Amélioration d'une Méthode de Clustering des Traces Moodle via l'Encodage des SSF

Noura Joudieh [1], **Marwa Trablesi** [1], **Ronan Champagnat** [1], **Mourad Rabah** [1], **Samuel Nowakowski** [2], **Nikleia Eteokleous** [3]

1. *L3i - Université de La Rochelle*
   Avenue Michel Crépeau
   17 042 La Rochelle, France
   nom.prenom@univ-lr.fr
2. *LORIA - Université de Lorraine*
   Campus Scientifique, 615 rue du jardin-botanique
   54 506 Vandœuvre-lès-Nancy, France
   samuel.nowakowski@loria.fr
3. *Frederick University*
   Department of Education
   3080 Limassol, Cyprus
   n.eteokleous@frederick.ac.cy

RÉSUMÉ. *Les apprenants mettent en place des stratégies d'apprentissages variées, ce qui rend leur traces d'apprentissage riches et précieuses pour déterminer des recommandations de parcours d'apprentissage pour d'autres apprenants. Dans ce contexte, la fouille de processus permet de découvrir des modèles qui révèlent les parcours d'apprentissage des apprenants dans une plateforme éducative. Dans cet article, nous discutons des limitations et proposons des améliorations d'une approche de regroupement des traces nommée « FSS-encoding ». Nos améliorations visent à enrichir la définition du vecteur caractérisant les traces. Notre méthode a été appliquée aux traces Moodle collectées entre 2018 et 2022 à l'Université Frederick à Chypre.*

ABSTRACT. *Learners adopt various learning patterns and behaviors while learning, rendering their experience a valuable asset for recommending learning paths for other learners. Process Mining is useful in this case to discover models that reveal learners' taken learning paths in an educational platform. In this paper, we address the limits of and improve on a feature-based trace clustering approach known as FSS-encoding. Our enhancements include a refined pattern selection, preserving the uniqueness of less frequent events and increasing the overall effectiveness of the trace clustering process. Our method was applied to Moodle logs collected from 2018 to 2022 in the Frederick University.*

MOTS-CLÉS : *SI pédagogique, Scénarios d'apprentissage, trace clustering*

KEYWORDS: *Learning management system, Trace Clustering in process mining, Learning Paths*

1. Introduction

L'évolution des technologies a étendu les possibilités d'apprentissage au-delà des salles de classe traditionnelles et des interactions conventionnelles entre enseignants et élèves. De nos jours, l'apprentissage en ligne est de plus en plus populaire, offrant un accès à une multitude de ressources éducatives à tout moment et en tout lieu. Cependant, cette accessibilité accrue comporte son lot de défis : les apprenants peuvent se sentir surchargés lorsqu'ils cherchent à atteindre leurs objectifs d'apprentissage, ce qui peut potentiellement diminuer leur motivation et leur efficacité.

Dans cette perspective, les systèmes de recommandation (RS) (Aggarwal, 2016) pour l'apprentissage en ligne visent à personnaliser l'expérience d'apprentissage en filtrant de manière intelligente le contenu en ligne en fonction des préférences, des actions et des besoins individuels des apprenants, s'éloignant ainsi des modèles génériques. De plus, les utilisateurs des systèmes d'information laissent des traces enregistrées par le système de journalisation sous forme de journaux d'événements.

La fouille de processus (*Process Mining*), une discipline qui combine la fouille de données, l'apprentissage automatique et la modélisation des processus métier, exploite ces journaux d'événements pour découvrir les modèles de processus qui décrivent le comportement des utilisateurs au sein d'un système (Aalst, 2016). Dans les plates-formes et systèmes éducatifs, cela a ouvert la voie à de prometteurs travaux de recherches visant à identifier les comportements des étudiants lorsqu'ils s'engagent dans diverses activités d'apprentissage telles que suivre un cours ou passer une évaluation (Cenka, Anggun, 2022).

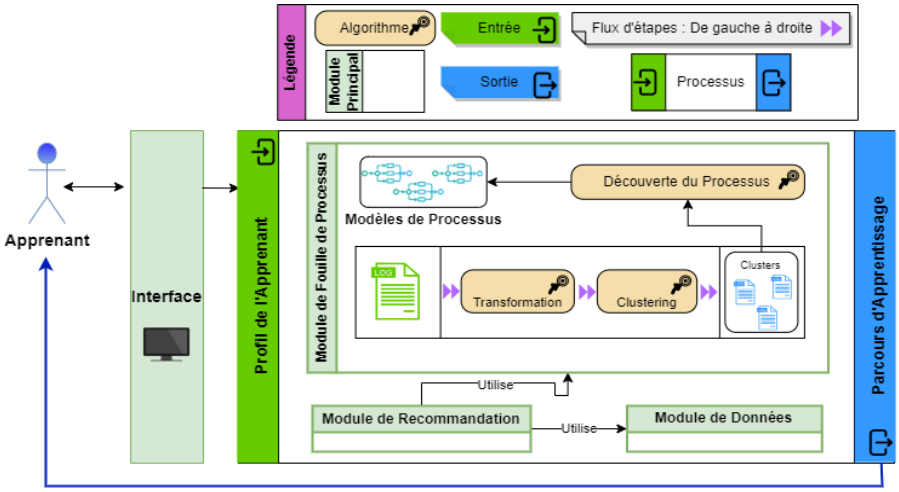


FIGURE 1 – Plate-forme pour la recommandation de parcours d'apprentissage

Dans nos travaux précédents (Joudieh *et al*., 2023) et comme présenté dans la Figure 1, nous avons proposé un framework pour recommander un parcours d'apprentissage adaptatif personnalisé pour un apprenant possédant un objectif d'apprentissage, en utilisant son parcours d'apprentissage passé extrait via la fouille de processus. Ce framework est composé de trois modules principaux : le module de recommandation, le module de données et le module de *Process Mining*. Le présent article se concentre sur le module de fouille de processus. Ce dernier est chargé d'analyser les logs du système de gestion de l'apprentissage Moodle pour découvrir un modèle montrant les parcours d'apprentissage empruntés par les étudiants. Ces modèles forment ainsi la base de recommandations de parcours d'apprentissage personnalisées et efficaces. Cependant, une grande quantité de traces peut générer des modèles incompréhensibles en forme de « spaghettis », d'où la nécessité du regroupement des traces (*trace clustering*) (Song *et al*., 2008) comme étape préliminaire pour identifier les différents types de parcours d'apprentissage et découvrir ainsi des modèles plus explicites.

