Class-Imbalanced Dynamic Feature Selection for Dropout Prediction in Virtual Learning Environments

Ikram Gagaoua, Chahrazed Labba, Armelle Brun

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2025) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

29th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2025)

Class-Imbalanced Dynamic Feature Selection for Dropout Prediction in Virtual Learning Environments

Ikram Gagaoua[a,*], Chahrazed Labba[b], Armelle Brun[b]

[a] *RiseUp, Université de Lorraine, CNRS, LORIA, Paris, France*
[b] *Université de Lorraine, CNRS, LORIA, Nancy, France*

Abstract

Prediction of student dropout is a major challenge for educational institutions. Although AI-based approaches can predict dropout through data analysis, they often rely on a static set of features and fail to adapt to the evolving nature of data over time, thus limiting performance. Other application domains have the same characteristic of evolving data and several algorithms have been proposed to ensure dynamic feature selection over time. However, they are not adapted to dropout prediction, characterized by imbalanced data, which limits their effectiveness. In this paper we introduce CI-DFS, a novel algorithm that achieves dynamic feature selection on class imbalanced data and used to perform dropout prediction. CI-DFS relies on three main elements: 1) a weighted mutual information-based metric to deal with imbalanced class distributions, 2) an adaptive threshold mechanism to dynamically assess feature relevance, and 3) a temporal drift management system to ensure feature relevance over time. CI-DFS has been evaluated on two real-world benchmark educational datasets, and compared with a state-of-the-art dynamic feature selection algorithm. CI-DFS outperforms this algorithm in three key aspects: it improves dropout prediction by 9%, reduces the number of features selected by over 50%, and significantly accelerates feature selection time, making it 10 times faster.

Keywords: Class Imbalance; Feature Drift; Mutual Information; Early Prediction; Data Streaming

1. Introduction

Artificial Intelligence (AI) is increasingly being used to tackle the student dropout prediction challenge by harnessing a variety of data sources, such as academic results, engagement measures and behavioral patterns [1, 2, 3]. The aim is to develop predictive models capable of identifying at-risk students as early as possible, enabling institutions to intervene proactively. However, these AI models often rely on a predefined, static set of features, making these models less suitable to the dynamic nature of online educational data. As student behaviors, learning environments

* Corresponding author.
E-mail address: ikram.gagaoua@riseup.ai

and external factors evolve, some features may have a relevance that changes over time, and sometimes become less informative [4]. Further, new features may emerge, offering valuable information that static feature selection models fail to incorporate. For example, in dropout prediction, engagement characteristics may initially be strong predictors, but as the semester progresses, attendance may become more relevant [5]. Thus, dynamic selection of relevant features over time is essential to ensure that predictive models remain accurate and robust to the evolving nature of educational data. To the best of our knowledge, dynamic feature selection was never applied for educational purposes.

Other domains dealing with evolving data have developed various approaches for Dynamic Feature Selection (DFS). Among them, Mutual Information (MI)-based approaches [6, 7, 8, 9] have proven to be particularly effective, as they efficiently assess feature relevance and redundancy in evolving data environments. However, MI-based approaches used for DFS within a classification task, generally overlook class imbalance, which can significantly affect their performance for underrepresented class labels. This bias is obvious in educational data, where dropout cases are often underrepresented, leading to a stronger focus on the dominant non-dropout class. Further, they rely on fixed relevance thresholds, which do not adapt to the evolving nature of the data. Additionally, most of these approaches do not take into account feature drift that occurs when the relationships between features and the target variable change over time with the inclusion of new data.

To address these limitations, we present the Class Imbalanced Dynamic Feature Selection (CI-DFS) algorithm, based on weighted mutual information and designed to perform DFS, within a classification task in the presence of imbalanced class data. CI-DFS exploits three components: 1) a class-imbalanced weighted mutual information metric that considers feature relevance across classes, 2) an adaptive threshold approach that dynamically assesses the relevance of new features, and 3) a temporal drift management system used to ensure the relevance of the previously inserted features. CI-DFS is evaluated on two benchmark educational datasets (XuetangX[1] and OULAD[2]), and compared against a naive baseline approach that performs no feature selection and against SAOLA [7], one of the most referenced MI-based algorithms for dynamic feature selection. The results demonstrate that CI-DFS consistently outperforms existing methods. First, it improves dropout prediction sensitivity, ensuring better identification of at-risk students at an early stage. Second, it reduces the number of selected features, while maintaining predictive performance. Finally, it accelerates feature selection, making it particularly suitable for real-time and large-scale educational applications.

