Proactive Student Persistence Prediction in MOOCs via Multi-domain Adversarial Learning

Sreyasee Das Bhattacharjee(✉) and Junsong Yuan(✉)

State University of New York at Buffalo, Buffalo, USA
{sreyasee,jsyuan}@buffalo.edu

Abstract. Automatic evaluation of a student's STEM learning profile to understand her persistence is of national interest. In this paper, we propose an early "dropout" and behavior prediction model that can identify the potentially 'marginalized' student learning patterns to facilitate early instructional intervention in Massive Open Online Courses (MOOC) learning platform. Note that in the MOOC setting, building a comprehensive learning profile of the students is particularly more challenging due to the lack of available information and constrained communication modes. Unlike most existing works, which ignore these environmental constraints of missing information to formulate an over-simplified problem of 'one-time' prediction task in a supervised setting, the proposed model introduces a continual automated monitoring and proactive estimation process, which transforms its decision making capacity over time with evolving data patterns. In a semi-supervised scenario, the Multi-Domain Adversarial Feature Representation (mDAFR) strategy promotes the emergence of features, which are discriminative for the main learning task, while remaining largely invariant to the data sources (course from which the data was captured) in consideration. This ensures an enhanced distributed learning capacity over different course environments. Compared to transfer learning, mDAFR reports 11–15% improved classification accuracy in KDDCup dataset, and demonstrates a competitive performance against several state-of-the-art methods in both KDDCup and MOOCDropout datasets.

Keywords: Multi-feature learning · Adversarial learning · Domain adaptation · Classification · MOOC

1 Introduction

As we march into this new era of Fourth Industrial Revolutions as World Economic Forum calls it [29], it reflects on how education is evolving at a faster pace than ever before, to suit the increasing demand for the right skills. Massive Open Online Courses (MOOCs), such as Coursera, Edx are turning increasingly popular for their online course offerings. However, despite ensuring more

C. Wallraven et al. (Eds.): ACPR 2021, LNCS 13188, pp. 569–583, 2022.
https://doi.org/10.1007/978-3-031-02375-0_42

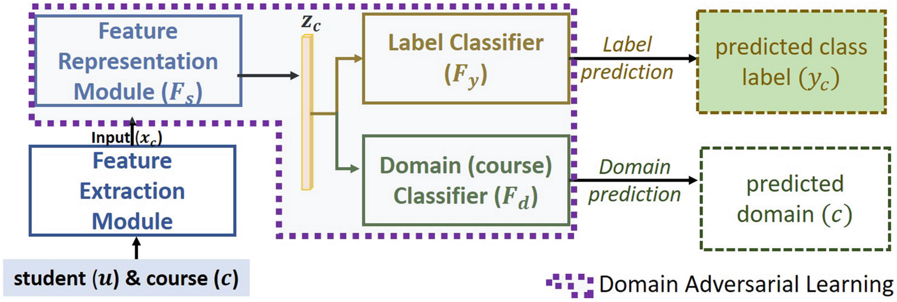
Feature Representation Module (F_s)
z_c
Label Classifier (F_y)
Label prediction
predicted class label (y_c)
Input (x_c)
Feature Extraction Module
Domain (course) Classifier (F_d)
Domain prediction
predicted domain (c)
student (u) & course (c)
Domain Adversarial Learning

Fig. 1. Workflow Diagram for the proposed method that takes into consideration of both students' course related activity details & learning behavior patterns to design a multi-domain adversarial learning, which is discriminative of the underlying category information y_c for an input sample $\mathbf{x}_c$, however invariant to the underlying course (or domain) specification (c), from which the sample was originated. During test time, F_y is used to make the persistence behavior prediction for the student u.

flexible, personalized, and collaborative learning environment compared to traditional classroom-based course offerings, attrition remains to be a challenge for the MOOC courses [13]. Recent surge of success in artificial intelligence (AI) that aims to automate several complex tasks in manufacturing, transportation, e-commerce, health care, and financial markets, triggers a fundamental research question on its applicability to support an evolving education system. Although efficient data processing tools and sophisticated multimodal data analytic algorithms have been instrumental to demonstrate impressive performances in the domains of anomaly detection, signal processing, and multimodal data analytic research [1], it is still not evident, how to utilize the power of AI most effectively to assist each actor in the STEM life cycle (student, instructors, councilors, college professionals) to augment their respective capacities toward mitigating the attrition in the educational environment.

In fact, depending on the course requirement details and the student's individual learning style, learning patterns may slightly vary across courses (like comparing activities in two courses 'Introduction to Physics' Vs 'Introduction to CS1') [10]. However, the course completion objectives may remain same for all the courses the student is presently enrolled in. Therefore, to ensure generalized performance across multiple course environments, we introduce a multi-domain adversarial feature representation learning module that cannot discriminate across the single training (*Source*) and multiple testing (*Target*) domains, and yet makes an accurate early prediction on students' persistence behavior. While most existing methods addressing this problem formulate it as a one-time prediction task and perform some post-hoc analysis [15], in a practical scenario it is important to note that the behavioral processes like self-determination and self-efficacy may evolve over a relatively short time-period. Often the change is triggered by certain surrounding environmental conditions (like the subtle presence of microaggression in a TA's response), which may frequently create some differentiated impacts on a young mind. The identification of a potentially 'marginalized' student profile is useful only if such a prediction is accurate and early enough. This would help to design appropriate intervention by the course instructors or other concerned authorities to reduce the overall attrition rate. While prediction at a fixed time-stamp may not be of much help, the proposed method employs a continual automated monitoring process, which learns the sequential activity patterns over time to ensure an early and timely risk identification more accurately. Figure 1 gives a workflow overview of the proposed Multi-Domain Adversarial Feature Representation (mDAFR) learning method. The primary contributions of the work may be summarized as:

1. ***Sequential Learning Activity Analysis for Early Detection*** that may proactively identify the 'marginalized' profiles at every pre-defined interval to facilitate a timely instructional intervention for personalized assistance.
2. ***Understanding Learning Behavior within Student Contexts*** is facilitated by clustering them into groups using the explainable k-means algorithm ExKMC [9], which not only reflects different types of learning patterns observed in the student population, but also helps understand the student-specific unique activity details, which may have impacted the clustering configurations. ExKMC enhances interpretability of the model's prediction by visualizing each cluster configuration using a small decision tree, wherein the cluster assignment of each sample is interpreted by a short sequence of single-feature thresholds.
3. ***Multi-Domain Adversarial Feature Representation Learning*** that in a semi-supervised setting, utilizes annotated samples from a *Source* course and promotes the emergence of features, which are discriminative for the main learning task and invariant to the *domain shifts* over multiple smaller *Target* courses.
4. ***Evaluating Generalization Performance across Diverse Course Environments*** using two large scale MOOC datasets in Instructor-led course settings.

The rest of the paper is organized as follows: Sect. 2 briefly describes some related works; The proposed method is described in Sect. 3; Sect. 4 and Sect. 5 respectively presents the experimental results and the follow-up discussions; and the conclusion is in Sect. 5.

2 Related Works

Although MOOC has shown tremendous potential for ensuring an enhanced accessibility to distance and lifelong learners, from the research perspective, digital activity details of participants offer a tremendous amount of data to describe students' individual learning patterns including watching video lectures, participation in discussion forums, timely submission of assignments, etc. Evidences show that these may be investigated to predict student completion [13] or engagement [4]. In this section, we briefly describe the related methods to predict dropout using students' learning activities and patterns. Since the proposed method develops a multi-domain adversarial feature representation learning model, we will also discuss related works based on adversarial learning.

Modelling MOOC Dropouts based on Learning Activity Details: A significant amount of research have explored the problem in K-12 settings [6, 28]. With the recent development in educational technologies and resources, MOOC is rapidly becoming more popular to the global learner community as a steady alternative that offers a more flexible as well as a personalized learning environment. However, the success rate of MOOC learners is often lower than that achieved by students in a physical classroom setting [16]. In fact, high dropout rate in MOOC appears to be a prominent issue, requiring immediate attention [27]. A set of recent works use deep neural network models to address the dropout prediction task in MOOC environment for predicting whether a user is likely to dropout in the next weeks [18, 26]. To enable a more accurate time-stamped analysis, some works [7, 24] model the sequential feature information to build variants of Recurrent Neural Network (RNN) models. Jeon et al. [15] present a multi-layer representation module based on Branch and Bound (BB) algorithm from the raw clickstream data. However, to ensure interpretability, useful sequence information is lost. A comprehensive literature review covering the recent progress in addressing the task of MOOC-based dropout prediction problem can be found in [2].

Adversarial Domain Invariant Learning: The proposed work is also related to Generative Adversarial Network (GAN) [12]. Existing methods develop generative models for domain adaptation or domain generalization. Both these models propose to learn an effective classifier useful for the target domain by leveraging a large collection of source domain labeled data. However, several domain adaptation techniques [3, 25] utilize its limited access to the labeled data and unlabeled data generated from the target domain to learn the target data pattern. While the evolving data characteristic and the availability of large annotated sample collection in *Target* domain pose additional challenges for our problem setting, the category sets in both *Source* and *Target* domains are identical in our scenario. Given this, we develop a novel variant of multi-domain adversarial feature representation learning model that promotes emergence of a learned descriptor, while demonstrating significant invariance to the underlying course context. This is critical as often a student's individual learning behaviour and sense of persistence have a uniformly dominant influence in all the courses, the student has recently been enrolled, wherein other course-specific details may not be equally discriminative.

3 Proposed Method

In particular we have $\mathbf{x}_c \overset{def}{=} \{\mathbf{x}_c^{(t)}\}_{t=1}^{T} \in \mathbb{R}^{m \times T}$, where the learning activity at a given time-stamp $t \in \{1, ..., T\}$ and $c \in \mathcal{C}$, is represented in terms of a compact m dimensional descriptor $\mathbf{x}_c^{(t)}$ that may capture student data from two different

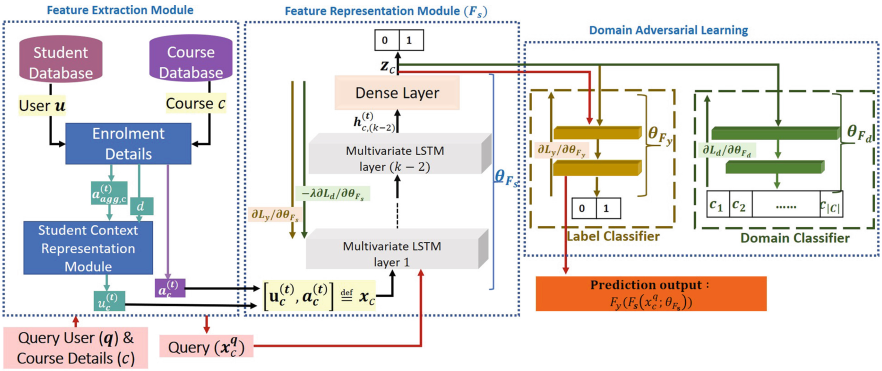
Feature Extraction Module
Student Database
Course Database
User u
Course c
Enrolment Details
$a_{agg,c}^{(t)}$
d
Student Context Representation Module
$a_c^{(t)}$
$u_c^{(t)}$
Query User (q) & Course Details (c)
Query (x_c^q)
Feature Representation Module (F_s)
0
1
z_c
Dense Layer
$h_{c,(k-2)}^{(t)}$
Multivariate LSTM layer $(k-2)$
$-\lambda\partial L_d/\partial\theta_{F_s}$
$\partial L_y/\partial\theta_{F_s}$
Multivariate LSTM layer 1
$\left[\mathbf{u}_c^{(t)}, a_c^{(t)}\right] \overset{\text{def}}{=} x_c$
θ_{F_s}
Domain Adversarial Learning
$\partial L_y/\partial\theta_{F_y}$
θ_{F_y}
0
1
Label Classifier
$\partial L_d/\partial\theta_{F_d}$
θ_{F_d}
c_1
c_2
......
$c_{|C|}$
Domain Classifier
Prediction output :
$F_y(F_s(x_c^q;\theta_{F_s}))$