Guidés par cet objectif, nous adoptons une approche récente de regroupement des traces (Trabelsi *et al*., 2021) qui repose sur l'encodage des traces en termes de sous-séquences fréquentes, nommée *FSS Encoding* (*Frequent Subsequent Patterns Encoding*). Cette méthode générique, intialement développée pour les utilisateurs des bibliothèques numériques, a réussi à identifier trois profils distincts d'utilisateurs à partir de données simulées, confirmés par des données réelles de la Bibliothèque Nationale de France. La méthode a montré que l'utilisation répétée d'une même séquence d'événements pour une tâche indique que cette séquence satisfait l'utilisateur pour atteindre son objectif. En éducation, il est intéressant de prendre en compte les patterns d'apprentissage (une sous-séquence) plutôt que les activités atomiques en analysant les traces d'apprentissage. Une simple consultation de ressources peut être informative, mais un pattern construit d'une consultation de ressources suivie d'exercices est bien plus intéressant. Pour cela, une méthode de regroupement basée sur les FSS, peut révéler les stratégies d'apprentissage et permettre de regrouper les étudiants en conséquence.

Dans le présent article, nous abordons les limites de cette méthode et proposons une extension qui améliore les résultats de regroupement et les modèles découverts. L'approche améliorée est appliquée aux traces Moodle collectées auprès de 471 étudiants suivant des cours au Département d'informatique et de génie informatique de l'Université Frederick à Chypre pour la période 2018-2022.

Dans la section 2, nous décrivons les différentes approches de regroupement des traces, suivi d'une analyse approfondie de la méthode *FSS Encoding*, de ses limitations et des améliorations proposées dans la section 3. La section 4 fournit une description détaillée de la collecte des logs générés par la plateforme Moodle et du prétraitement des données. Les résultats de l'application de la méthode FSS améliorée sur les logs Moodle sont présentés dans la section 5, ainsi que la compairison avec la méthode originale et une autre méthode identifiée dans l'état de l'art. La section 6 conclut le travail et présente les perspectives.

2. État de l'art

Le *Process Mining* est un domaine scientifique qui comble le fossé entre l'analyse orientée données et l'analyse orientée processus, visant à extraire des connaissances à partir des journaux d'événements (logs). Les techniques de fouille de processus sont appliquées dans divers domaines par les hôpitaux, les banques ou encore les collectivités territoriales (Aalst, 2016 ; Lu *et al*., 2019 ; Trabelsi *et al*., 2021).

Ces techniques utilisent les logs comme entrée pour générer, améliorer ou valider des modèles de processus (Aalst, 2016). Un journal d'événements se compose d'un ensemble de traces d'exécution, chacune représentant une instance spécifique du processus. Prenons par exemple le flux de travail d'un étudiant sur une plateforme d'apprentissage en ligne. À partir de l'action de consulter un cours (*Course viewed*), l'étudiant peut naviguer pour explorer des éléments spécifiques dans le cours (*Course module viewed*). Ces éléments peuvent inclure des cours, des vidéos, des devoirs ou des quiz. En sélectionnant un élément de quiz, l'étudiant procède ensuite à la soumission de ses réponses (*Quiz submitted*). Chacune de ces activités constitue une trace unique au sein du processus principal.

Par exemple, dans la plateforme Moodle, chaque étudiant peut être considéré comme un cas suivant un parcours d'apprentissage. La série d'événements associés à un cas spécifique est appelée une trace. Chaque ligne du tableau 1 représente un événement exécuté, comprenant des détails tels que l'identifiant de l'événement (*CaseId*), le *Timestamp* (jour, heure, minute et seconde), l'*Activité* concernée, ainsi que d'éventuels attributs supplémentaires pertinents pour l'événement selon les cas d'étude. Formellement, un journal d'événements $L = t_1, t_2, ..., t_k$ est un ensemble de k traces où chaque trace t_i ($1 \leq i \leq k$) est un ensemble de n_i événements consécutifs $t_i =< e_{i1}, e_{i2}, ...e_{in_i} >$ réalisés par le même *CaseId*.

TABLEAU 1 – Exemple de journaux d'événements.

CaseId	***Timestamp***	**Activité**
1	2018-01-12T10:34:25	*Course viewed*
2	2018-01-12T10:36:25	*Course viewed*
1	2018-01-12T10:34:26	*Course module viewed*
1	2016-01-12T10:34:28	*Submission viewed*
3	2018-01-12T10:36:26	*Course viewed*
3	2018-01-12T10:36:27	*Submission form viewed*

De nombreuses méthodes de découverte de processus ont été proposées dans la littérature dans le but de générer automatiquement des modèles de processus. Les algorithmes de découverte de processus visent à extraire des modèles de processus à partir des logs. Ces algorithmes ont pour objectif de représenter l'ensemble des activités capturées dans les logs. Divers modèles peuvent être générés à cette fin, notamment les réseaux de Petri et les Fuzzy modèles (Aalst, 2016).

