STUDENT DROPOUT PREDICTION IN ONLINE COURSES BASED ON SUPERVISED CONTRASTIVE LEARNING

Doan Van Thanh Phong[1], Le Khac Toan[1], Van-Vinh Le[1*], Daniel Ambach[2], and Tran Van Lang[3]

[1] Faculty of Information Technology, Ho Chi Minh City University of Technology and Engineering
[2] Data Science & Computational Statistics, DBU University of Applied Sciences, Berlin Germany
[3] Journal Editorial Department, Ho Chi Minh City University of Foreign Languages - Information Technology

phong.dvt@ou.edu.vn, toanlk26.ncs@hcmute.edu.vn, vinhlv@hcmute.edu.vn, daniel.ambach@dbuas.de, langtv@huflit.edu.vn

ABSTRACT — Online learning provides numerous benefits and has become increasingly popular in recent years. However, the dropout rate in these learning environments is very high and thus represents a major issue for educational institutions. Early identification of students who are likely to drop out enables educators or course providers to offer appropriate support and improve student engagement. Some existing approaches for predicting student dropouts are based on traditional machine learning or deep learning techniques. However, unique characteristics of Massive Open Online Courses (MOOC) data pose significant challenges for effective dropout prediction. This work proposes a deep learning-based method, called DP-SCL, for the student dropout prediction. The proposed method applies a supervised contrastive learning method in which a shared encoder with Long Short-Term Memory and Multi-Head Attention layers is trained using supervised contrastive loss. Experimental results on two datasets from different educational platforms demonstrate the strength of the proposed model compared with twelve baseline methods. The source code of DP-SCL can be directly downloaded from our github page https://github.com/dvt-phong/DP-SCL.

Keywords — MOOC, student dropout prediction, supervised contrastive learning.

I. INTRODUCTION

Massive Open Online Courses (MOOCs) have become increasingly popular and are now widely adopted across educational institutions. Students in universities or learners around the world can access thousands of online courses at low cost or even for free [1]. Despite these advantages, one of the major concerns in MOOCs platforms is the extremely high dropout rates which can reach up to 90% in many courses [2]. This leads to a waste of resources and requires better tools for monitoring learning activities of students [2], [3]. Thus, student dropout prediction, which allows universities to provide timely support for learners is a crucial task to improve educational quality in online environments. However, characteristics of online learning data, such as imbalance, noise, and diversity, pose significant research challenges.

Some initial student dropout prediction methods in online courses are mainly based on traditional machine learning techniques. Kloft et al. [4] employed the Support Vector Machine (SVM) method to predict student dropout risk in MOOCs. The method utilizes history features to capture changes in students' behavior over weeks, and adopts a soft margin mechanism to tolerate misclassification in complex cases. DT-ELM [5] is a hybrid algorithm for the student dropout prediction which combines decision tree and extreme learning machine techniques. In the method, Decision Tree (DT) provides the ability to select features based on its tree structure and weighting mechanism.

Several ensemble learning techniques such as AdaBoost and XGBoost [6] are also applied for the dropout prediction. Those methods sequentially combine multiple models to build a stronger predictive model and achieve more accurate predictions. AdaBoost trains the models sequentially, where higher weights are assigned to misclassified samples so that subsequent models focus more on correcting the errors made by previous ones. The final prediction is then determined based on a voting mechanism. Otherwise, XGBoost [6] operates in a similar manner but constructs a series of decision trees, where each new tree attempts to correct the errors of the previous trees by optimizing a loss function. MMSE [7] is a stacking ensemble method with the combination of different models for the dropout prediction. One limitation of these methods is their limited ability to capture the temporal patterns in activity data of students leading to reduced prediction performance.

The advance of deep learning methods brings promising solutions for the student dropout prediction in online courses. The work of Mubarak et al. [8] applies the Long Short-Term Memory (LSTM) model which is able to capture behavioral information of students on online systems through its memory cell mechanism, enabling the model to learn complex features. Xiong et al. [9] treats the task as a time series prediction problem and applies a model based on LSTMs and RNN for its classification process. Otherwise, the method uses optimization techniques to find better prediction results. MFCN-VIB [10] employs three parallel Fully Convolutional Networks (FCNs) with

* Corresponding Author

different kernel sizes to extract features from learning behavior time-series data from student activities on online systems. This enables the model to capture both long-term trends and short-term fluctuations.

A group of methods combines CNN and Recurrent Neural Networks (RNN) or LSTM for the student dropout prediction in MOOCs. The work of Niu et al. [11] proposes a deep learning approach that combines Convolutional Neural Networks (CNN) and RNN architectures. While CNN is employed to extract features from daily learning activities of students, RNN is used to capture temporal changes in their behaviors to support dropout probability prediction. CNNAE-LSTM [12] leverages a CNN-based autoencoder to extract feature representations, and then the latent embedding is subsequently passed to the CNN decoder and the LSTM for final prediction. The work of Talebi et al. [13] also employs a combination of CNN and LSTM modules. However, it integrates ensemble learning strategies, including boosting, AdaBoost, and voting, to fuse outputs of the two models and thus enhance prediction performance.

Some other student dropout prediction methods utilize the strength of CNN and BiLSTM. Among the methods, CLSA [14] uses a CNN for feature extraction, followed by a BiLSTM to model temporal relationships. A static attention module is then employed to focus on important features. The study of Zheng et al. [15] also employs a CNN to extract data representations, followed by a self-attention mechanism to emphasize important features. The resulting features are then passed through a BiLSTM to capture bidirectional temporal relationships supporting the classification.