Fig. 2. For each $c \in \mathcal{C}$ and student $u \in \mathcal{U}$, a sample $\mathbf{x}_c \in \mathcal{D}_c \subset \mathbb{R}^{m \times T}$ represents the learning activity in terms of a T-length sequence. At a given time-stamp $t \in \{1, ..., T\}$ for course $c \in \mathcal{C}$, it captures both *Course Activity* ($\mathbf{a}_c^{(t)}$), and *Student Context* ($\mathbf{u}_c^{(t)}$) as its components. In our experiments, we have a new time-stamp on every *third* day in the entire span of the course duration, wherein the last day of the course is denoted by T. The figure illustrates an overview of the proposed Multi-Domain Adversarial Feature Representation (mDAFR) learning method to ensure a domain invariant descriptor $\mathbf{z}_c$ with minimized *domain shift* across several course settings in $\mathcal{C}$. During testing phase, $F_y(F_s(\mathbf{x}_c^q;\theta_{F_s});\theta_{F_y})$ is used to predict the labels for the query $\mathbf{x}_c^q$, for any $c \in \mathcal{C}$.

perspectives: (1) An overall *Student Context* that reports a holistic understanding of student's overall working style in the course and its comparative pattern against the overall course population at the time-stamp t. This component is denoted by $\mathbf{u}_c^{(t)}$ and the details of its derivation are described in Sect. 3.1; and (2) *Course Activity Descriptor* that reports the student's course-specific learning activity details (like 'access', 'navigate', 'video views' etc.) for $c \in \mathcal{C}$ at the time instant t and denoted by $\mathbf{a}_c^{(t)}$. The details of the specific types of features that we use to build $\mathbf{a}_c^{(t)}$ is discussed in Sect. 4.2. We concatenate these two components to describe a student's comprehensive learning pattern $\mathbf{x}_c^{(t)}$ at a given time instant t. The label $y_c = 0$ (or $y_c = 1$) represents the fact that the student described by the multivariate sequence vector $\mathbf{x}_c$ successfully completed all course requirements (or dropped out) of the course $c \in \mathcal{C}$ within the course lifespan $[1, T]$. In addition to making a summative evaluation of an input query $\mathbf{x}_c$, by designing an effective multi-domain adversarial sequence modelling scheme, the proposed method also enables proactive evaluation, wherein for a predefined $\eta > 0$ and a subsequence $\mathbf{x}_c^{t_0} \overset{def}{=} [\mathbf{x}_c^{(t_0-\eta)}, ..., \mathbf{x}_c^{(t_0)}]$ of $\mathbf{x}_c$ describing the student's course-specific learning history at time t_0 ($t_0 > \eta$), the system can also make an early prediction on their 'marginalization' score highlighting their dropout risk. The entire dataset $\mathcal{D} = \cup_{c \in \mathcal{C}} \mathcal{D}_c$ is comprised of multiple course-specific collections of students' activity details.

Note that for each course-specific collection $\mathcal{D}_c$, the sample population representing each of the two classes, may be highly unbalanced, like in a practical scenario, students dropping out of a course in the middle, would be comparably a rare phenomenon. Therefore, a dataset with nearly uniform distribution for samples representing both the categories (‘successful completion’ and ‘dropping out’) may not be always possible. To address such data scarcity, Transfer Learning (TL) (or more specifically termed as semi-supervised Domain Adaptation) is often found as a solution, where a ML model learned using the data collection from a *Source* domain (e.g. a course with plenty of labeled data), is transferred to a *Target* domain (e.g. a course for which the dataset is more unbalanced, or the dataset size is not reasonably large enough to build a sophisticated ML/DL model from scratch) to be finetuned in the context of the *Target* domain [20]. However, this risks the transferred neural network model prone to catastrophic forgetting [17,30].

While some existing works propose to directly combine the data gathered from multiple courses to compensate for such data imbalance, subtle yet critical course-specific fine-grained context details preserved within a learned feature descriptor may not be reasonable generalizable and thus may negatively impact on the prediction performance. Therefore, our model relies on the theory of domain adaptation [10], which suggests that the predictions must be made on the feature descriptors that cannot discriminate among multiple domains. To this effect, given a collection of annotated samples $\mathcal{D}$, we design an effective and efficient domain adversarial feature representation learning model that promotes emergence of a learned descriptor, which is discriminative to the main task (i.e. identifying profiles potentially at the risk for dropping out) and also optimized to demonstrate invariance to the underlying data distribution variance observed across *Source* and multiple potential *Target* domains. An overview of the proposed algorithm is illustrated in Fig. 2.