Cependant, de nombreuses études en matière de fouille de processus ont démontré que la création d'un seul modèle de processus pour un ensemble de journaux entier n'est pas idéale, notamment pour les ensembles de données très volumineux contenant des processus non structurés. Un processus non structuré est généralement piloté par un utilisateur plutôt que par un logiciel. Il en résulte de nombreux chemins possibles, mais seuls quelques-uns sont pertinents. Les techniques de *process mining* conduisent souvent à des modèles complexes et/ou surajustés, tels que le modèle en « spaghettis » ou le modèle en « fleur » identifiés dans la littérature (Aalst, 2016). Pour surmonter ces problèmes, les travaux existants ont proposé des méthodes de regroupement des traces avant la modélisation (Diamantini *et al*., 2016). La littérature propose de nombreuses approches de regroupement des traces, qui peuvent être catégorisées en trois types de techniques de regroupement en fonction de la façon dont les traces sont présentées avant le regroupement (Song *et al*., 2008 ; Zandkarimi *et al*., 2020). De plus, il existe une catégorie de regroupement dite hybride, qui intègre diverses techniques issues des méthodes mentionnées précédemment (De Koninck, De Weerdt, 2019).

La première catégorie, appelée *Trace based clustering*, regroupe les traces en fonction de leur similarité syntaxique, comme expliqué dans (Bose, Aalst, 2009a) et (Chatain *et al*., 2017). Cette approche s'inspire de la distance de Levenshtein, qui mesure la dissimilarité entre deux chaînes de caractères. Dans ce contexte, une trace peut être transformée en une autre par le biais d'opérations d'édition telles que la substitution, l'ajout ou la suppression d'événements. La distance d'édition entre deux traces est ensuite calculée comme le nombre minimum d'opérations d'édition nécessaires pour convertir une trace en une autre. Une distance d'édition plus faible indique un niveau de similarité plus élevé entre les traces. Ensuite, des algorithmes de regroupement basés sur la distance sont appliqués pour regrouper les traces en clusters distincts. Dans le domaine de l'éducation, à la fois (Laksitowening *et al*., 2023) et (Zhang *et al*., 2022) se concentrent sur les logs des étudiants pour capturer différentes caractéristiques et schémas d'apprentissage. Ils utilisent tous deux le regroupement hiérarchique comme algorithme de regroupement pour regrouper les traces des étudiants.

La deuxième catégorie est le *Model-based clustering*. Elle met l'accent directement sur la qualité des modèles découverts et la distribution des traces parmi les groupes de traces. Elle suppose que les modèles de processus précis et pertinents sont découverts à partir de sous-logs homogènes (Cadez *et al*., 2003 ; Ferreira *et al*., 2007). Le modèle de processus est considéré comme une entrée pour le regroupement afin de structurer les traces. Ces traces sont ensuite utilisées pour extraire de nouveaux modèles de processus. Les groupes de traces obtenus dépendent fortement des résultats de mesures d'évaluation de la qualité des modèles découverts (De Weerdt *et al*., 2013).

La troisième catégorie, appelée *Feature-based clustering*, implique la conversion de chaque trace en un vecteur de caractéristiques basé sur des caractéristiques prédéfinies. La similarité entre deux traces est ensuite déterminée par la similarité entre leurs vecteurs respectifs. Les méthodes existantes dans cette catégorie reposent souvent sur des métriques telles que la fréquence des événements ou la fréquence des relations de

succession directe entre les événements pour transformer les traces en vecteurs (Song *et al*., 2008). Par exemple, (Song *et al*., 2008) a analysé des traces provenant de systèmes d'information de santé, les convertissant en caractéristiques telles que le nombre d'occurrences individuelles d'événements ou le nombre de paires d'événements en succession immédiate. (Bose, Aalst, 2009b) a utilisé une technique similaire sur de plus longues sous-parties de traces, évaluant l'occurrence de motifs plus complexes tels que les répétitions, définies comme des *n-grammes* observés à différents points de la trace. Par la suite, des algorithmes de regroupement basés sur la distance sont appliqués pour regrouper les traces en clusters distincts (Zandkarimi *et al*., 2020).

Dans cet article, notre travail s'inscrit dans l'approche basée sur les caractéristiques (*Feature-based clustering*), où nous améliorons une méthode appelée *FSS Encoding* (Trabelsi *et al*., 2021). Cette méthode, initialement conçue pour les bibliothèques numériques, vise à extraire des caractéristiques des séquences en identifiant les sous-séquences fréquentes et en les encodant. L'encodage des sous-séquences fréquentes prend en compte divers paramètres pour différencier efficacement les séquences des utilisateurs des bibliothèques numériques. Un algorithme de regroupement est ensuite appliqué sur les traces converties afin d'assigner chaque trace d'apprenant au *cluster* approprié. Dans le contexte des bibliothèques numériques, la méthode *FSS encoding* a montré son efficacité pour modéliser les trajectoires des utilisateurs. Ces résultats ont été validés sur des données réelles issues de la Bibliothèque Nationale de France.

3. *FSS-encoding* appliqué aux traces d'apprenants utilisant Moodle

Comme mentionné dans la section 1, nous avons amélioré la méthode *FSS encoding* (désignée dorénavant comme la méthode de référence) proposée par (Trabelsi *et al*., 2021). Cette amélioration préserve davantage d'informations sur les traces ainsi que l'unicité de chaque trace, ce qui permet de découvrir de meilleurs motifs à partir des traces.

3.1. FSS Encoding : référence

Nous allons décrire brièvement la méthode de référence et l'algorithme *FSS encoding* proposé dans (Trabelsi *et al*., 2021). La stratégie fondamentale de cette méthode repose sur l'hypothèse qu'une trace est caractérisée par sa ou ses sous-séquences fréquentes (FSS) la ou les plus significatives (Lu *et al*., 2019). Cette stratégie implique de regrouper les traces en fonction des sous-séquences fréquentes. Une FSS, désignée comme $< e_1, ..., e_n >$, comprend un ensemble fini d'événements de longueur n ($n > 1$), où les événements sont exécutés dans l'ordre au moins deux fois.