Graph neural network-based models are also applied for the dropout prediction such as SDP-GAT-KLA [16], MST-GCN [17], SIG-Net [18]. SDP-GAT-KLA is a method that constructs a bipartite graph to model student and course relationships. The method then employs a Graph Attention Network to learn data representations. Furthermore, to address the class imbalance in the dataset, the authors propose a KNN Link Augmentation strategy which creates additional links between students with similar characteristics. MST-GCN constructs a series of heterogeneous graphs for snapshots. Each snapshot captures information about students, courses, and learning objects at a specific stage of the learning process. The model learns from these snapshots to capture relational information to generate data representations supporting prediction of the probability of student dropout. SIG-Net is another GNN-based approach for MOOCs dropout prediction which builds a student interaction graph consisting of three types of nodes representing enrollment, object, and course. The model applies relational graph convolutional networks to learn embeddings from subgraphs generated at different time intervals and generate classification results.

In the fields of image recognition and text classification, contrastive learning has become a promising approach for learning high-quality feature representations. In particular, supervised contrastive learning can learn better feature representations by bringing samples from the same class closer together and separating samples from different classes, even when the data are imbalanced. However, the potential of supervised contrastive learning for the student dropout prediction in MOOCs has not been fully explored.

This work proposes a student dropout prediction method in online courses, called DP-SCL, which is based on supervised contrastive learning techniques. The proposed method aims to learn features using a shared encoder implemented by Long Short-Term Memory and Multi-Head Attention layers. A supervised contrastive loss is employed to jointly optimize the model, supporting better classification performance. Experiments conducted on two real datasets show the strength of the proposed method compared with baseline methods. The details of the proposed method are presented in Section II. The experimental findings are presented and analyzed in Section III, followed by concluding remarks in Section IV.

II. METHODS

A. PROBLEM FORMULATION

Consider a MOOC platform with a set of students S and a set of courses C, where $s \in S$ denotes a student and $c \in C$ denotes a course. Let *(s, c)* denote an enrollment of a student s in a course c. For each enrollment *(s, c)*, the system records the F activities of the student in the course over T days, e.g., watching videos, completing assignments, accessing documents, and participating in forums. The entire activity sequence of an enrollment *(s, c)* is represented as a matrix $X \in \mathbb{R}^{T \times F}$. In which, $x_{t,f} \in X$, $t \in [1 \dots T]$, and $f \in [1 \dots F]$ denotes the frequency of activity f on day t.

For each enrollment, the platform records a learning outcome label $y_i \in \{0, 1\}$ as follows:

$$y_i = \{1, \text{if student } s_i \text{ drops out from course } c_i;\ 0, \text{if student } s_i \text{ completes the course } c_i\}.$$

Given the dataset $D = \{(X_i, y_i)\}_{i=1}^{N}$, in which N is the number of enrollments, the objective is to learn a mapping function $f_\theta: \mathbb{R}^{T\times F} \rightarrow [0,1]$ parameterized by θ, such that for each input X_i, the function f_θ predicts the dropout probability: $\hat{y}_i = f_\theta(X_i) \in [0,1]$.

The goal of this task is to predict the probability that a student s_i will drop out of a MOOC course c_i, based on their activity sequence during T days. The final predicted label is determined by a threshold δ:

$$\hat{y}_i^{class} = 1[\hat{y}_i \geq \delta].$$

B. PROPOSED METHOD

The proposed method first applies data augmentation to generate two augmented views. These views are passed through a shared encoder, constructed by LSTM, Multi-Head Attention, and Attentive Pooling, to generate feature representations. The extracted features are then fed into a projection head before being used to compute supervised contrastive loss. Simultaneously, the output from the encoder is passed through a classifier to predict the dropout probability, optimized using BCE Loss. An overview of the DP-SCL architecture is presented in Figure 1.

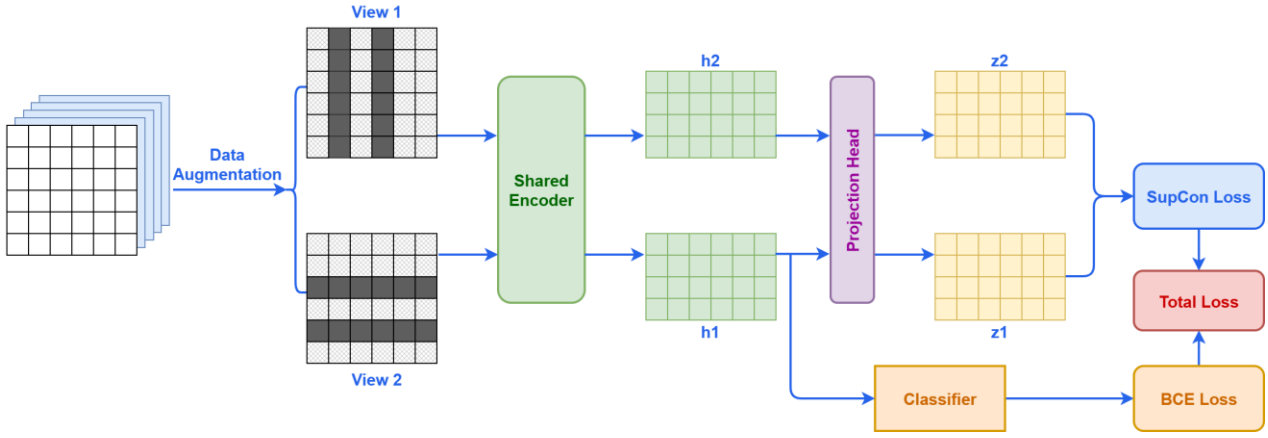

Figure 1. *Overview of the DP-SCL architecture*

1. DATA AUGMENTATION

This work applies augmentation techniques to each data sample X using three stochastic techniques as follows:

Time Masking: Each day $t \in \{1, \dots, T\}$ has a probability p_t of having its entire feature vector masked, simulating a day when the student has no activity on the platform: $\tilde{x}_{t,:} = m_t x_{t,:}$. Here $m_t = 0$ with probability p_t and $m_t = 1$ with probability $1\text{-}p_t$.