3.1 Feature Extraction

The entire set of students in our data collection is represented as $\mathcal{U}$. In order to gain a better understanding of a student’s overall learning behavior at a given time-stamp t compared to the whole class population, the *Course Activity Descriptor* reports the student’s course-specific learning activity details (like ‘access’, ‘navigate’, ‘video views’ etc.) for $c \in \mathcal{C}$ at the time instant t. For each $u \in \mathcal{U}$, the *Course Activity Descriptor* is denoted by $\mathbf{a}_c^{(t)}$, the details of which is discussed in Sect. 4.2. Given $\mathbf{a}_c^{(t)}$, the overall *Student Context* descriptor $\mathbf{u}_c^{(t)}$, is designed to capture a holistic understanding of student’s overall working style in the course c and its comparative pattern against the overall course population at the time-stamp t.

Toward facilitating the derivation of a comprehensive student-specific learning pattern, a clustering analysis is performed to capture an aggregated understanding of the student’s learning pattern from all the courses the student is currently enrolled in. For a given $u \in \mathcal{U}$, an aggregated learning activity in course c is represented as: $\mathbf{a}_{agg,c}^{(t)} = \sum_{n=1}^{t} \mathbf{a}_c^{(n)}$. We use the explainable k-means algorithm ExKMC [9] that takes inputs k as an estimated cluster number and the entire data collection $\mathcal{U}$ represented using the set $\{\mathbf{a}_{agg,c}^{(t)}\}$, to partition into

k clusters $\{\mathcal{U}_k\}_{k=1}^{5}$. Following Silhouette Analysis [23], the number of clusters is set to be $k = 5$. The primary objective of ExKMC is to generate an explainable k-means clustering using a threshold tree with a specified number of leaves. In fact, the clustering algorithm is initiated by building a threshold tree with k leaves using the Iterative Mistake Minimization (IMM) algorithm [5]. IMM first runs a standard k-means algorithm, producing a set of k centers that are given as an additional input. Then, given a budget of $k^{'}$ leaves, it greedily expands the tree to reduce the clustering cost. At each step, the clusters form a refinement of the previous clustering by adding more thresholds to allow for more flexibility in the data partition and employ a surrogate cost to enable multiple leaves to correspond to the same cluster. The main idea is that by fixing the centers between steps, we can more efficiently determine the next feature-threshold pair to add. The surrogate cost is non-increasing throughout the execution as the number of leaves $k^{'}$ grows. When $k^{'} = n$, then the k-means cost matches that of the reference clustering. In our experiments, we have used $k^{'} = 2k$. In fact, extending the tree to use $k^{'}$ leaves with ExKMC leads to a lower-cost result that better approximates the reference clustering and helps find an explainable clustering with high accuracy, while using only $O(k)$ leaves for k-means clustering. Note that in this specific application scenario, just tagging a student as 'marginalized' based on their cluster assignment may be risky and may also prove biased. Thus, an additional interpretation supporting the system prediction on a student's cluster assignment may be significant.

Student Context Descriptor: The overall *Student Context* vector is defined as $\mathbf{u}_c^{(t)} \overset{def}{=} \left[u_{c,1}^{(t)}, ..., u_{c,5}^{(t)}, d\right] \in \mathbb{R}^6$, where $u_{c,k}^{(t)}$ represents the probability that the student u belongs to cluster $\mathcal{U}_k$. The term $d \overset{def}{=} 1 - \frac{E_u}{E_f + E_u}$ computes the student's overall *Persistence Score*, by computing the ratio between the number of dropped out courses E_u and the total number of courses that the student has enrolled, including the ones that the student has completed. The term E_f represents the number of courses the student has completed by now. Note that the *Persistence Score* is 1 until a student drops out of a course. When demographic data (e.g. age group, gender, education level) $\mathbf{g}_u$ is available for each student $u \in \mathcal{U}$, we further cluster samples (represented using an augmented vector $[\mathbf{a}_{agg,c}^{(t)}, \mathbf{g}_u]$) within each $\mathcal{U}_k$ into 5 different groups $\{\mathcal{U}_{k,l}^{d}\}_{l=1}^{5}$. The resulting *Demography Context* vector for u is defined as $\mathbf{d}_c^{(t)} \overset{def}{=} \left[d_{c,1}^{(t)}, ..., d_{c,5}^{(t)}\right]$, where $d_{c,k}^{(t)}$ represents the Gower distance between u and the cluster center of $\mathcal{U}_{k,l}^{d}$. Then, the overall *Demography Aware Student Context* uses a combined representation as $\mathbf{u}_c^{(t)} \overset{def}{=} \left[u_{c,1}^{(t)}, ..., u_{c,5}^{(t)}, d_{c,1}^{(t)}, ..., d_{c,5}^{(t)}, d\right] \in \mathbb{R}^{11}$.

3.2 Feature Representation

The feature representation module in Fig. 2 uses a specific *Source* $\in \mathcal{C}$, for which the course-specific subcollection $\mathcal{D}_{sr} \subset \mathcal{D}$ is used to learn the feature representation module F_s that can effectively identify the potentially 'marginalized' students in the course c. Typically a larger subcollection of course-specific samples with a balanced distribution across various classes is considered as a *Source* subcollection D_{sr}. In Sect. 4.3, we report results using different choices of *Source* domains from $\mathcal{C}$. Long Short-Term Memory (LSTM) network model, a variant of Recurrent Network Model (RNN), is used as the feature extractor module [14]. Given a sample $\mathbf{x}_c = \{\mathbf{x}_c^{(t)}\}_t \in \mathcal{D}_{sr}$, passed as an input to F_s, each of its recurrent layers is designed to propagate historical information via a chain-like neural network architecture that integrates the current input and the hidden state $\mathbf{h}^{(t-1)}$ at $(t-1)^{th}$ time stamp [19] along with the gating functions into its state dynamics [14].