The rest of the paper is organized as follows. Section 2 presents the related work. The proposed CI-DFS is introduced in Section 3. Section 4 presents the experimental protocol, followed by the experiments and results in Section 5. Finally, Section 6 concludes the paper and presents directions for future work.

2. Related Work

Numerous studies are dedicated to school dropout prediction with machine learning [10, 5, 1, 2, 3]. They rely on static feature sets, predefined offline, i.e. features that remain unchanged throughout the prediction process. This lack of consideration for the evolution of features, and the dynamic streaming of data through interactions, in the educational domain considerably reduces the effectiveness of these approaches for identifying at-risk students. Indeed, some features become relevant, while others lose their informative nature. In contrast to static approaches, DFS has been extensively developed in other domains with evolving data environments [9] [11] such as healthcare, drug discovery and advertisement, where the changing relevance of features over time is actively considered to update the set of features used for prediction. Among the proposed approaches, we cite Markov blanket-based methods [12] [13], evolutionary algorithms [14], rough set theory [15]. More recently, reinforcement learning has been employed for DFS [16]. However, all these approaches often require extensive training and computational resources, which can hinder their scalability for large-scale or real-time applications.

In turn, Mutual information (MI)-based methods have emerged as an effective alternative for optimizing DFS. For example, OSFSMI [6] selects features by assessing their relevance and redundancy using MI, providing strong performance without relying on complex model assumptions. SAOLA [7], widely recognized as a benchmark approach,

[1] http://moocdata.cn/data/user-activity
[2] https://analyse.kmi.open.ac.uk/open-dataset

adopts an efficient approximation framework to balance relevance and redundancy through pairwise MI comparisons, enabling scalable feature selection in streaming environments. OFS-Density [8] combines MI with density-based approaches to enhance feature selection by considering the structural distribution of data. OSSFS-DD [9] introduces a scalable framework that leverages mutual information and density-based clustering to dynamically add or remove features in streaming environment. Unlike Markov blanket or RL-based approaches, MI-based approaches are computationally efficient, making them suitable for evolving data streams. However, they present some limitations for DFS, including : 1) *Lack of consideration for class imbalance*: MI treats all classes equally when assessing feature relevance. As a result, features that are more informative for the majority class may be favored, while those that are crucial for identifying samples from the minority class, such as in predicting student dropout, may be neglected. 2)*Dependence on fixed relevance thresholds*: fixed thresholds are not adaptive to changing patterns in the data, preventing the model from adjusting its feature selection criteria dynamically. This rigid approach can lead to sub-optimal feature selection as it fails to accommodate evolving data distributions and changing feature importance over time. 3) *Temporal feature drift*: as data evolve over time, feature drift can occur, i.e. features that were highly predictive may become less relevant. Ignoring drift may also limit feature selection performance.

In this work, we focus on the use of MI to ensure DFS in student dropout context, while addressing the identified key limitations.

3. Class-Imbalanced Dynamic Feature Selection (CI-DFS)

In this section, we introduce Class-Imbalanced Dynamic Feature Selection (CI-DFS), a dynamic feature selection algorithm for imbalanced streaming data, detailed in Algorithm 1. At each time step t, CI-DFS is run to update the set of features from the previous step t-1, by inserting new relevant features and discarding irrelevant ones. CI-DFS exploits a fixed predefined set of classes C. CI-DFS processes in two steps. First, it iterates through each candidate feature $f_i \in F_t$ in the input stream, checking f_i's relevance with respect to the given classes using the *check_relevance* function (Line 5). Only features deemed relevant proceed to the next stage, where the *check_redundancy* function evaluates whether f_i provides unique information relative to the existing feature set (Line 7). If f_i is non-redundant, it is added to the selected feature set S_t (Line 9) and its relevance is added to the queue that stores the history of relevance (Line 10). Second, CI-DFS focuses on the newly formed set of features S_t, and applies a drift management technique (Line 15) to ensure that each feature in S_t remains meaningful as the data stream evolves. If not, the associated feature is discarded. We now detail the core components that constitute CI-DFS.

For both steps a threshold is used to assess the relevance of new features to be integrated (Line 3) and to discard old features that have become irrelevant (Line 14). Depending on the type of step ("add" or "discard"), the threshold is determined as follows: i) To add new features, CI-DFS fixes the threshold as the k^{th} percentile of the relevance values in R_t. $k \in [0, 100]$ is a user-defined parameter specifying the percentile threshold; ii) To discard an old feature, CI-DFS fixes the threshold as the $(100 - k)^{th}$ percentile of these relevance values. CI-DFS is available in a github repository[3].