Feature Masking: Each activity type $f \in \{1, \dots, F\}$ has a probability p_f of being entirely masked along the temporal dimension, simulating a scenario where certain activities are not recorded: $\tilde{x}_{:,f} = m_f x_{:,f}$. Here $m_f = 0$ with probability p_f and $m_f = 1$ with probability $1\text{-}p_f$.

Gaussian Noise: After masking, a small amount of noise is added to each element of the feature matrix X to increase the diversity of the views: $\tilde{x}_{t,f} = \tilde{x}_{t,f} + \varepsilon_{t,f}$. In this setting, the noise is independently sampled for each element $x_{t,f}$.

This work generates views $X^{(1)}$ using time masking and Gaussian noise, and $X^{(2)}$ using feature masking and Gaussian noise.

2. SUPERVISED CONTRASTIVE LEARNING

Two views $X^{(1)}$ and $X^{(2)}$ are fed into an encoder which consists of three sequential processing layers:

Layer 1 – LSTM: The input sequence $X^{(v)} \in \mathbb{R}^{T\times F} (v \in \{1,2\})$ is passed through an LSTM network to model the temporal relationship in the learning activity sequence. Let $x_t = [x_{t,1}, x_{t,2}, \dots, x_{t,F}] \in \mathbb{R}^F$ represent the activity vector of day t. At each time step t, the LSTM updates the hidden state through three gating mechanisms:

$$f_t = \sigma(W_f x_t^{(v)} + U_f r_{t-1}) + b_f,$$

$$i_t = \sigma(W_i x_t^{(v)} + U_i r_{t-1}) + b_i,$$

$$o_t = \sigma(W_o x_t^{(v)} + U_o r_{t-1}) + b_o,$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c x_t^{(v)} + U_c r_{t-1} + b_c),$$

$$r_t = o_t \odot \tanh(c_t) \in \mathbb{R}^H.$$

in which f_t, i_t, o_t denote the forget gate, input gate, and output gate, respectively; c_t is the cell state; and W_*, U_*, b_* are the learnable parameters. The complete sequence of hidden states is collected as: $R^{(v)} = \left[r_1^{(v)}, \dots, r_T^{(v)}\right] \in \mathbb{R}^{T \times H}$, where H is the hidden dimension.

Layer 2 - Multi-Head Attention: The sequence of hidden states $R^{(v)}$ from the LSTM is passed through a Multi-Head Attention mechanism [19] with $K = 4$ attention heads. In the mechanism, Query *(Q)*, Key *(K)*, and Value *(V)* matrices are projected from $R^{(v)}$:

$$Q = R^{(v)}W^Q, K = R^{(v)}W^K, V = R^{(v)}W^V,$$

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V.$$

The output of the Multi-Head Attention module is combined with the original input through a residual connection, followed by Layer Normalization:

$$Z^{(v)} = MHA\left(R^{(v)}, R^{(v)}, R^{(v)}\right) + R^{(v)},$$

$$A^{(v)} = LayerNorm(Z^{(v)}).$$

Layer 3 - Attentive Pooling: This work utilizes Attentive Pooling technique [20] to aggregate the outputs of the LSTM and the Multi-Head Attention module into a single representation vector. A learnable query vector $q \in \mathbb{R}^{1 \times H}$ is randomly initialized and jointly trained with the entire model, allowing the model to automatically identify the most important temporal regions in the sequence:

$$\alpha_t^{(v)} = \frac{exp\ (q.A_t^{(v)}/\sqrt{H})}{\sum_{t'=1}^{T} exp\ (q.A_{t'}^{(v)}/\sqrt{H})},$$

$$h^{(v)} = \sum_{t=1}^{T} \alpha_t{}^{(v)}.A_t^{(v)}.$$

in which $\alpha_t^{(v)}$ is the attention weight at time step t and $A_t^{(v)}$ is the vector at time step t of $A^{(v)}$. The encoder generates two representation vectors corresponding to $X^{(1)}$, $X^{(2)}$ as follows:

$$h_1 = Encoder\left(X^{(1)}; \theta_{enc}\right), h_2 = Encoder\left(X^{(2)}; \theta_{enc}\right),$$

$$h_1, h_2 \in \mathbb{R}^H.$$

The representation vectors $h_1, h_2 \in \mathbb{R}^H$ are then passed through a projection head and produce z_1, z_2:

$$z_1 = \frac{\mathrm{W}_2.\,\mathrm{ReLU}(\mathrm{W}_1 h_1 + \mathrm{b}_1) + \mathrm{b}_2}{\|\mathrm{W}_2.\,\mathrm{ReLU}(\mathrm{W}_1 h_1 + \mathrm{b}_1) + \mathrm{b}_2\|2} \in \mathbb{R}^{\mathrm{H}},$$

$$z_2 = \frac{\mathrm{W}_2.\,\mathrm{ReLU}(\mathrm{W}_1 h_2 + \mathrm{b}_1) + \mathrm{b}_2}{\|\mathrm{W}_2.\,\mathrm{ReLU}(\mathrm{W}_1 h_2 + \mathrm{b}_1) + \mathrm{b}_2\|2} \in \mathbb{R}^{\mathrm{H}}.$$