As shown in Fig. 2, F_s has a stack of $(k-2)$ LSTM layers, followed by the $(k-1)^{th}$ layer as a fully connected dense layer and k^{th} layer as a softmax layer. For each sample $\mathbf{x}_c$, the intermediate $(k-2)^{th}$ layer output $\{\mathbf{h}_{c,(k-2)}^{(t)}\}_{t=1}^{N_c}$ is fed as an input to the $(k-1)^{th}$ dense layer of F_s and produces $\mathbf{z}_c \in \mathbb{R}^n$ as a compact derived sample descriptor (the dimension n of $\mathbf{z}_c$ depends on the $(k-1)^{th}$ layer size of F_s), which is learned to be discriminative of its underlying category information y_c^i. However, having been an effective representative of several course-specific learning activity patterns, which may not generalize well across multiple courses. Each LSTM layer coupled with dropout layer has 64 hidden units. With Rectified Linear unit (ReLU), the FC layer has 16 units.

3.3 Multi-domain Adversarial Feature Representation (mDAFR) Learning

Note that the entire dataset $\mathcal{D}$ is essentially a collection of samples collected from different courses, where the samples representing a student's learning pattern in course $c \in \mathcal{C}$ belongs to its sub-collection $\mathcal{D}_c$. In a practice setting, not all these course-specific subcollections may have sufficiently large annotated collection to build a indigenous course-specific model from scratch. Therefore, in practical settings, the samples of $\mathcal{D}$ are typically originated from two types of courses (or domains): the data collection from *Source* domain denoted by $\mathcal{D}_{sr}$, which has comparably larger collection of samples representing each label; and $(|\mathcal{C}|-1)$ smaller subcollections representing samples from *Target* domains $\{\mathcal{D}_{tar,j}\}_{j=1}^{(|\mathcal{C}|-1)}$. Therefore, $\mathcal{D}$ can be decomposed as $\mathcal{D} = \{\mathcal{D}_{sr}\} \cup \{\mathcal{D}_{tar,j}\}_{j=1}^{(|\mathcal{C}|-1)}$. The proposed mDAFR model aims to leverage the larger collection of *Source* data (student activity details from the source *sr*) and smaller unlabelled sample collections of multiple *Target* domains (student activity details from multiple target *tar* courses) to build a robust classifier. Note that while there may exist multiple distributions representing the data patterns of the *Source* and various *Target* domains, which are all unknown, they all represent the identical set of semantic categories. Hence, given a test sample $\mathbf{x}_c$ our ultimate goal is to design a model that can accurately predict its label y_c irrespective of its originating domain in $\mathcal{C}$.

The mDAFR module employs a deep feed-forward architecture that for each input $\mathbf{x}_c$, predicts its label $y_c \in \{0,1\}$ and its underling domain $c \in \mathcal{C}$ [10]. As shown in Fig. 2, the input vector $\mathbf{x}_c$ is passed through the initial feature representation module F_s to generate a n-dimensional derived descriptor $\mathbf{z}_c \in \mathbb{R}^n$, which is then transformed by a mapping F_y to the label y_c. The proposed domain invariant feature representation module uses F_s (trained using $\mathcal{D}_{sr} \subset \mathcal{D}$) as the initial feature representation module. In order to achieve domain invariance, we also introduce a multiclass domain classifier module F_d that can predict the originating domain of the input sample. The weight parameters of all the $(k-1)$ layers (except the last Softmax layer) of F_s represented by θ_{F_s} along with the network parameters θ_{F_y} (and θ_{F_d}) of F_y (and F_d) are further updated jointly using a combined loss term defined as below [10]:

$$E(\theta_{F_s}, \theta_{F_y}, \theta_{F_d}) = \frac{1}{|\mathcal{D}_{sr}|} \sum_{i=1}^{|\mathcal{D}_{sr}|} \mathcal{L}_y^i(\theta_{F_s}, \theta_{F_y}) - \lambda \Bigg(\frac{1}{|\mathcal{D}_{sr}|} \sum_{i=1}^{|\mathcal{D}_{sr}|} \mathcal{L}_d^i(\theta_{F_s}, \theta_{F_d}) + \frac{1}{(|\mathcal{C}|-1)|\mathcal{D} \setminus \mathcal{D}_{sr}|} \sum_{j}^{(|\mathcal{C}|-1)} \sum_{i=1}^{|\mathcal{D}-tar,j|} \mathcal{L}_d^i(\theta_{F_s}, \theta_{F_d}) \Bigg) \quad (1)$$