3.1. Evaluating Feature Relevance

Mutual Information (MI), defined in equation (1), is a fundamental concept in information theory that quantifies the statistical dependence between two random variables. In the context of feature selection, MI is used to measure a feature's predictive power by quantifying how much information this feature contains about the target variable [17] [18].

$$MI(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) \cdot \log\left(\frac{P(x,y)}{P(x) \cdot P(y)}\right) \tag{1}$$

[3] https://anonymous.4open.science/r/CI-DFS-5C1B

Algorithm 1 CI-DFS

Require: F_t set of new candidate features at time t; S_{t-1} set of selected features from time t-1; R_{t-1} queue of history of relevance values of features S_{t-1}; C set of classes; λ redundancy threshold; k percentile threshold for feature relevance; w window size.

Ensure: Selected feature set S_t; Set of relevance values R_t

$S_t \leftarrow S_{t-1}$
$R_t \leftarrow R_{t-1}$
// Computes the k^{th} percentile of feature relevance at time t-1
$\theta \leftarrow Percentile_k(R_t)$
for each new feature f_i in F_t **do**
// Check feature relevance - Function 1
    $(relevant, relvalue) \leftarrow$ check_relevance(f_i, C, R_t, θ)
    if *relevant* **then**
// Check feature redundancy with already identified features - Function 2
        redundant $\leftarrow$ check_redundancy(f_i, S_t, λ)
        if not *redundant* **then**
            $S_t \leftarrow S_t \cup \{f_i\}$
            $R_t[f_i] \leftarrow new(relvalue)$
        end if
    end if
end for
// Drift Management - Function 3
$\theta \leftarrow Percentile_{(100-k)}(R_t)$
$(S_t, R_t) \leftarrow$ manage_drift(S_t, C, R_t, w, θ)
return (S_t, R_t)

The evaluation of MI is highly influenced by both the distribution of variables and the probability of values in the variables. In the frame of feature selection, MI struggles with class imbalance, typically biasing towards majority classes. In the dropout prediction problem, one of the two classes (non-dropout) is generally in the majority. To overcome this limitation, we adopt Weighted Mutual Information (WMI), a metric used in the literature [19, 20] to mitigate class imbalance. However, to the best of our knowledge, this metric has not been studied in the context of DFS, only in the frame of static feature selection.

For dropout prediction, we propose to apply weights at the class level to address class imbalance. Concretely, we over-weight the minority class. Equation (2) presents WMI, with a weight dependent of the prior probability of the classes, resulting in an equal importance between classes.

$$WMI(f_i; C) = \sum_{j=1}^{|C|} (1 - p(c_j)) \sum_{v \in val(f_i)} \log\left(\frac{P(v, c_j)}{P(v) \cdot P(c_j)}\right) \tag{2}$$

where v ranges over the set of values of f_i and $1 - p(c_j)$ is the weight of class c_j. Using WMI, we assume that the relevance of each feature will be more precisely assessed and that the downstream prediction task will be more accurate.

3.2. Relevance Evaluation for New Features

Rather than using a static threshold to determine if a feature is relevant or not, as proposed in the literature [7], CI-DFS employs a distribution-based approach to adaptively determine the relevance threshold. This approach seamlessly adjusts to the changing scale and distribution of feature relevance values within the data stream, fostering robust and

adaptive feature selection. For example, if the relevance of the features in the current set of features tends to be high, the threshold should adapt accordingly and be increased. The Function 1 summarizes the relevance computation.

Function 1 check_relevance

Input: f_i the candidate feature; C set of classes; R_t queue of history of relevance values of the features S_t; θ relevance threshold

Output: relevant: boolean for feature relevance; *relevance_value*: relevance value

//Compute WMI value for f_i (Equation 2)

1: $relevance\_value \leftarrow \text{WMI}(f_i, C)$

//Compute dynamic threshold at top of the queues in R_t

2: $relevant \leftarrow relevance\_value > \theta$

3: **return** *relevant*, *relevance_value*

3.3. Reducing Redundancy of the Set of Features

In feature selection, redundancy occurs when two or more features carry similar information, making some of them unnecessary for the prediction task. In line with the literature, in CI-DFS features are considered redundant if they exhibit high correlation with other already selected features [21][22].