For each training sample X_i, two augmented views are generated and passed through the shared encoder and the projection head to obtain two embeddings, denoted by $z_{1,i}$ and $z_{2,i}$. These embeddings are then used to compute the supervised contrastive loss, which aims to pull embeddings of the same class closer together and push embeddings of different classes farther apart in the representation space [21]. Let I be the set of indices of all embeddings. For each anchor embedding z_i, the set of all remaining embeddings is defined as $A(i)=I\backslash\{i\}$, and the set of positive embeddings is defined as $P(i) = \left\{p \in A(i): \tilde{y}_p = \tilde{y}_i\right\}$, where $\tilde{y}_i$ is the class label of anchor i. Based on these definitions, the supervised contrastive learning loss function is defined as follows [21]:

$$L_{SupCon} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} log\ \frac{exp\ (z_i.z_p/\tau)}{\sum_{a \in A(i)} exp\ (z_i.z_a/\tau)}.$$

In this formulation, τ is a temperature parameter that controls the sharpness of the similarity distribution [22], z_i denotes the anchor embedding, z_p denotes a positive embedding that shares the same class label with the anchor, and z_a denotes any embedding in $A(i)$.

3. CLASSIFICATION LOSS

Let L_{BCE} be the Binary Cross Entropy loss computed over all training samples as follows:

$$L_{BCE}(\hat{y}, y) = -\frac{1}{N}\sum_{i=1}^{N}[y_i.log\hat{y}_i + (1-y_i)log\ (1-\hat{y}_i)].$$

Where N is the number of training samples, $y_i \in \{0,1\}$ is the ground-truth label, and $\hat{y}_i$ is the predicted probability for sample i. The model is trained with a combined loss function that integrates two complementary objectives [21]:

$$L_{total} = L_{BCE}(\hat{y}, y) + \lambda.L_{SupCon}(z_1, z_2, y).$$

Where λ is a weight value to control the balance between the classification objective and the representation learning objective. The algorithm of the training process of DP-SCL is presented in.

C. ALGORITHMS

The algorithm of the training process of DP-SCL for each epoch is presented in Algorithm 1. Given the training set $D = \{(X_i, y_i)\}^N_{i=1}$, and four hyperparameters λ, τ, η, and θ. For each training sample (X_i, y_i), we use time masking and feature masking with a small noise to generate two views $X_i^{(1)}$ and $X_i^{(2)}$. Both views are passed into a shared encoder (using LSTM, Multi-head Attention and Attentive Pooling) to get vector feature representations $h_{1,i}$ and $h_{2,i}$. After that, a projection head maps these features into an embedding space, and output $z_{1,i}$ and $z_{2,i}$. At the same time, a classifier takes $h_{1,i}$ as input to predict the dropout probability $\hat{y}_i$.

During the training process, this work computes two losses L_{BCE} and L_{SupCon}. LBCE uses the predicted dropout probability and the truth label, while L_{SupCon} uses z_1 and z_2 from the projection head. The two losses are combined to form L_{total} = LBCE + λ.LSupCon. After that, parameter θ is updated for each epoch.

Algorithm 1: Training process of the DP-SCL for each epoch

Input: Training set $D = \{(X_i, y_i)\}_{i=1}^N$; hyperparameters λ, τ, η, θ

Output: Updated model parameters θ

for each (X_i, y_i) in D **do**

$X_i^{(1)}, X_i^{(2)} \leftarrow Augmentation(X_i)$

$h_{1,i} \leftarrow Encoder\left(X_i^{(1)}\right), h_{2,i} \leftarrow Encoder(X_i^{(2)})$

$z_{1,i} \leftarrow ProjectionHead\left(h_{1,i}\right), z_{2,i} \leftarrow ProjectionHead\left(h_{2,i}\right)$

$\hat{y}_i \leftarrow Classifier(h_{1,i})$

end for

$L_{BCE} \leftarrow \frac{1}{N}\sum_{i=1}^{N} BCE(\hat{y}_i, y_i)$

$L_{SupCon} \leftarrow SupCon\left(\{z_{1,i}, z_{2,i}\}_{i=1}^N, \{y_i\}_{i=1}^N, \tau\right)$

$L_{total} = L_{BCE} + \lambda.L_{SupCon}$

$\theta \leftarrow \theta - \eta\nabla_\theta L_{total}$

III. EXPERIMENTAL RESULTS

A. DATASET

The experiments are conducted on the XuetangX dataset [23], and OULAD dataset [24], presented in Table 1. XuetangX dataset comprises 225,642 enrollments of 77,083 students and 247 courses. It contains 22 types of learning activities recorded on a daily basis over the first five weeks from the date of enrollment. The dropout rate in the dataset is approximately 75.8%, reflecting the class imbalance in data from real-world MOOC platforms [23]. OULAD is a dataset from The Open University in the UK with 32,593 enrollments from 28,785 students and 22 module-presentations across 7 modules. The dataset records 20 types of activities over 5 weeks. All datasets are split into 3 sets of training, validation, and test set with a 60:10:30 ratio.

