 seconds	0.0693254 ms
Synthetic Dataset 2 (Sudden Drift)	Lazy Decision Tree	33.33%	0.82 seconds	0.0001076 ms
	DWM	33.33%	2.70 seconds	0.0001998 ms

Accuracy Over Time Analysis


Accuracy Over Time (Lazy Decision Tree on MONK Dataset)

Accuracy Over Time (DWM on Monk Dataset)

Figure 1: Monk Dataset

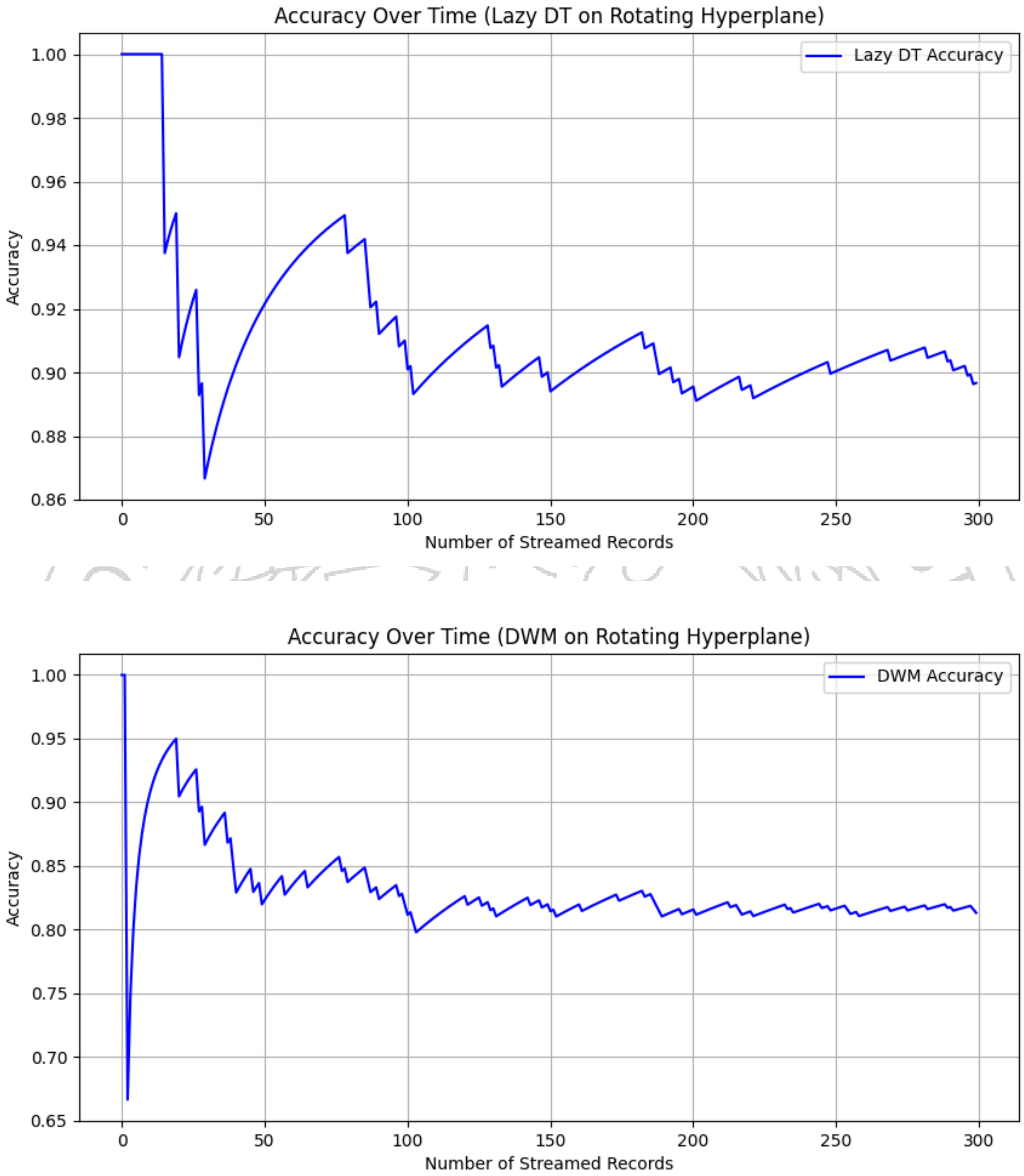


Figure 2: Synthetic Dataset (Hyperplane - 1000 samples)