For each relevant feature f_i, CI-DFS computes its correlation $\rho(f_i, f_j)$ with each previously selected features $f_j \in S_t$. f_i is considered redundant if its correlation with any other feature exceeds a predefined threshold λ. This approach ensures that each selected feature provides discriminative information, while preventing the accumulation of highly correlated features, limiting the size of the set of selected features. Redundancy checking is defined in Function 2.

Function 2 check_redundancy

Input: f_i the candidate feature; S_t selected features; R_t queue of history of relevance values of the features in S_t; λ redundancy threshold

Output: Boolean: feature redundancy

1: $redundant \leftarrow false$
2: **for** each $f_j \in S_t$ **do**
3: **if** $|\rho(f_i, f_j)| \geq \lambda$ **then**
4: $redundant \leftarrow true$
5: **break**
6: **end if**
7: **end for**
8: **return** *redundant*

3.4. Managing Temporal Feature Drift

In online environments, the relevance of features may change over time, a phenomenon known as feature drift. As a consequence, a given feature, initially relevant, might become less informative through time as data characteristics evolve. Thus, this requires continuous monitoring and adaptation of the selected feature set. CI-DFS addresses the feature drift challenge by using a sliding window of length w. The approach systematically tracks feature relevance evolution by maintaining a historical record of feature relevance values in the w previous time steps (see Algorithm 1, Line 10 and Function 3, Line 3), and computing average relevance. By continuously monitoring these averages and comparing them against an adaptive threshold (see Line 14 in Algorithm 1), the algorithm can remove features that no longer provide meaningful information, ensuring that the selected feature set remains relevant throughout the data stream. The average relevance of feature f_i is defined as the average of the history of the relevance values $R[f_i]$. Drift management is detailed in Function 3.

Function 3 manage_drift

Input: S_t selected features; C classes; R_t queue of history of relevance values of the features S_t; w window size; θ relevance threshold
Output: Updated feature set S

for each feature $f_j \in S_t$ **do**
  $relevance\_value \leftarrow \text{WMI}(f_j, C)$
  $R_t[f_j].push(relevance)$
  if $R_t[f_j].length > w$ **then**
    $R_t[f_j].pop()$
  end if
// Compute the average relevance
  $avg\_relevance \leftarrow avg(R[f_i])$
  if $avg\_relevance < \theta$ **then**
    $S \leftarrow S \setminus f_j$ // feature f_j is discarded from the set of features.
    $R[f_j] \leftarrow$ NULL
  end if
end for
return S

4. Datasets and Evaluation Protocol

4.1. Datasets Description

To evaluate CI-DFS, we conduct dropout prediction experiments, viewed as a classification task, on two real-world virtual learning environments (VLE) datasets. Both datasets are chosen for their representative class imbalance, temporal dynamics and comprehensive feature space.

The XuetangX dataset [23] consists of student interaction logs from a six-week MOOC course. Learning material interaction (features) continuously appear throughout the six weeks. The dataset is provided with a train/test split, made in the context of the KDDCup15[4]. The train set contains 2,613 students and the test set contains 1,087 students. The data exhibits typical VLE class imbalance with dropout rate of 32.7% and 32.3% in the training and test sets respectively.

The OULAD dataset [24], specifically course CCC-2014J which has the highest enrollment (2,498 students), covers a larger period: 269 days (38 weeks). Here again, learning material interactions (features) appear throughout the dataset. The ratio of dropouts is 38.3%, which is close, although slightly higher than on XuetangX.

The datasets provide comprehensive student information including demographic data, course interactions, and academic outcomes. To ensure compliance with data protection principle, we have limited the student information to only essential educational data (highest education level and previous attempts), omitting other demographic variables (gender, region, age band, IMD band, and disability status).

4.2. Feature Processing

We generate weekly interaction features from student logs, resulting in a temporal feature stream that aligns with the course progression. The weekly basis has been chosen as it represents a traditional trade off to track how student behavior evolves. Features represent resources (lectures, exercises, quiz, etc.) and meta information (enrollment date, highest level of education, etc.), feature values with resource features are the interactions with these features.

[4] https://www.kdd.org/kdd2015/calls.html#calls-call-for-kdd-cup

4.3. Evaluation Protocol

To evaluate the effectiveness of CI-DFS, we conduct a comprehensive comparison with two baseline models. CI-DFS is mainly compared against SAOLA [7], a state-of-the-art dynamic feature selection algorithm, may not be the most recent, but it continues to be widely used as a benchmark for comparison in current studies. A baseline model, that processes no feature selection, is also evaluated. We chose not to include static feature selection models in our evaluation, as they do not account for the dynamic nature of educational data in virtual learning environments (VLEs).