Table 1. *The details of the XuetangX and OULAD datasets*

Statistic	XuetangX	OULAD
Number of enrollments	225,642	32,593
Number of students	77,083	28,785
Number of courses	247	22 module-presentations / 7 modules
Number of activity types	22	20
Observation period	5 weeks (35 days)	5 weeks (35 days)
Dropout rate (positive class)	~75.8%	~31.2%
Training set	135,384	19,555
Testing set	67,693	9,778
Validation set	22,565	3,260

B. EVALUATION METRICS

The performance of the classification methods is evaluated using four metrics of AUC, Precision, Recall, and F1-Score [12]. AUC is a metric used to evaluate the ability of the model to distinguish between the two classes across multiple decision thresholds. The metric is less sensitive to class imbalance and is therefore commonly used in imbalanced classification problems. Let TP, TN, FP, and FN denote the number of True Positives, True Negatives, False Positives, and False Negatives, respectively [12]. The metrics are defined as follows.

The AUC is computed as the area under the Receiver Operating Characteristic (ROC) curve, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR) across all decision thresholds:

$$TPR = \frac{TP}{TP + FN}, FPR = \frac{FP}{FP + TN}.$$

The AUC is formally defined as: $AUC = \int_0^1 TPR(t)d(FPR(t))$ and is numerically approximated via the trapezoidal rule over K thresholds:

$$AUC \approx \sum_{k=1}^{K} \frac{(FPR_k - FPR_{k-1})(TPR_k + TPR_{k-1})}{2}.$$

Besides, Precision metric measures the proportion of correctly predicted dropouts out of the total number of predicted dropouts:

$$Precision = \frac{TP}{TP + FP}.$$

The Recall metric measures the proportion of actual dropouts that are correctly identified:

$$Recall = \frac{TP}{TP + FN}.$$

F1-score is the harmonic mean of Precision and Recall, balancing both objectives:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}.$$

Because AUC is regarded as the most effective evaluation metric for the high imbalance of classes in data [12], it is selected as the primary metric in this study for benchmark comparison between student dropout prediction methods.

C. EXPERIMENTAL SETTINGS

The performance of the proposed method is compared with twelve models. Among the methods, there are seven traditional machine learning approaches including Decision Tree (DT) [5], AdaBoost [6], Support Vector Machine (SVM) [4], Logistic Regression (LR) [25], Gradient Boosting Decision Tree (GBDT) [6], XGBoost [6], KNN [26] and five deep learning based methods, i.e., LSTM [27], GRU [28], CNN [11], CNN-RNN [11], CNN-LSTM-AT1 [14]. All models are evaluated on the two datasets and trained with five fixed seeds of 1, 11, 111, 1111, and 11111. The average values across five runs are computed to report the results. The hyperparameter settings are as follows: hidden dimension $H = 128$, number of attention heads $K = 4$, masking rate for both time and feature $p_t = p_f = 0.15$, noise standard deviation $\sigma = 0.05$, temperature $\tau = 0.07$, contrastive weight $\lambda = 0.1$, classifier dropout rate of 0.3. The models are trained for 200 epochs with a batch size of 256, using the Adam optimizer and a learning rate of 1×10^{-4}. Besides, early stop technique is also applied for all methods.

D. RESULTS

The performance of DP-SCL and the baseline methods on the XuetangX and OULAD datasets are presented in Table 2, and Table 3, respectively. It can be seen in the table that the proposed method achieves the best performance for most metrics. In particular, for the primary metric, DP-SCL achieves an AUC improvement of 1.05% to 14.72% on XuetangX dataset, and from 1.27% to 8.41% on OULAD dataset over the baseline methods. Besides, deep learning based approaches achieve better results than traditional machine learning methods. Particularly, KNN and DT return the lowest AUC values of 0,7161 and 0,7581, respectively, on XuetangX dataset. In the case of the OULAD dataset, the two methods also achieve the lowest AUC with 0.6969 and 0.7185. It indicates the limitations of the traditional methods which rely on manually engineered features. XGBoost, an ensemble model, shows considerable improvements with an AUC value of 0,8437 on XuetangX, and 0.7665 on OULAD but still lower than most of the deep learning models.

When compared specifically with the deep learning methods, DP-SCL achieves higher results across most metrics. For instance, the proposed method has higher AUC values from 1.05% to 1.81% compared to the remaining deep learning methods on the XuetangX dataset. Besides, DP-SCL achieves the lowest standard deviation (± 0.0010) and shows a more stable training process compared to the baselines on the dataset. In the case of KNN on XuetangX, and LSTM on OULAD, although DP-SCL gets lower Recall values than the two models, the proposed method achieves much higher Precision values compared to the methods. This results in better F1 scores of the proposed method compared to models.

Table 2. *Comparison of models on XuetangX dataset*

Model	AUC	F1	Precision	Recall
KNN	0,7161 ± 0,0015	0,8738 ± 0,0009	0,7916 ± 0,0034	**0,9751 ± 0,0034**
DT	0,7581 ± 0,0012	0,8945 ± 0,0008	0,8398 ± 0,0030	0,9568 ± 0,0049
AdaBoost	0,7918 ± 0,0053	0,8906 ± 0,0006	0,8309 ± 0,0010	0,9596 ± 0,0008
SVM	0,8211 ± 0,0017	0,8948 ± 0,0004	0,8484 ± 0,0063	0,9466 ± 0,0080
LR	0,8217 ± 0,0015	0,8960 ± 0,0002	0,8494 ± 0,0026	0,9481 ± 0,0030
GBDT	0,8395 ± 0,0011	0,9042 ± 0,0005	0,8554 ± 0,0044	0,9589 ± 0,0048
XGBoost	0,8437 ± 0,0008	0,9055 ± 0,0005	0,8621 ± 0,0017	0,9535 ± 0,0027
LSTM	0,8465 ± 0,0010	0,9060 ± 0,0004	0,8600 ± 0,0037	0,9571 ± 0,0046
GRU	0,8452 ± 0,0012	0,9060 ± 0,0004	0,8608 ± 0,0022	0,9561 ± 0,0027
CNN	0,8528 ± 0,0014	0,9060 ± 0,0004	0,8616 ± 0,0041	0,9552 ± 0,0056
CNN-RNN	0,8494 ± 0,0056	0,9054 ± 0,0008	0,8624 ± 0,0026	0,9529 ± 0,0022
CNN-LSTM-AT1	0,8517 ± 0,0042	0,9052 ± 0,0013	0,8615 ± 0,0055	0,9536 ± 0,0064
DP-SCL (our)	**0,8633 ± 0,0010**	**0,9080 ± 0,0005**	**0,8635 ± 0,0048**	0,9574 ± 0,0063