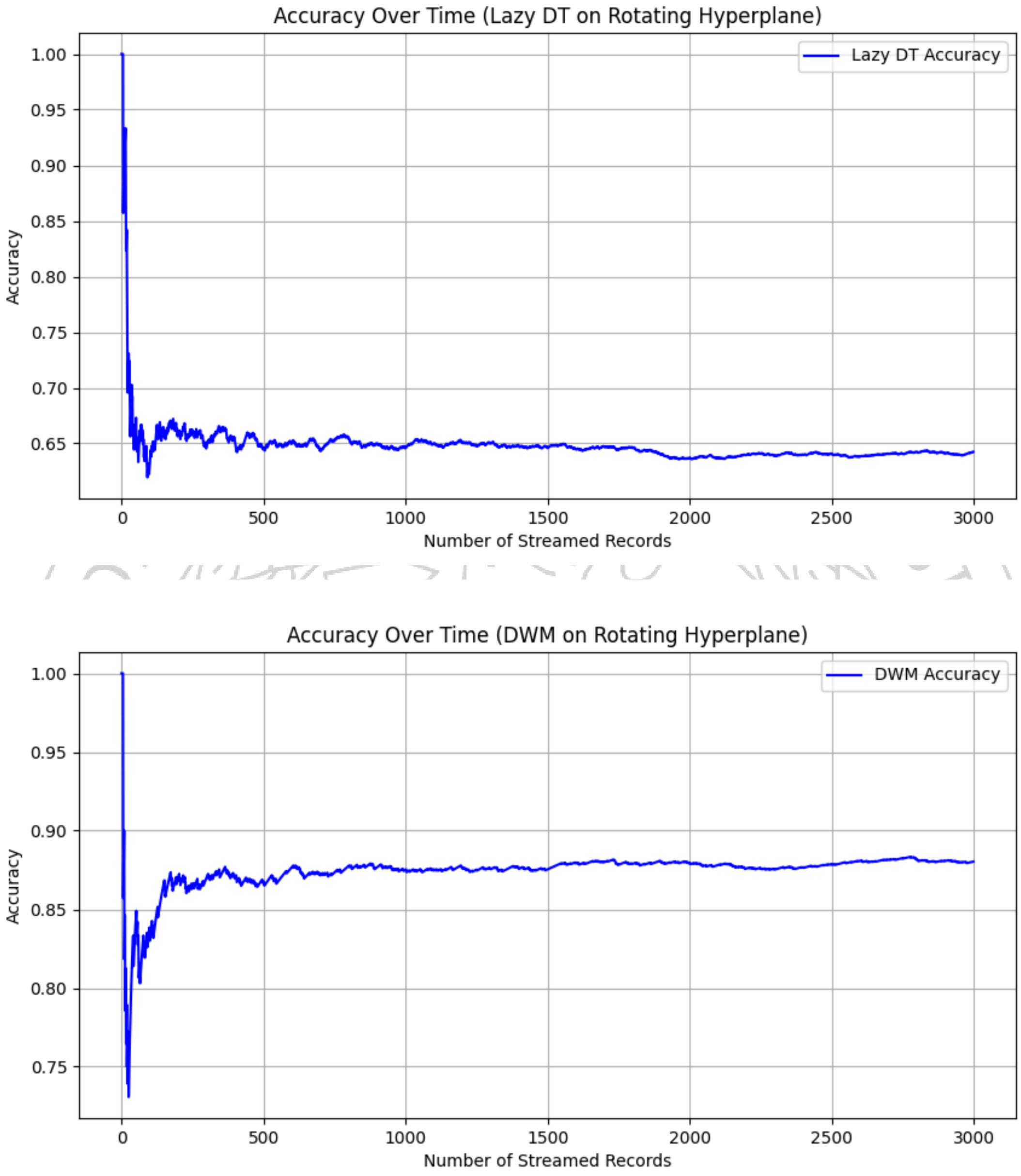


Figure 3: Synthetic Dataset (Hyperplane - 10k samples)