After completing the feature selection process using both CI-DFS and SAOLA, their effectiveness is evaluated on the dropout prediction classification task and compared with the *all features* approach. Datasets are prepared following standard machine learning practices by applying a 80-20 train-test split on the OULAD dataset, while utilizing the pre-existing split for the XuetangX dataset. The optimal configuration for CI-DFS was determined through extensive parameter tuning (k=25, λ =0.95, w=10). The reference parameters of SAOLA are maintained, with the fixed relevance threshold set to 0.01, as this algorithm serves as the foundation for CI-DFS's enhanced feature selection approach. Prediction performance is evaluated with a Random Forest classifier.

4.4. Evaluation Metrics

We propose to evaluate DFS models using a comprehensive set of metrics that assess both effectiveness and computational efficiency. First, considering effectiveness, we propose to use the traditional **accuracy** and **sensitivity** metrics [25]. Sensitivity, also known as recall, is the ability to accurately identify dropout students and is defined as:

$$Sensitivity = \frac{\#True\ Positives}{\#True\ Positives + \#False\ Negatives} \tag{3}$$

In this work, Positive means prediction of dropout and Negative means prediction of non-dropout. Considering efficiency metrics, we focus on the **number of selected features** and on the **running time**.

5. Results

5.1. CI-DFS Performance Compared to Baseline Models

This section focuses on the evaluation of the proposed CI-DFS algorithm, with the four metrics presented in Section 4.4, in comparison with baseline models.

Considering accuracy, we observe an increase through weeks on the XuetangX dataset (from 5% to 10%), which reflects the natural benefit of accumulating information. On OULAD, accuracy remains stable across weeks at around 0.65 for all models. CI-DFS performs slightly lower than the other models on XuetangX but slightly better on OULAD, confirming that dynamic selection does not reduce overall predictive performance.

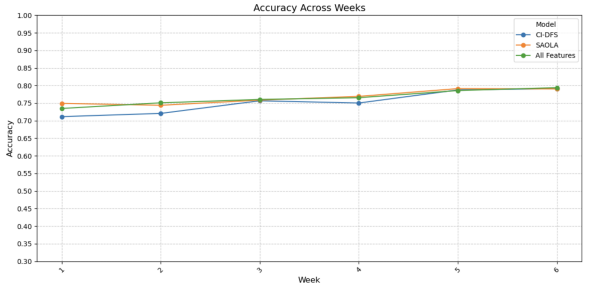

Fig. 1. Accuracy on XuetangX

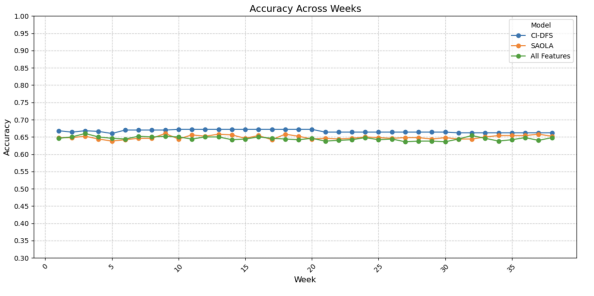

Fig. 2. Accuracy on OULAD

Looking at sensitivity, baseline models show poor results in predicting dropout students, especially in early weeks on XuetangX (under 30%) and consistently on OULAD (45%). CI-DFS consistently outperforms other methods.

On XuetangX, its improvement is especially marked in early weeks (from 50% to 84% compared to SAOLA), which is crucial for enabling early intervention. On OULAD, CI-DFS maintains higher sensitivity with greater temporal stability (around 0.55–0.60). These results confirm the advantage of managing class imbalance in dropout prediction.