Table 3. *Comparison of models on OULAD dataset*

Model	AUC	F1	Precision	Recall
KNN	0.6969 ± 0.0074	0.5272 ± 0.0048	0.5543 ± 0.0789	0.5147 ± 0.0573
DT	0.7185 ± 0.0073	0.5712 ± 0.0088	0.5581 ± 0.0372	0.5879 ± 0.0290
AdaBoost	0.7475 ± 0.0068	0.5775 ± 0.0061	0.5955 ± 0.0549	0.5677 ± 0.0511
SVM	0.7141 ± 0.0093	0.5574 ± 0.0061	0.5080 ± 0.0345	0.6235 ± 0.0528
LR	0.7234 ± 0.0029	0.5673 ± 0.0072	0.5288 ± 0.0352	0.6173 ± 0.0478
GBDT	0.7620 ± 0.0069	0.5899 ± 0.0045	0.5989 ± 0.0510	0.5881 ± 0.0532
XGBoost	0.7665 ± 0.0077	0.5956 ± 0.0068	0.5960 ± 0.0387	0.5995 ± 0.0434
LSTM	0.7661 ± 0.0041	0.5985 ± 0.0066	0.5696 ± 0.0343	**0.6338 ± 0.0346**
GRU	0.7673 ± 0.0072	0.6010 ± 0.0051	0.5935 ± 0.0276	0.6105 ± 0.0242
CNN	0.7664 ± 0.0082	0.5975 ± 0.0073	0.5744 ± 0.0100	0.6229 ± 0.0172
CNN-RNN	0.7682 ± 0.0066	0.5993 ± 0.0064	0.5828 ± 0.0306	0.6188 ± 0.0248
CNN-LSTM-AT1	0.7683 ± 0.0072	0.5970 ± 0.0057	0.5972 ± 0.0275	0.5983 ± 0.0222
DP-SCL (our)	**0.7810 ± 0.0080**	**0.6122 ± 0.0063**	**0.6384 ± 0.0355**	0.5899 ± 0.0213

E. ABLATION STUDY

In order to evaluate the contribution of each component to the proposed method, this work conducts experiments to compare the full model with two of its variants. The first variant (w/o SupCon) is the case of not using the supervised learning module, and the second one (w/o BCE) is for the case of not using BCE loss. The bar chart shows that removing either module results in decreased classification performance, with reductions ranging from 2.23% to 29.28%. The results demonstrate the effectiveness of each component in the DP-SCL model.

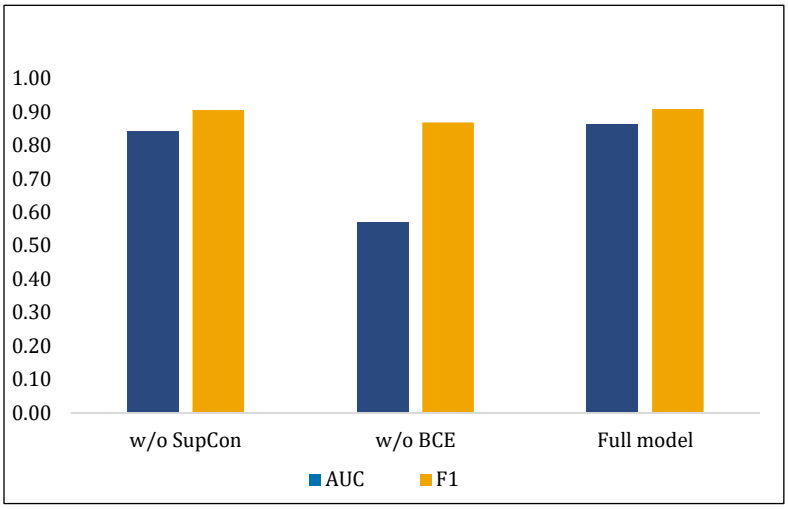

Figure 2. *The performance of DP-SCL (Full model) and its variants on dataset XuetangX*

Figure 3 is the loss curve of the ablation study of DP-SCL, including SupCon Loss L_{SupCon}, BCE Loss L_{BCE}, and Total Loss L_{Total} with the red dot marking the early stopping. The method with L_{BCE} early stops at epoch 57. It shows that the model learns to classify quickly from the extracted feature vectors. The one with L_{SupCon} and L_{Total} early stop at epoch 93 and 96, respectively, presenting that the models take more time to learn the vector representation by pulling students with the same pattern closer and pushing the others away.