where, the empirical classification loss on a labeled example x_c^i from course c is denoted as $\mathcal{L}_y^i$ and the domain discrimination loss is denoted as $\mathcal{L}_d^i$. They are defined as $\mathcal{L}_y^i(\theta_{F_s}, \theta_{F_y}) = \mathcal{L}_y(F_y(F_s(x_c^i; \theta_{F_s}); \theta_{F_y}), y_i)$ and $\mathcal{L}_d^i(\theta_{F_s}, \theta_{F_d}) = \mathcal{L}_d(F_d(\mathcal{R}(F_s(x_c^i; \theta_{F_s})); \theta_{F_d}), c)$, where c (and y_i) represents the course (and ground truth persistence category details) information for x_c^i. The term $\mathcal{L}_y$ (e.g. multinomial) and $\mathcal{L}_d$ (e.g. multi-class cross-entropy loss) are the corresponding loss functions. In all experiments, we use $\lambda = 1$. A 'pseudo function' $\mathcal{R}(\mathbf{x})$ is introduced by defining two (incompatible) equations describing its forward and backpropagation behavior [10] as $\mathcal{R}(\mathbf{x}) = \mathbf{x}$ and $\frac{dR}{d\mathbf{x}} = -\mathbf{I}$. The joint learning using the combined loss term defined in Eq. (1) can be implemented using Stochastic Gradient Descent by optimizing the saddle points $\theta_{F_s}^0, \theta_{F_y}^0, \theta_{F_d}^0$ as, $(\theta_{F_s}^0, \theta_{F_y}^0) = \underset{\theta_{F_s}, \theta_{F_y}}{\arg\min}\ E(\theta_{F_s}, \theta_{F_y}, \theta_{F_d}^0)$ and $\theta_{F_d}^0 = \underset{\theta_{F_d}}{\arg\max}\ E(\theta_{F_s}^0, \theta_{F_y}^0, \theta_{F_d})$. This enables the system attain an equilibrium between the classification performance, the mitigating system's ability for domain discrimination. This results in obtaining a domain invariant feature representation that may influence a more accurate label prediction task.

Early Prediction: For any $c \in \mathcal{C}$ and a given query $\mathbf{x}_c^q$ of length t_0 such that $\eta < t_0 < T$, during the testing phase, we decompose it into $(t_0 - \eta + 1)$ equal sized subsequences $\{\mathbf{x}_{c,i}^q\}_{i=1}^{(T-\eta+1)}$. Each $\mathbf{x}_{c,i}^q$ as a query, depicts the learning activity pattern for η consecutive time stamps, extracted from the original sequence $\mathbf{x}_c^q$. Then an average 'marginalization' score of $\{F_y(F_s(\mathbf{x}_{c,i}^q; \theta_{F_s}); \theta_{F_y})\}_{i=1}^{(t_0-\eta+1)}$ is used to classify $\mathbf{x}_c^q$. In our experiments, we use $\eta = 5$ to obtain 4 different partial subsequences from each $\mathbf{x}_c$ (or $\mathbf{x}_c^q$), each of which is treated as a separate training sample, labeled same as $\mathbf{x}_c$.

4 Experiments

4.1 Dataset

We use KDDCup[1] and the recent MoocDropout[2] dataset for our experiments. The information contained in both the datasets is of three types: 1) Object/module data; 2) log data; and 3) label data to specify course completion or dropout. Object/module data comprises of course/module-specific details (e.g. chapter, course info, peer-grading, course, video, dictation, problem, start and end of each module etc.).

KDDCup dataset is a collection of event and relation-based activity details of 39 Instructor-paced mode courses, which include the information of a total of $200,904$ enrollments and $112,448$ unique students. The MoocDropout dataset contains 698 Instructor-paced courses. This data collection has the log details for $1,319,032$ video activities, $10,763,225$ forum participation activities, $2,089,933$ assignment activities, $738,0344$ web page access related activities. Among the total $200,904$ student population, $159,223$ students dropped out before completing the course and $41,681$ completed all the requirements of their enrolled course within a given timeframe.

MoocDropout also provides students' demographic information (age, gender, education level), which, as described in Sect. 3.1, is used to describe the *Demographic Context* of students. In the instructor paced environment (IPE) of 698 courses, it has the log details for $50,678,849$ video activities, $443,554$ forum participation activities, $7,773,245$ assignment activities, $9,231,061$ web page access related activities. Among the total $467,113$ student population, $372,088$ students dropped out before completing the course and $95,025$ completed all the

Table 1. Performance comparison of the proposed mDAFR model against the Transfer Learning [31] in KDDCup dataset: In each experimental iteration a specific *Source* collection (indexed as sr) and each of the other 38 courses is treated as a *Target* domain. Columns 2–7 report the performance of the proposed method for each iteration that uses a specific sr as a source collection. Column 8 reports the average performance, that computes the mean of Columns 2–7. Similarly, Columns 2–7 in Row 2 report the performance of another set of experimental iterations, where transfer learning method is adopted for each iteration using a specific sr as a source collection to learn the base model which is then transferred to each of the other 38 *Target* locations and the base model learned at sr is finetuned by the entire non-source subcollection $\mathcal{D} \backslash \mathcal{D}_{sr}$ and finetuned model is used to classify the samples from the entire test collection across all the courses in $\mathcal{C}$.

	$sr = 6$	$sr = 11$	$sr = 13$	$sr = 16$	$sr = 18$	$sr = 22$	Average
Proposed method	0.864	0.895	0.853	0.884	0.875	0.872	**0.874**
Transfer learning [31]	0.689	0.782	0.793	0.748	0.766	0.801	0.763

[1] https://www.biendata.xyz/competition/kddcup2015/.

[2] http://moocdata.cn/data/user-activity.

requirements of their enrolled course within a given timeframe. The other subset of the dataset reports the details of 515 courses in the self-paced environment (SPE). This has the log details for $38,225,417$ video activities, $90,815$ forum participation activities, $3,139,558$ assignment activities, $5,496,287$ web page access related activities. Among the total $218,274$ student population, $205,988$ students dropped out before completing the course and $12,286$ completed all the requirements of their enrolled course within a given timeframe. IPE courses follow a similar offering pattern as the conventional classrooms, however in SPE individual students follow their individual learning schedules, which can be more than 16 weeks, typically fixed for any IPE course. The learning activities in SPE courses also include video watching (watch, stop, and jump), forum discussion (ask questions and replies), assignment completion (with correct/incorrect answers, and reset), and web page clicking (click and close a course page). The label data contains information on whether the student has completed the course or not, where label 1 indicates that the student dropped out, and 0 indicates that the student completed the course.