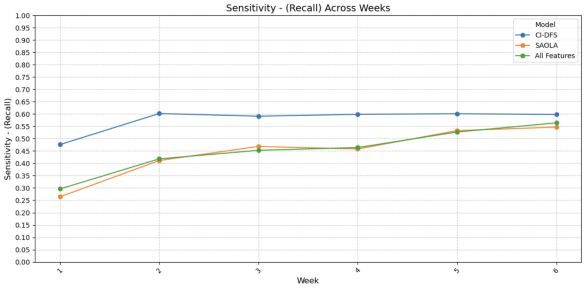

Fig. 3. Sensitivity on XuetangX

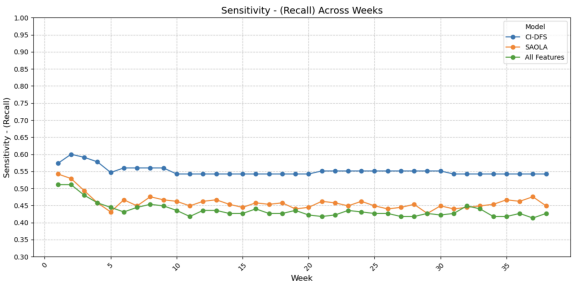

Fig. 4. Sensitivity on OULAD

In terms of the number of selected features, CI-DFS selects a significantly smaller set of features than SAOLA, while maintaining comparable accuracy and higher sensitivity. On XuetangX, CI-DFS reduces feature count by 50% to 70%; on OULAD, the reduction ranges from 73% to 85%. In contrast to SAOLA, which shows continuous feature accumulation, CI-DFS demonstrates better control, particularly on OULAD, likely due to its drift management strategy.

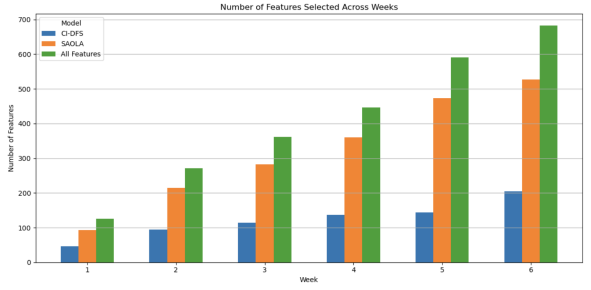

Fig. 5. Number of features on XuetangX

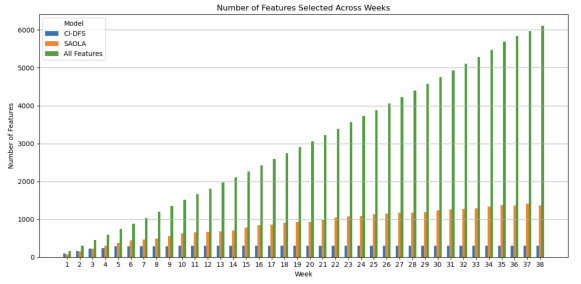

Fig. 6. Number of features on OULAD

Regarding running time, CI-DFS shows substantially better performance. On XuetangX, selection time remains constant (1–2 seconds), while SAOLA increases linearly. On OULAD, CI-DFS remains stable around 2–3 seconds, whereas SAOLA reaches up to 560 seconds by week 38. This improvement is especially relevant for large-scale and real-time applications.

In summary, although CI-DFS does not consistently surpass other models in accuracy, it maintains higher and more stable sensitivity, selects fewer features, and operates with significantly lower computational costs across both datasets. These results raise important questions regarding the specific contribution of each component within CI-DFS.

5.2. Ablation Study: Influence of the Elements of CI-DFS on Performance

In this section, we evaluate CI-DFS with an ablation study, resulting in three models: 1) Classic MI represents the model where MI is used instead of WMI, 2) Fixed Threshold manages a fixed relevance threshold θ instead of a dynamic one, 3) No Drift represents the model where no drift management is performed in the last step of CI-DFS. These three models are studied in terms of sensitivity and number of selected features (see Figures 9 to 12).

Considering sensitivity, the three models reach a significant lower sensitivity than CI-DFS, which confirms the actual contribution of each of the three elements. In particular, the dynamic threshold and drift management seem to