Les traces sont converties à l'aide d'un encodage spécifique. Dans cet encodage, chaque FSS identifiée dans une trace est remplacée par son encodage correspondant. Les événements qui n'appartiennent à aucune FSS sont considérés comme non pertinents, et seules leurs positions contribuent au regroupement. Par conséquent, de tels événements dans les traces sont remplacés par la valeur 1.

La méthode de référence elle-même vise à distinguer efficacement les traces au sein de différents clusters en considérant des facteurs tels que [*1- la longueur*] et [*2- la fréquence*] des FSS, [*3- la fréquence des événements*] au sein des FSS, et [*4- la fréquence des relations de succession directe entre les événements*] dans les FSS. Cette stratégie d'encodage améliore la représentation vectorielle des traces, en mettant l'accent sur l'importance des FSS plus longues, des fréquences plus élevées, ainsi que sur l'occurrence d'événements et de relations spécifiques au sein des séquences.

Toutes les FSS extraites sont encodées dans chaque trace comme suit :

$$Encoding(FSS) = \frac{1}{f_{FSS} \sum_{i=1}^{n-1} f_{e_i} f_{e_{i+1}} f_{r_{i,i+1}}} \tag{1}$$

Où, f_{FSS} est la fréquence de la FSS extraite, n est sa longueur (nombre d'événements), f_{e_i} est la fréquence de l'événement e_i dans les logs et $f_{r_{i,i+1}}$ est la fréquence de la relation directe entre tous les événements consécutifs de la FSS dans les logs. La valeur d'encodage résultante est comprise entre 0 et 1. Une valeur proche de 0 indique l'importance de la FSS dans l'ensemble des journaux d'événements.

Cette méthode d'encodage permet de distinguer les traces qui partagent la même FSS mais pas dans la même position. De plus, en remplaçant tous les événements ne faisant pas partie d'une FSS par 1, l'information sur la position de la FSS est conservée, tout comme les écarts entre différentes FSS et la taille de la trace. Après l'encodage FSS, toutes les traces dans les logs sont converties en vecteurs. Ces vecteurs sont ensuite regroupés en fonction de leur similitude.

3.2. FSS Encoding : Amélioration

L'approche de référence remplace tous les événements qui ne font pas partie d'une sous-séquence fréquente par 1. Cela entraîne une perte d'informations concernant l'unicité des activités individuelles qui ne participent pas à un motif. Ces activités peuvent avoir une importance même si elles se produisent moins fréquemment. La simplification de la méthode de référence peut négliger la diversité et l'importance de telles activités singulières, ce qui pourrait avoir un impact sur l'exhaustivité de l'analyse. Par exemple, deux traces, $t_1 = < e_1, e_2, e_3, e_4, e_5 >$ et $t_2 = < e_0, e_1, e_2, e_3 >$, seront converties en vecteurs $[E_{FSS_1}, 1, 1]$ et $[1, E_{FSS_1}]$, respectivement, où E_{FSS_1} représente l'encodage de $< e_1, e_2, e_3 >$. Les événements e_0, e_4 et e_5 ne seront pas pris en compte dans le regroupement.

D'autre part, notre amélioration prend en compte la fréquence et les relations à la fois des FSS et des événements individuels, offrant une représentation plus nuancée qui préserve les caractéristiques distinctives de chaque activité. Par exemple dans Moodle, un apprenant est plus susceptible de consulter un cours plusieurs fois avant de passer un quiz une seule fois. Ainsi, il est important de préserver la position et l'identité de ces activités moins fréquentes car elles pourraient contenir des informations significatives pour comprendre le processus d'apprentissage entrepris. Guidés par cela,

en utilisant notre approche améliorée, les traces t_1 et t_2 dans l'exemple précédent sont respectivement converties en vecteurs $[E_{FSS_1}, f(e_4), f(e_5)]$ et $[f(e_0), E_{FSS_1}]$. $f(a)$ représente la fréquence d'occurrence de l'activité a dans toutes les traces, préservant ainsi son identité en fonction de sa fréquence qui reflète finalement sa signification.

L'algorithme 1 présente la version améliorée de la méthode de référence, décrivant l'approche proposée dans le présent article. La première étape consiste à transformer les logs originaux R (voir le tableau 1) en un ensemble de traces L. Cet ensemble organise la séquence d'événements chronologiquement en se basant sur l'identifiant unique *CaseId*. Par exemple, en se basant sur le tableau 1, la trace correspondante de *CaseId* 1 est $<$ *Course viewed*, *Course module viewed*, *Submission viewed* $>$.

Data : Original log file R, Minimum pattern support percentage $minSup$, Minimum pattern Length $minLen$, Number of Clusters $n\_clusters$
Result : Log Files F corresponding to resulting clusters
begin
- Convert R to a set of traces L;
- From L, extract frequent sub-sequences FSS with length $>= minLen$ and minimum support $>= minSup$;
- From FSS, remove $x \in FSS$ if x does not exist as is in L;
- For $x \in FSS$, compute $Encoding(x)$;
- Sort FSS in descending order of pattern lengths ;
- For each trace in L, replace any existing FSS by their encoding;
- Remove traces from L where no FSS is found;
- For remaining traces in L, replace remaining activities with their frequency ;
- Scale the values in the traces using MinMax Scaler, to have a range of $[0, 1]$;
- Add padding of -1 for the traces to have same lengths ;
- Cluster the traces in L into $n\_clusters$;
- Generate Log Files F for resulting clusters;
- Return F;

end

Algorithme 1 : Algorithme d'encodage FSS amélioré

Ensuite, nous utilisons l'algorithme *PrefixSpan*pour extraire les motifs séquentiels FSS à partir des logs modifiés L. Nous choisissons cet algorithme pour sa capacité à identifier de manière efficace les motifs fréquents dans les traces, qu'il s'agisse de motifs avec des événements contigus ou non contigus, ce qui facilite la découverte de séquences d'activités significatives.