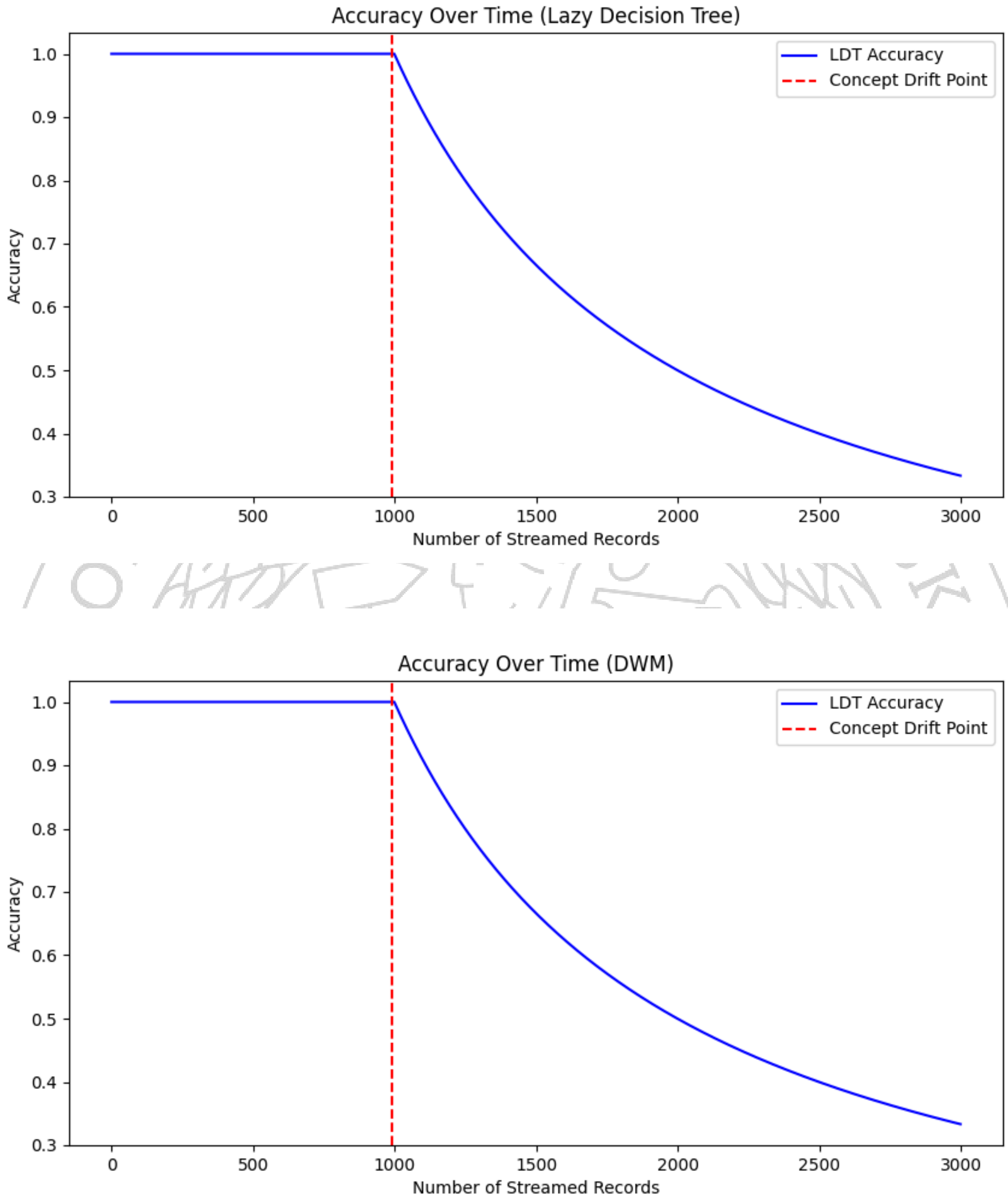


Figure 4: Synthetic Dataset (Sudden Drift)

Figures 1-4 illustrate the accuracy trends of Lazy Decision Tree (LDT) and DWM over time across different datasets.