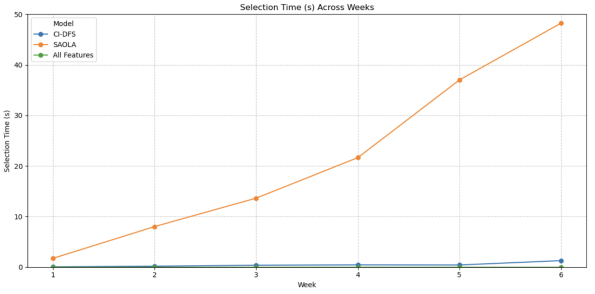

Fig. 7. Selection time on XuetangX

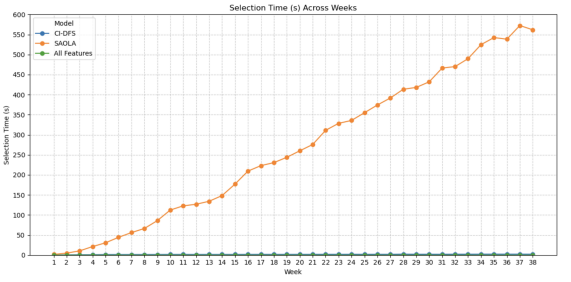

Fig. 8. Selection time on OULAD

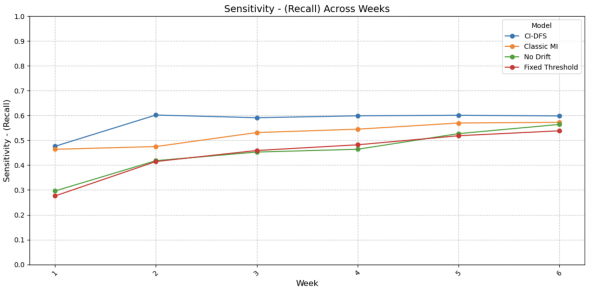

Fig. 9. Sensitivity on XuetangX

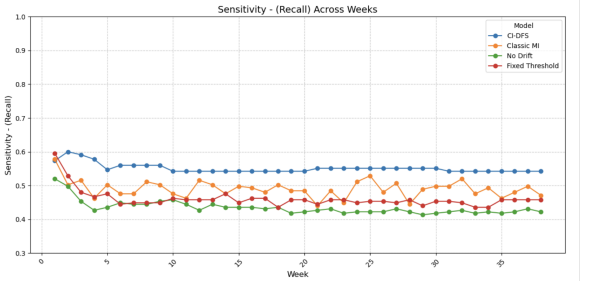

Fig. 10. Sensitivity on OULAD

Fig. 11. # Features on XuetangX

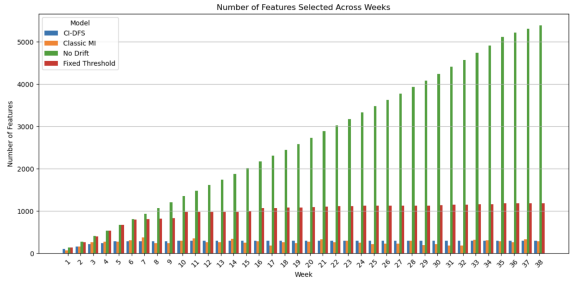

Fig. 12. # Features on OULAD

contribute the more to the sensitivity of CI-DFS, especially in the early weeks for XuetangX. An additional evaluation of the evolution of value of the θ parameter across weeks in CI-DFS confirms that the value actually significantly varies, which confirms that the characteristics of the datasets actually evolve.

Considering the number of features, we can also conclude that the three elements significantly contribute to the decrease of the number of features. Here again, the dynamic relevance threshold and drift management play crucial roles in controlling the growth of the feature set, for both datasets.

5.3. Qualitative Analysis of the Selected Features

To understand more the dynamic in the set of features selected through weeks, we show here some features that remain relevant through weeks and some whose relevance evolves. First of all, in each dataset, a feature consistently appears as a key feature across all weeks. Registration date for OULAD and In XuetangX, it is the education level. Considering feature relevance decrease, in both datasets, the early forum activity features are discarded after the first weeks, indicating their decreasing predictive value. However, some features appear later during the course, and become very relevant in the prediction. This evolution demonstrates CI-DFS's ability to adapt to changing student interaction patterns while maintaining relevant features.

6. Conclusion

In this work, we have proposed CI-DFS, a new dynamic feature selection algorithm, dedicated to imbalanced data. Experiments on two educational datasets demonstrate CI-DFS's effectiveness in handling class imbalance and managing the evolution of feature selection in learning environments. CI-DFS demonstrates several key strengths, but reveals areas for future investigation. Especially, our experiments have so far been conducted with binary classification, but we need to explore the extension to multi-class problems. Further, we intend to study the development of incremental training capabilities to update the model as new data arrives, reducing the need for complete retraining, which is currently used.