4.2 Implementation Details

Given the activity information of all unique enrolments in the entire course collection in the dataset, 14 features are derived to represent the action of a student at any time instance t: access; discussion; navigate; page close; problem, video; wiki; server; browser; chapter; sequential; total time; and session. The time span of each course was divided in 7 nearly equal-sized segment, in which each segment was of 4 consecutive days except the last one, which was either 2 or 3 days depending on the month length. The *Course-Specific Feature Representation Module* described in Sect. 3.2 consists of $(k-2)=3$ LSTM layers, each of which was paired with a dropout layer with a dropout ratio as 0.1. For compactness, each LSTM layer coupled with its corresponding dropout layer is treated as 1 layer. The number of hidden units in each layer was set to be 64. The $(k-1)^{th}$ Fully Connected (FC) layer is designed with 16 units and defined with Rectified Linear unit (ReLU) activation. The learning of this course-specific feature representation module occurs with 60 epochs with 20% of the training samples are used for validation at every learning epoch. To deal with data imbalance, the samples from the two classes were assigned weights derived using Sklearn [21] util function class_weight() so that the training data collection appears as a balanced representation of both the classes.

4.3 Results and Comparative Study

We use accuracy as an evaluation metric that computes the ratio of the correct predictions over all the predictions made by a classifier, for reporting the performance [11]. To compare the performance against that of several methods reported by [8], we use F1 score that is the harmonic mean of the precision and recall with its best value reached at 1 for perfect precision and recall [11].

Table 2. Average Performance of the proposed method (mDAFR) with F1 score (in %) as the evaluation metric, in KDDCup and MoocDropout dataset. To perform an equivalent comparison with other methods, mDAFR model is finetuned in an active learning setting with a small fraction of the annotated *Target* samples and Column mDAFR(A) reports the result. The result is compared against the average performance obtained by using several off-the-shelf classifiers that includes Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting Decision Tree (GBDT), Context-aware Feature Interaction Network (CFIN) [8], Deep sequential (a combination of Convolution Neural Network and LSTM, denoted as ConRec), Deep Feed Forward Neural Network (DNN-3), and Simple-LSTM [22].

Dataset	mDAFR(A)	mDAFR	CFIN	SVM	RF	GBDT	ConRec	DNN-3	Simple-LSTM
KDDCup	93.45	88.68	92.27	91/65	91.73	91.88	0.86	0.85	0.84
MoocDropout	92.27	87.53	90.48	82.86%	83.11	85.18	0.76	0.75	0.73

Table 1 reports the accuracy scores [11] to compare the performance of the proposed method against transfer learning [31] using 6 different courses as the *Source* domain, for which there are at least 800 samples representing the minority category (i.e. usually the *dropped out* category in this scenario). This specific problem scenario being prone to a severe data imbalance issue, not all $\mathcal{D}_c$ for $c \in \mathcal{C}$ may be an appropriate representative of the problem spectrum with a reasonable number of samples from each category in consideration. In a typical transfer learning setting, the base model is learned using the *Source* course data and transferred to each of the *Target* course environment for later finetuning and the resulting finetuned model is used classify the samples from the entire test collection across all the courses in $\mathcal{C}$. As observed, the proposed method demonstrates a significantly robust performance compared to transfer learning using a variety of choices for the source collections $sr = 6, 11$, etc. In fact, a comparison of the average performances reported in Column 8, clearly demonstrates the effectiveness of the proposed method over the traditional transfer learning method that frequently suffers from catastrophic forgetting and thereby fails to remain equally effective for the *Source* domain sr, on which the model was originally learned. However, the proposed mDAFR model remains to be invariant of the underlying domain information, from which the query sample was originated. The proposed method attains around 11% improved accuracy score over the transfer learning method.

Table 2 uses F1 score to report the performance of the proposed method against the state-of-the-art results described by [8], which in a supervised setting, use the combined training collection of $\mathcal{D}$ (comprising of 10–30 times more 'annotated' training samples representing students' learning activity across all the courses in two datasets) to train a Context-aware Feature Interaction Network (CFIN) in a strictly supervised setting. Note that in KDDCup dataset, CFIN obtains F1 Score of 92.27%. However, the proposed mDAFR model uses significantly smaller course-specific annotated collection $\mathcal{D}_{sr}(\subset \mathcal{D})$ to present a competitive average F1 score 88.68% over all choices of $sr \in \{6, 11, 13, 16, 18, 22\}$.

To compare the performances improvement of the proposed method in an equivalent experiment setting as in CFIN, the mDAFR model is then finetuned in an active learning environment[1] with the *Target* course data, wherein mDAFR required only 1–3% of the total *Target* annotated samples, we achieve a significant performance gain. In KDDCup dataset, the finetuned model obtains 93.45% F1 score. A similar average performance is also observed for the MoocDropout, where Instructor-paced courses with more than 1000 samples were chosen as the *Source* classes to report 92.27% average F1 Score.

5 Conclusion

The proposed method designs a continual monitoring system that employs a multi-domain adversarial feature representation (mDAFR) strategy to early identify the potentially 'marginalized' students, who may need personalized instructional support. While domain-adaptation offers a promise to assist educators in their effort for personalize pedagogical approach by highlighting some determining feature attributes, the proposed student-centric model benefits all participants involved in the course life-cycle. In addition to encouraging the emergence of features that are more exclusive and discriminative to the main learning task and invariant to the *domain shifts*, across a variety of courses, the proposed mDAFR model is also suitable for interactive learning in a distributed data environment, wherein the model can be learned in a large *Source* course and can be easily customized with a smaller data collection of *Target* courses. This shows a greater promise to be adopted in a real-life setting, where an extensive data sharing (specifically the annotated data) across departments/schools/universities may be an issue due to its confidentiality concerns. By facilitating a more personalized interaction with a small set of identified 'marginalized' student profiles, the proposed model offers practical assistance to help improve student retention.