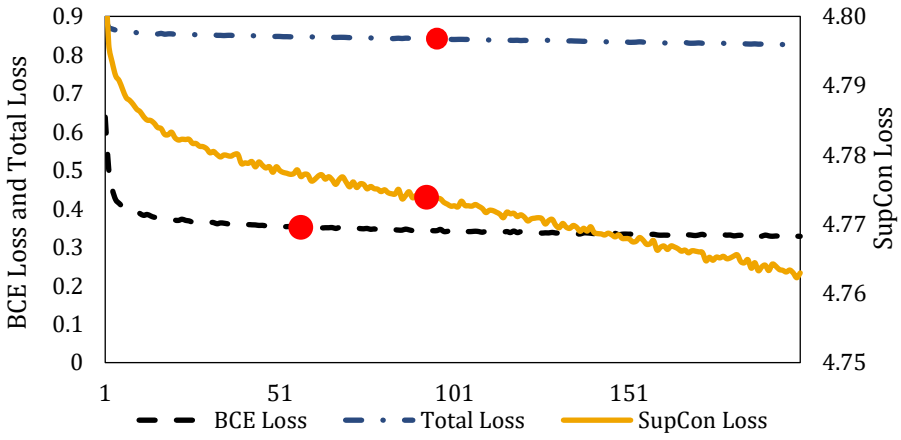

Figure 3. *Loss Curves of DP-SCL on dataset XuetangX*

F. PARAMETER SENSITIVITY

We conduct further experiments to evaluate the effects of hyperparameters λ and τ on the proposed method. While λ controls the balance between the classification objective and the representation learning objective, τ controls the sharpness of the distribution of the contrastive learning model. The value of λ is set in the range of [0.01, 0.5], τ in the range of [0.03, 0.2], and the other hyperparameters are fixed. The experimental results are presented in Figure 4 for λ, and Figure 5 for τ. It can be seen from the charts that DP-SCL is stable with the changes of both λ and τ. The model achieves slightly higher AUC and F1 values than other cases when λ is set to 0.1 and τ is set to 0.07.

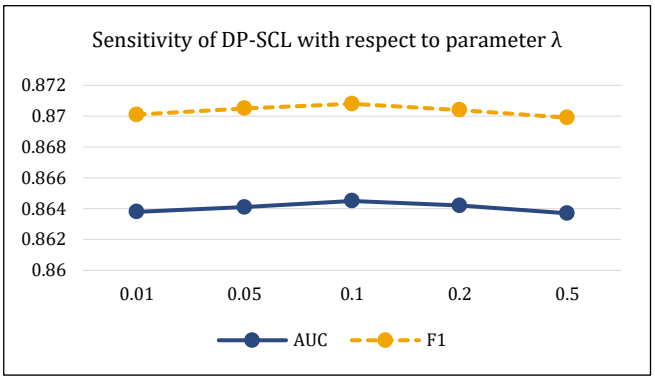

Figure 4. *Sensitivity of DP-SCL with respect to parameter λ on dataset XuetangX*

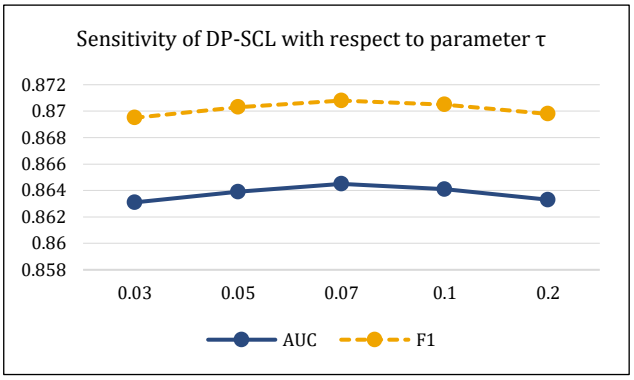

Figure 5. *Sensitivity of DP-SCL with respect to parameter τ on dataset XuetangX*

IV. CONCLUSION

This work proposes an effective method for student dropout prediction in online courses based on supervised contrastive learning techniques. The proposed method models learning activity sequences through a shared parameter encoder consisting of an LSTM and Multi-Head Attention. It then learns discriminative feature representations through supervised contrastive learning loss to encourage data points from the same class to be clustered together while separating those from different classes. Experiments on two datasets of XuetangX and OULAD demonstrate that the proposed method achieves competitive results compared with baseline methods. In future work, we plan to jointly capture temporal learning behaviors and the social interaction structure among students within a course to enhance classification performance. Besides, applying more advanced augmentation strategies for the learning activity time series data to further improve the quality of positive pairs in contrastive learning, as well as explainable AI techniques to interpret and analyze the model are also our research directions.

V. REFERENCES

[1] Clow, D. (2013), MOOCs and the funnel of participation, Proceedings of the Third International Conference on Learning Analytics and Knowledge, Association for Computing Machinery: Leuven, Belgium. pp. 185-189.

[2] Onah, D.F.O., J. Sinclair, R. Boyatt (2014), Dropout rates of massive open online courses : behavioural patterns, 6th International Conference on Education and New Learning Technologies, IATED Academy: Barcelona, Spain. pp. 5825-5834.

[3] Miladi, F., D. Lemire, V. Psyché (2023), Learning Engagement and Peer Learning in MOOC: A Selective Systematic Review, *Augmented Intelligence and Intelligent Tutoring Systems*, Cham, pp. 324-332.

[4] Kloft, M., F. Stiehler, Z. Zheng, N. Pinkwart (2014), Predicting MOOC dropout over weeks using machine learning methods, *Proceedings of the EMNLP 2014 workshop on analysis of large scale social interaction in MOOCs*, pp. 60-65.