References

[1] A. M. Rabelo, L. E. Zárate, A model for predicting dropout of higher education students, Data Science and Management (2024).
[2] E.-Y. Seo, J. Yang, J.-E. Lee, G. So, Predictive modelling of student dropout risk: Practical insights from a south korean distance university, Heliyon 10 (11) (2024).
[3] M. G. Gallego, A. P. Perez de los Cobos, J. C. G. Gallego, Identifying students at risk to academic dropout in higher education, Education Sciences 11 (8) (2021) 427.
[4] J. Chen, B. Fang, H. Zhang, X. Xue, A systematic review for mooc dropout prediction from the perspective of machine learning, Interactive Learning Environments 32 (5) (2024) 1642–1655.
[5] M. Youssef, S. Mohammed, E. K. Hamada, B. F. Wafaa, A predictive approach based on efficient feature selection and learning algorithms' competition: Case of learners' dropout in moocs, Education and Information Technologies 24 (6) (2019) 3591–3618.
[6] M. Rahmaninia, P. Moradi, Osfsmi: Online stream feature selection method based on mutual information, Applied Soft Computing 68 (2018) 733–746.
[7] K. Yu, X. Wu, W. Ding, J. Pei, Scalable and accurate online feature selection for big data, ACM Transactions on Knowledge Discovery from Data (TKDD) 11 (2) (2016) 1–39.
[8] P. Zhou, X. Hu, P. Li, X. Wu, Ofs-density: A novel online streaming feature selection method, Pattern Recognition 86 (2019) 48–61.
[9] P. Zhou, S. Zhao, Y. Yan, X. Wu, Online scalable streaming feature selection via dynamic decision, ACM Transactions on Knowledge Discovery from Data (TKDD) 16 (5) (2022) 1–20.
[10] L. Qiu, Y. Liu, Y. Liu, An integrated framework with feature selection for dropout prediction in massive open online courses, IEEE Access 6 (2018) 71474–71484.
[11] N. Al Nuaimi, M. M. Masud, Online streaming feature selection with incremental feature grouping, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 10 (4) (2020) e1364.
[12] D. You, X. Wu, L. Shen, Z. Chen, C. Ma, S. Deng, Online feature selection for streaming features with high redundancy using sliding-window sampling, in: In International Conference on Big Knowledge (ICBK), IEEE, 2018, pp. 205–212.
[13] X. Wu, K. Yu, H. Wang, W. Ding, Online streaming feature selection, in: Proceedings of the 27th international conference on machine learning (ICML-10), Citeseer, 2010, pp. 1159–1166.
[14] S. G. Devi, M. Sabrigiriraj, Swarm intelligent based online feature selection (ofs) and weighted entropy frequent pattern mining (wefpm) algorithm for big data analysis, Cluster Computing 22 (Suppl 5) (2019) 11791–11803.
[15] P. Zhou, Y. Zhang, P. Li, X. Wu, General assembly framework for online streaming feature selection via rough set models, Expert Systems with Applications 204 (2022) 117520.
[16] I. C. Covert, W. Qiu, M. Lu, N. Y. Kim, N. J. White, S.-I. Lee, Learning to maximize mutual information for dynamic feature selection, in: In Proceedings of the 40th International Conference on Machine Learning, Proceedings of Machine Learning Research, 2023, pp. 6424–6447.
[17] M. Thomas, A. T. Joy, Elements of information theory, Wiley-Interscience, 2006.
[18] J. R. Vergara, P. A. Estévez, A review of feature selection methods based on mutual information, Neural computing and applications 24 (2014) 175–186.
[19] E. Schaffernicht, H.-M. Gross, Weighted mutual information for feature selection, in: 21st International Conference on Artificial Neural Networks (ICANN), Springer, 2011, pp. 181–188.
[20] K. Li, M. Yu, L. Liu, T. Li, J. Zhai, Feature selection method based on weighted mutual information for imbalanced data, International Journal of Software Engineering and Knowledge Engineering 28 (08) (2018) 1177–1194.
[21] E. A. K. Zaman, A. Mohamed, A. Ahmad, Feature selection for online streaming high-dimensional data: A state-of-the-art review, Applied Soft Computing 127 (2022) 109355.
[22] N. AlNuaimi, M. M. Masud, M. A. Serhani, N. Zaki, Streaming feature selection algorithms for big data: A survey, Applied Computing and Informatics 18 (1/2) (2022) 113–135.
[23] W. Feng, J. Tang, T. X. Liu, Understanding dropouts in moocs, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 33, 2019, pp. 517–524.
[24] J. Kuzilek, M. Hlosta, Z. Zdrahal, Open university learning analytics dataset, Scientific data 4 (1) (2017) 1–8.
[25] C. Sammut, G. I. Webb, Encyclopedia of machine learning, Springer Science & Business Media, 2011.