Dans la méthode de référence, les critères de sélection des sous-séquences fréquentes avec *PrefixSpan* reposent sur les k-motifs les plus fréquents extraits. Cependant, cette approche entraîne une perte de précision sur la pertinence des motifs en fonction du nombre de traces qui les contiennent. Il peut en résulter des situations où tous les motifs les plus fréquents ont des pourcentages d'apparition supérieurs à 90%, ou au contraire, certains motifs ont des pourcentages d'apparition inférieurs à 50%. En outre, l'utilisation des motifs les plus fréquents peut être coûteuse en termes de calcul, car elle implique la génération et la vérification de nombreux motifs potentiels. Notre approche améliore cette méthode en affinant les critères de sélection des motifs, en intégrant deux seuils : (i) un pourcentage d'apparition minimum ($minSup$) et (ii) une longueur de motif ($minLen$). Ainsi, lors de cet affinage, l'ensemble de données est

parcouru une seule fois pour déterminer l'apparition des motifs candidats, puis seuls ceux dépassant le seuil spécifié sont conservés. L'apparition d'un motif X est défini comme le rapport des traces dans lesquelles X apparaît par rapport au nombre total de traces, tandis que la longueur d'un motif correspond au nombre d'activités qu'il contient. De plus, étant donné que *PrefixSpan* peut extraire des motifs qui ne sont pas présents en tant que tels dans les traces, ces motifs sont filtrés lors de la sélection.

Ensuite, l'encodage de chaque FSS est calculé en utilisant l'équation 1 et les FSS sont triées par ordre décroissant de leurs longueurs (plus le motif est long, plus il est important). Cela définit la priorité de remplacement dans l'étape suivante. Pour chaque trace, où une FSS est découverte, elle est remplacée par son encodage. Si deux FSS sont trouvées dans la même trace, la FSS la plus longue est d'abord remplacée, puis l'autre FSS est cherchée dans le reste de la trace.

Après l'encodage, les traces sans FSS trouvées sont supprimées car elles sont considérées comme non représentatives. Pour les traces restantes, les activités individuelles qui ne sont pas associées à un motif sont remplacées par leur fréquence comme expliqué précédemment. Les valeurs encodées sont normalisées entre $[0, 1]$ puis l'algorithme de clustering est exécuté sur les traces restantes, converties en vecteurs numériques. Enfin, les fichiers de logs correspondant à chaque cluster (identifiés par *CaseIds*) sont générés et renvoyés en sortie. Ces fichiers sont ensuite utilisés pour découvrir un modèle de processus décrivant le comportement général des apprenants dans chaque cluster.

En résumé, notre méthode améliore l'encodage des traces de deux manières principales. Tout d'abord, elle affine les critères de sélection des motifs en remplaçant l'approche des k-motifs les plus fréquents par une combinaison du pourcentage d'apparition minimum et de la longueur des sous séquences fréquentes. Ensuite, lors de la conversion des traces, les activités individuelles au sein des traces possédant des FSS sont remplacées par leurs fréquences respectives au lieu d'une valeur uniforme de 1. Cette modification préserve l'identité et la signification positionnelle des événements moins fréquents. Ces améliorations ont un impact direct sur la représentation des traces, ce qui améliore la qualité du regroupement des traces. Par conséquent, cette amélioration facilite une compréhension et une analyse plus détaillées des scénarios d'apprentissage au sein de chaque cluster.

4. Prétraitement des données

Étant donné que les données avec lesquelles nous travaillons concernent les logs issus de la plate-forme Moodle. Cette section est ainsi dédiée à une explication détaillée des étapes de collecte et de traitement de ces données.

4.1. Collecte et prétraitement des données

Moodle est une plate-forme d'apprentissage en ligne bien connu et largement utilisé dans les universités et les établissements éducatifs. Elle contient un système de journalisation qui capture toutes les interactions des utilisateurs avec le système. Dans

le cadre de nos travaux, les journaux d'événements Moodle de 471 étudiants, inscrits à des cours du département d'informatique et de génie informatique de l'Université Frederick à Chypre de 2018 à 2022, ont été collectés. Les logs collectés ont été nettoyés pour ne conserver que les actions effectuées par les étudiants sur Moodle pendant leurs études, telles que suivre des cours, passer des tests et effectuer des devoirs. En effet, les logs initiaux contenaient les actions effectuées par tous les utilisateurs du système (étudiants, instructeurs, assistants, gestionnaires, etc.). De plus, un identifiant unique a été créé pour chaque étudiant car l'identification des étudiants change en fonction des années dans les logs initiaux.

La structure d'un fichier logs est présentée dans le tableau 2. Le *Regnum* est le numéro d'inscription, utilisé comme identifiant unique pour suivre le parcours d'un étudiant à travers différents cours et différentes années, c'est-à-dire qu'il est utilisé comme *CaseId*. Le *Timestamp* enregistre l'heure exacte de chaque événement effectué par les étudiants. Il est utilisé pour ordonner les événements. Le « Nom de l'événement » est utilisé comme Activité et le « Contexte de l'événement » donne des informations sur la ressource d'apprentissage concernée (fichier, devoir, dossier, etc.) par l'événement. Enfin, la colonne « Description » décrit l'événement de manière plus détaillée.