References

1. Bhattacharjee, S.D., Tolone, W.J., Paranjape, V.S.: Identifying malicious social media contents using multi-view context-aware active learning. Future Gener. Comput. Syst. **100**, 365–379 (2019)
2. Borrella, I., Caballero, S., Ponce-Cueto, E.: Predict and intervene: addressing the dropout problem in a MOOC-based program, pp. 1–9 (June 2019). https://doi.org/10.1145/3330430.3333634
3. Chen, S., Zhou, F., Liao, Q.: Visual domain adaptation using weighted subspace alignment. In: 2016 Visual Communications and Image Processing (VCIP), pp. 1–4. IEEE (2016)
4. Dascalu, M.D., et al.: Before and during COVID-19: a cohesion network analysis of students' online participation in moodle courses. Comput. Hum. Behav. **121**, 106780 (2021)
5. De Raedt, L., Blockeel, H.: Using logical decision trees for clustering. In: Lavrač, N., Džeroski, S. (eds.) ILP 1997. LNCS, vol. 1297, pp. 133–140. Springer, Heidelberg (1997). https://doi.org/10.1007/3540635149_41

6. Dupéré, V., Dion, E., Leventhal, T., Archambault, I., Crosnoe, R., Janosz, M.: High school dropout in proximal context: the triggering role of stressful life events. Child Dev. **89**(2), e107–e122 (2018)
7. Fei, M., Yeung, D.: Temporal models for predicting student dropout in massive open online courses. In: 2015 IEEE International Conference on Data Mining Workshop (ICDMW), pp. 256–263 (2015)
8. Feng, W., Tang, J., Liu, T.X.: Understanding dropouts in MOOCs. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 517–524 (2019)
9. Frost, N., Moshkovitz, M., Rashtchian, C.: ExKMC: expanding explainable k-means clustering. arXiv preprint arXiv:2006.02399 (2020)
10. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning, ICML 2015, vol. 37, pp. 1180–1189. JMLR.org (2015)
11. Godbole, S., Sarawagi, S.: Discriminative methods for multi-labeled classification. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, pp. 22–30. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24775-3_5
12. Goodfellow, I.J., et al.: Generative adversarial networks. arXiv preprint arXiv:1406.2661 (2014)
13. Halawa, S., Greene, D., Mitchell, J.: Dropout prediction in MOOCs using learner activity features. Proc. Second Eur. MOOC Stakehold. Summit **37**(1), 58–65 (2014)
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
15. Jeon, B., Park, N., Bang, S.: Dropout prediction over weeks in MOOCs via interpretable multi-layer representation learning. arXiv preprint arXiv:2002.01598 (2020)
16. Jordan, K.: Massive open online course completion rates revisited: assessment, length and attrition (June 2015). https://doi.org/10.13140/RG.2.1.2119.6963
17. Li, Z., Hoiem, D.: Learning without forgetting. CoRR abs/1606.09282 (2016). http://arxiv.org/abs/1606.09282
18. Nagrecha, S., Dillon, J.Z., Chawla, N.V.: MOOC dropout prediction: lessons learned from making pipelines interpretable. In: Proceedings of the 26th International Conference on World Wide Web Companion, pp. 351–359 (2017)
19. Pascanu, R., Gülçehre, Ç., Cho, K., Bengio, Y.: How to construct deep recurrent neural networks. CoRR abs/1312.6026 (2013)
20. Patricia, N., Caputo, B.: Learning to learn, from transfer learning to domain adaptation: a unifying perspective. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1442–1449 (2014)
21. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
22. Prenkaj, B., Velardi, P., Distante, D., Faralli, S.: A reproducibility study of deep and surface machine learning methods for human-related trajectory prediction. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 2169–2172 (2020)
23. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987)
24. Tang, C., Ouyang, Y., Rong, W., Zhang, J., Xiong, Z.: Time series model for predicting dropout in massive open online courses. In: Penstein Rosé, C., et al. (eds.) AIED 2018, Part II. LNCS (LNAI), vol. 10948, pp. 353–357. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93846-2_66

25. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7167–7176 (2017)
26. Wang, W., Yu, H., Miao, C.: Deep model for dropout prediction in MOOCs. In: Proceedings of the 2nd International Conference on Crowd Science and Engineering, pp. 26–32 (2017)
27. Whitehill, J., Mohan, K., Seaton, D., Rosen, Y., Tingley, D.: MOOC dropout prediction: how to measure accuracy? In: Proceedings of the Fourth 2017 ACM Conference on Learning@ Scale, pp. 161–164 (2017)
28. Wood, L., Kiperman, S., Esch, R., Leroux, A., Truscott, S.: Predicting dropout using student- and school-level factors: an ecological perspective. Sch. Psychol. Q. **32**, 35–49 (2017). https://doi.org/10.1037/spq0000152
29. World Economic Forum, W.: Fourth industrial revolution (2020). https://www.weforum.org/agenda/archive/fourth-industrial-revolution
30. Xiao, Z., Wang, L., Du, J.Y.: Improving the performance of sentiment classification on imbalanced datasets with transfer learning. IEEE Access **7**, 28281–28290 (2019). https://doi.org/10.1109/ACCESS.2019.2892094
31. Zhuang, F., et al.: A comprehensive survey on transfer learning. Proc. IEEE **109**(1), 43–76 (2020)