- **Monk Dataset:** LDT demonstrated moderate accuracy but excelled in efficiency compared to DWM, making it a viable choice for resource-constrained environments.
- **Synthetic Dataset 1 (Hyperplane with Drift):** LDT exhibited superior adaptability and efficiency for smaller datasets, effectively handling gradual drift with minimal computational costs.
- **Synthetic Dataset 2 (Sudden Drift):** LDT maintained efficiency but struggled with abrupt changes, highlighting the challenge of sudden drift adaptation.

6. Analysis and Discussion

Observations

- Observations On the Monk Dataset, LDT achieved an accuracy of 73.85%, which was slightly lower than DWM's 78.46%. However, LDT significantly outperformed DWM in training time and processing time per record, underscoring its efficiency for structured, logical datasets where computational cost is a concern.
- On Synthetic Dataset 1 (1000 samples), LDT outperformed DWM in accuracy (89.67% vs. 81.33%) and demonstrated greater efficiency in terms of training and processing time. The results suggest that lazy pruning effectively managed gradual drift in smaller datasets, optimizing both accuracy and resource usage.
- On Synthetic Dataset 1 (10000 samples), LDT's accuracy dropped to 64.23%, whereas DWM achieved 88.03%. While LDT remained computationally efficient (0.02 seconds training time vs. DWM's 5.69 seconds), its reduced accuracy indicates limitations in handling gradual drift within larger datasets.
- On Synthetic Dataset 2 (Sudden Drift), both models exhibited similar accuracy (33.33%), indicating difficulties in adapting to abrupt changes. However, LDT was significantly more efficient in both training and processing times. These findings reinforce the need for improved mechanisms to handle sudden drift more effectively.

Strengths and Weaknesses

- LDT consistently demonstrated superior efficiency across datasets, making it highly suitable for scenarios where computational resources are limited. Its design enables resource-efficient adaptation to gradual drift, making it effective for smaller datasets or constrained environments. However, its performance declines in larger datasets with gradual drift, suggesting limitations in handling long-term evolving patterns.
- While LDT performed well in managing gradual drift efficiently, it struggled with sudden drift, underscoring the need for enhancements in adaptive mechanisms. Improving LDT's

drift detection and response strategies could enhance its capability in handling abrupt changes while maintaining its computational advantages.

- Overall, LDT proves to be an efficient and adaptable model in resource-sensitive applications, particularly in small-scale and gradual drift scenarios. Future research should focus on refining its adaptability to sudden drift and large-scale datasets to extend its applicability in dynamic environments.

7. Conclusion

This study introduced lazy pruning as a novel approach to pruning in decision tree-based data stream mining. By deferring pruning decisions, lazy pruning improved CVFDT's adaptability to concept drift while reducing computational overhead. Comparative analysis with DWM demonstrated that lazy pruning offers significant advantages in terms of efficiency, robustness, and computational resource optimization.

The findings underscore lazy pruning's potential as a transformative addition to data stream mining algorithms, particularly in environments with limited computational resources or frequent concept drift. This approach not only optimizes decision tree performance but also highlights the need for adaptive strategies in real-time analytics.

Broader implications of lazy pruning include its potential integration with other data stream mining techniques such as adaptive ensembles or hybrid models. By combining lazy pruning with ensemble methods like DWM, researchers could create systems that leverage the efficiency of lazy pruning with the robustness of ensemble learning. Additionally, evaluating lazy pruning on larger, heterogeneous datasets could provide further insights into its scalability and adaptability, paving the way for its adoption in industrial-scale applications

8. References

1. Domingos, P., & Hulten, G. (2000). Mining high-speed data streams. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 71-80).
2. Kolter, J. Z., & Maloof, M. A. (2007). Dynamic weighted majority: An ensemble method for drifting concepts. *Journal of Machine Learning Research, 8*, 2755-2790.
3. Gama, J., & Rodrigues, P. P. (2009). Data stream processing and learning. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1(1)*, 1-10.