TABLEAU 2 – Structure d'une ligne d'un fichier log

Regnum	*Timestamp*	*Event Context*	*Event Name*	*Description*

Le nombre initial d'activités était de 65, comprenant des événements liés aux actions de cours, à la réalisation de quiz, à la soumission de devoirs, aux discussions et chats, à la consultation de profils, et autres. Seuls 14 événements sont conservés, comme indiqué dans le tableau 3. Ces événements ont été choisis car ils sont représentatifs des actions telles que l'achèvement d'un devoir ou d'une ressource d'apprentissage, l'évaluation, la réception de feedback, l'étude et l'exploration. Le journal est filtré en fonction des événements choisis pour finalement aboutir à 471 étudiants avec un total de 3942 traces pour la période de 2018 à 2022.

TABLEAU 3 – Activités retenues dans les logs Moodle.

Noms des activités	
A submission has been submitted	*Quiz attempt submitted*
Course activity completion updated	*Course module viewed*
Zip archive of folder downloaded	*Content page viewed*
Clicked join meeting button	*Course summary viewed*
Course module instance list viewed	*Sessions viewed*
Lesson started	*Lesson resumed*
Feedback viewed	*Course viewed*

4.2. Enrichissement sémantique des données

Étant donné que les logs collectés observent l'interaction entre les étudiants et la plate-forme Moodle dans des différents cours, une étape d'enrichissement sémantique a été réalisée pour définir de nouvelles activités (au sens actions observées par la fouille de processus) qui donne plus d'informations sur le sens de l'action pédagogique effectuée. Nous appelons cette activité « activité sémantique ». La création de ces activités est basée sur des règles en prenant en compte le « contexte de l'événement », le « nom de l'événement » et la « description » des logs d'origine. Les détails de cette étape de transformation sont en dehors du cadre de cet article. Les activités peuvent avoir l'une des 12 valeurs présentées dans le tableau 4. Nous différencions deux types d'activités : passive et active (indiquées respectivement par _P et _A). Lorsqu'un étudiant télécharge du matériel de cours pour l'étudier, cette action est qualifiée de « passive » car elle ne garantit pas nécessairement que l'étudiant va ensuite consulter la ressource téléchargée. En revanche, lorsque qu'un étudiant soumet un devoir, nous présumons qu'il a terminé les exercices, ce qui constitue une action « active ».

TABLEAU 4 – Les activités sémantiques générées

Les activités sémantiques			
Study_P	*Study_A*	*Revise*	*Expand*
Exercise_P	*Exercise_A*	*View*	*Interact*
Assess_P	*Assess_A*	*Feedback*	*Apply*

4.3. Préparation des Traces

Cette section décrit l'étape initiale de l'algorithme 1. Pour effectuer le regroupement, les traces doivent être extraites des logs. Comme chaque log correspond aux activités ou événements réalisées dans un cours, nous définissons une trace comme une séquence ordonnée d'événements réalisés par un étudiant dans un cours. Ainsi, le fichier de traces utilisé pour effectuer le regroupement est structuré comme illustré dans le tableau 5, où *CaseId* est la valeur unique d'un étudiant se comportant dans un cours et la trace t_i ($1 \leq i \leq k$) est une séquence ordonnée de n_i événements (transformé en « activité sémantique ») $t_i =< se_{i1}, se_{i2}, ..., se_{in_i} >$ réalisés par le même *CaseId*.

TABLEAU 5 – La structure d'un fichier de trace.

CaseId	*Trace*
1_course1	<*View*, *Exercise_P*, *Assess_A* >
1_course2	<*View*, *View*, *Study_P*, *Study_A* >
2_course1	<*Interact* >

5. Implémentation et résultats

Dans ce qui suit, nous évaluons les améliorations de notre approche au niveau de l'extraction et l'enodage des motifs ainsi qu'au niveau des clusters et des modèles découverts.

5.1. Extraction et encodage des sous séquences fréquentes

Le tableau 6 compare la méthode de référence et notre approche *FSS encoding* améliorée au niveau de l'extraction des motifs et de l'encodage des traces. Nos critères de sélection affinés offrent un meilleur contrôle sur l'apparition minimum et la longueur des motifs, ce qui conduit à plus de motifs et à des motifs plus longs avec une apparition élevée. Les traces encodées dans notre approche sont plus courtes, mettant en avant des motifs plus longs et plus significatifs, surpassant la méthode de base à la fois dans l'extraction et l'encodage des motifs, comme le reflètent les résultats ultérieurs du regroupement des traces.

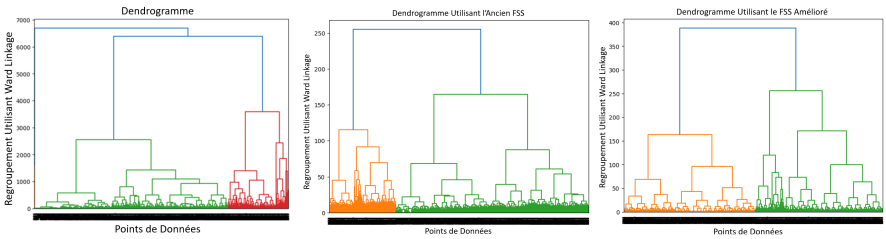
TABLEAU 6 – Comparaison des motifs extraits.

	Référence	**FSS amélioré**
Paramètres	K = 100	(MinSup = 80%, MinLen = 2)
Nb de motifs extraits	100	1412
Nb de motifs existants dans les traces	32	248
Parmi les motifs qui existent tels quels dans les traces		
[Min - Max] Longueur du motif	[1 - 6]	[2 - 9]
[Min - Max] Apparition du motif %	[86% - 100%]	[80% - 95%]
Niveau trace		
[Min - Max] Longueur originale de la trace	[2 - 5599]	[2 - 5599]
[Min - Max] Longueur des traces encodées	[1 - 2484]	[1 - 1618]
Nb de traces sans FSS	49	45

5.2. Regroupement des Traces

Dans la dernière étape de l'algorithme 1, les traces encodées à l'aide la méthode *FSS-encoding* sont regroupées à l'aide de l'algorithme de clustering *Hierarchical Agglomerative Clustering* (HAC) avec l'option « *ward linkage* », connue pour fusionner des clusters similaires selon une approche ascendante. Nous démontrons l'efficacité de notre méthode grâce à une comparaison avec la méthode de référence et une autre basée sur la fréquence d'activité (Song *et al.*, 2008). Cette dernière transforme les traces en vecteurs binaires. La longueur du vecteur est égale au nombre d'activités uniques, fournissant une représentation binaire de la présence de l'activité dans chaque trace. En ce qui concerne le regroupement, le nombre optimal de clusters, déterminé par l'analyse du dendrogramme, est de 3 pour toutes les approches, comme le montre la figure 2. Les métriques d'évaluation, y compris le coefficient de silhouette (-1 à 1, le

plus élevé étant le meilleur) et l'indice de Davies Bouldin (plus bas étant le meilleur), révèlent la qualité des clusters résultants, comme présenté dans le tableau 7.



(a) Fréquence des activités (b) FSS référence (c) FSS amélioré

FIGURE 2 – Les dendrogrammes générés par la méthode HAC en utilisant le *ward linkage*

TABLEAU 7 – La qualité du clustering de chaque méthode et les mesures de Silhouette pour les différents clusters

		Fréquence des activités	FSS-référence	FSS-amélioré
Nb Finale de Traces		3942	3893	3897
Silhouette Coefficient		0.546	0.117	0.360
Davies Bouldin Index		0.720	2.43	1.15
Détail de l'analyse Silhouette				
Cluster 0	Nb de Traces	710	1000	1496
	Silhouette	0.156	-0.065	0.260
Cluster 1	Nb de Traces	3	1482	1961
	Silhouette	0.520	0.038	0.470
Cluster 2	Nb de Traces	3229	1411	440
	Silhouette	0.632	0.408	0.250

Bien que l'on puisse initialement penser que le coefficient de silhouette est meilleur sans l'encodage des FSS, une analyse approfondie du tableau 7 révèle des interprétations potentiellement erronées des valeurs numériques. La méthode basée sur la fréquence des activités regroupe presque tous les éléments dans un seul cluster de traces, rendant ses résultats peu pertinents. En revanche, l'approche de référence divise les traces en différents clusters, mais ceux-ci manquent de séparation claire, comme en témoigne leur valeur de silhouette tandis que notre méthode combine des clusters bien séparés avec des coefficients de silhouette acceptables pour chaque cluster.

5.3. Découverte des modèles de comportements des apprenants

Nous utilisons l'algorithme *Fuzzy Miner*, implémenté dans l'outil ProM (Aalst, 2016) pour découvrir des modèles de processus de chaque cluster. Nous avons choisi l'algorithme de découverte *Fuzzy Miner* pour sa capacité à générer des modèles simplifiés mettant l'accent sur les nœuds significatifs et les arcs bien corrélées (Aalst,

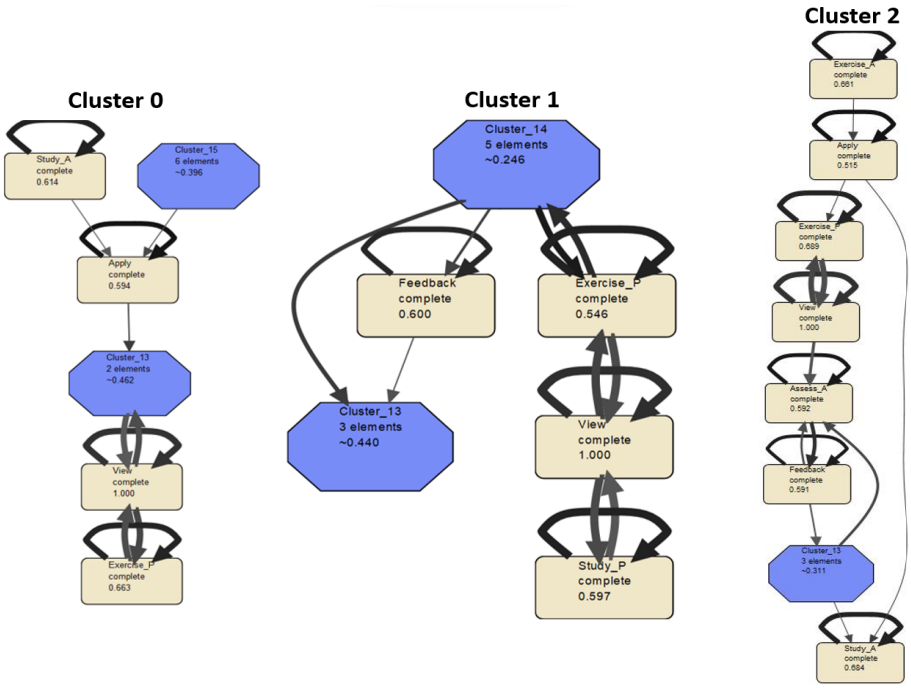


FIGURE 3 – Les modèles *Fuzzy* découverts à partir des clusters.

2016). Avec *Fuzzy Miner*, les nœuds les moins significatifs (moins fréquentes) mais fortement corrélés sont agrégés, c'est-à-dire cachés dans des clusters ayant la couleur violet au sein du modèle simplifié. La figure 3 affiche les modèles des clusters 0, 1 et 2, simplifiés en utilisant une métrique signification des nœuds élevée, ce qui conduit à l'agrégation de certains nœuds.

L'interprétation qualitative des modèles nous montre que les apprenants du cluster 1, représentant la majorité des apprenants, s'engagent principalement dans des activités routinières telles que la consultation, l'étude et la résolution d'exercices. En revanche, les étudiants du cluster 0, le deuxième plus grand cluster, regroupe des apprenants qui non seulement effectuent ces tâches routinières, mais présentent également un profil distinct en appliquant activement leurs connaissances, souvent à travers des soumissions de projets. Enfin, le cluster 2, avec le plus petit nombre d'apprenants, est composé d'individus qui préfèrent les quiz et les tests dans le cadre de leurs parcours d'apprentissage.

6. Conclusion et perspectives

Cette étude s'inscrit dans le prolongement d'un travail précédent présentant une plateforme visant à fournir des parcours d'apprentissage adaptatifs personnalisés, prenant en compte l'objectif de l'apprenant et exploitant l'expérience d'apprentissage

des apprenants précédents. En utilisant la fouille de processus, nous extrayons les parcours d'apprentissage passées grâce à la découverte de scénarios d'apprentissage. Cependant, traiter les données Moodle non structurées et volumineuses, qui possèdent des caractéristiques d'apprentissage spécifiques, constitue un défi, rendant le regroupement des traces crucial. Ainsi, notre approche améliore une méthode de regroupement des traces basée sur les sous-séquences fréquentes (FSS) en raffinant la sélection des motifs, en préservant notamment l'unicité des événements moins fréquents. Appliquée aux journaux Moodle, notre méthode montre des améliorations significatives, générant plus de motifs et des motifs plus longs, influençant les résultats d'encodage et conduisant à de meilleurs clusters comme le traduit le coefficient de silhouette.

Les clusters identifiés révèlent trois scénarios d'apprentissage distincts : l'un caractérisé par une concentration sur l'étude et la résolution d'exercices, un autre par l'application des connaissances acquises à travers des projets, et un troisième par une préférence pour réaliser plus d'évaluations. Ces scénarios fournissent des informations précieuses pour produire des recommandations personnalisées. Une évaluation par des experts des clusters identifiés, accompagnée d'une analyse approfondie des scénarios d'apprentissage, complétera bien notre démarche. Les perspectives de ces travaux visent à intégrer ces résultats dans le cadre de recommandation, en exploitant les expériences d'apprentissage passées pour un guidage plus efficace. Il convient de noter qu'une série de tests approfondis d'algorithmes de regroupement et de critères pour le clustering hiérarchique ont précédé la sélection de l'approche la plus performante présentée dans ce travail.

Bibliographie

Aalst W. Van der. (2016). *Process mining: data science in action*. Berlin, Heidelberg, Springer.

Aggarwal C. C. (2016). *Recommender Systems*. Cham, Springer International Publishing. Consulté sur http://link.springer.com/10.1007/978-3-319-29659-3

Bose R. J. C., Aalst W. M. Van der. (2009a). Context aware trace clustering: Towards improving process mining results. In *Proceedings of the international conference on data mining*, p. 401–412.

Bose R. J. C., Aalst W. M. van der. (2009b). Trace clustering based on conserved patterns: Towards achieving better process models. In *International conference on business process management*, p. 170–181.

Cadez I., Heckerman D., Meek C., Smyth P., White S. (2003). Model-based clustering and visualization of navigation patterns on a web site. *Data mining and knowledge discovery*, vol. 7, n° 4, p. 399–424.

Cenka N., Anggun B. (2022, mars). Analysing student behaviour in a learning management system using a process mining approach. *Knowledge Management & E-Learning: An International Journal*, vol. 14, n° 1, p. 62–80.

Chatain T., Carmona J., Van Dongen B. (2017). Alignment-based trace clustering. In *International conference on conceptual modeling*, p. 295–308.

De Koninck P., De Weerdt J. (2019). Scalable mixed-paradigm trace clustering using super-instances. In *2019 international conference on process mining*, p. 17–24.

De Weerdt J., Vanden Broucke S., Vanthienen J., Baesens B. (2013). Active trace clustering for improved process discovery. *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, n° 12, p. 2708–2720.

Diamantini C., Genga L., Potena D. (2016). Behavioral process mining for unstructured processes. *Journal of Intelligent Information Systems*, vol. 47, n° 1, p. 5–32.

Ferreira D., Zacarias M., Malheiros M., Ferreira P. (2007). Approaching process mining with sequence clustering: Experiments and findings. In *International conference on business process management*, p. 360–374.

Joudieh N., Eteokleous N., Champagnat R., Rabah M., Nowakowski S. (2023). Employing a process mining approach to recommend personalized adaptive learning paths in blended-learning environments. In *12th international conference in open and distance learning, athens, greece.*

Laksitowening K. A., Prasetya M. D., Suwawi D. D. J., Herdiani A. *et al.* (2023). Capturing students' dynamic learning pattern based on activity logs using hierarchical clustering. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, n° 1, p. 34–40.

Lu X., Tabatabaei S. A., Hoogendoorn M., Reijers H. A. (2019). Trace clustering on very large event data in healthcare using frequent sequence patterns. In *International conference on business process management*, p. 198–215.

Song M., Günther C. W., Aalst W. M. Van der. (2008). Trace clustering in process mining. In *International conference on business process management*, p. 109–120.

Trabelsi M., Suire C., Morcos J., Champagnat R. (2021). A new methodology to bring out typical users interactions in digital libraries. In *2021 acm/ieee joint conference on digital libraries (jcdl)*, p. 11–20.

Zandkarimi F., Rehse J.-R., Soudmand P., Hoehle H. (2020). A generic framework for trace clustering in process mining. In *2020 2nd international conference on process mining*, p. 177–184.

Zhang T., Taub M., Chen Z. (2022). A multi-level trace clustering analysis scheme for measuring students' self-regulated learning behavior in a mastery-based online learning environment. In *Lak22: 12th international learning analytics and knowledge conference*, p. 197–207.