[5] Chen, J., J. Feng, X. Sun, N. Wu, Z. Yang, S. Chen (2019), MOOC dropout prediction using a hybrid algorithm based on decision tree and extreme learning machine, *Mathematical Problems in Engineering*, Vol. 2019, No. 1, pp. 8404653.

[6] Alamri, A., M. Alshehri, A. Cristea, F.D. Pereira, E. Oliveira, L. Shi, C. Stewart (2019), Predicting MOOCs dropout using only two easily obtainable features from the first week's activities, *International Conference on Intelligent Tutoring Systems*, pp. 163-173.

[7] Zhou, Y., Z. Xu (2020), Multi-model stacking ensemble learning for dropout prediction in MOOCs, *Journal of physics: conference series*, pp. 012004.
[8] Mubarak, A.A., H. Cao, S.A. Ahmed (2021), Predictive learning analytics using deep learning model in MOOCs' courses videos, *Education and Information Technologies*, Vol. 26, No. 1, pp. 371-392.
[9] Xiong, F., K. Zou, Z. Liu, H. Wang (2019), Predicting learning status in MOOCs using LSTM, *Proceedings of the ACM Turing Celebration Conference-China*, pp. 1-5.
[10] Shou, Z., P. Chen, H. Wen, J. Liu, H. Zhang (2022), MOOC Dropout Prediction Based on Multidimensional Time-Series Data, *Mathematical Problems in Engineering*, Vol. 2022, No. 1, pp. 2213292.
[11] Niu, K., Y. Zhou, G. Lu, W. Tai, K. Zhang (2023), PMCT: Parallel Multiscale Convolutional Temporal model for MOOC dropout prediction, *Computers and Electrical Engineering*, Vol. 112, No., pp. 108989.
[12] Niu, K., G. Lu, X. Peng, Y. Zhou, J. Zeng, K. Zhang (2023), CNN autoencoders and LSTM-based reduced order model for student dropout prediction, *Neural Computing and Applications*, Vol. 35, No. 30, pp. 22341-22357.
[13] Talebi, K., Z. Torabi, N. Daneshpour (2024), Ensemble models based on CNN and LSTM for dropout prediction in MOOC, *Expert Systems with Applications*, Vol. 235, No., pp. 121187.
[14] Fu, Q., Z. Gao, J. Zhou, Y. Zheng (2021), CLSA: A novel deep learning model for MOOC dropout prediction, *Computers & Electrical Engineering*, Vol. 94, No., pp. 107315.
[15] Zheng, Y., Z. Shao, M. Deng, Z. Gao, Q. Fu (2022), MOOC dropout prediction using a fusion deep model based on behaviour features, *Computers and Electrical Engineering*, Vol. 104, No., pp. 108409.
[16] Khoushehgir, F., Z. Noshad, A. Mafakheri, S. Sulaimany (2025), Enhancing Student Dropout Prediction in MOOCs Using Graph Neural Networks and Link Augmentation, *SN Computer Science*, Vol. 6, No. 8, pp. 1000.
[17] Duan, Y., X. Chen (2026), An adaptive multi-scale spatio-temporal graph network for robust MOOC dropout prediction, *Sci Rep*, Vol. 16, No. 1, pp.
[18] Roh, D., D. Han, D. Kim, K. Han, M.Y. Yi (2024), SIG-Net: GNN based dropout prediction in MOOCs using student interaction graph, *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, pp. 29-37.
[19] Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin (2017), Attention is all you need, *Advances in neural information processing systems*, Vol. 30, No., pp.
[20] Wu, C., F. Wu, T. Qi, X. Cui, Y. Huang (2020), Attentive pooling with learnable norms for text representation, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2961-2970.
[21] Khosla, P., P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan (2020), Supervised Contrastive Learning, Advances in Neural Information Processing Systems, Curran Associates, Inc. pp. 18661-18673.
[22] Chen, T., S. Kornblith, M. Norouzi, G. Hinton (2020), A simple framework for contrastive learning of visual representations, *International conference on machine learning*, pp. 1597-1607.
[23] Feng, W., J. Tang, T.X. Liu (2019), Understanding Dropouts in MOOCs, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, No. 01, pp. 517-524.
[24] Jha, N.I., I. Ghergulescu, A.-N. Moldovan (2019), OULAD MOOC Dropout and Result Prediction using Ensemble, Deep Learning and Regression Techniques, *CSEDU (2)*, pp. 154-164.
[25] Whitehill, J., K. Mohan, D. Seaton, Y. Rosen, D. Tingley (2017), Delving Deeper into MOOC Student Dropout Prediction, *arXiv e-prints*, Vol., No., pp. arXiv: 1702.06404.
[26] Basnet, R.B., C. Johnson, T. Doleck (2022), Dropout prediction in Moocs using deep learning and machine learning, *Education and Information Technologies*, Vol. 27, No. 8, pp. 11499-11513.
[27] Charitaki, G., G. Andreou, A. Alevriadou, S.-G. Soulis (2024), A nonlinear state space model predicting dropout: the case of special education students in the Hellenic Open University, *Education and Information Technologies*, Vol. 29, No. 5, pp. 5331-5348.
[28] Dey, R., F.M. Salem (2017), Gate-variants of gated recurrent unit (GRU) neural networks, *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pp. 1597-1600.

DỰ ĐOÁN TÌNH TRẠNG BỎ HỌC CỦA SINH VIÊN TRONG CÁC KHÓA HỌC TRỰC TUYẾN DỰA TRÊN HỌC TƯƠNG PHẢN CÓ GIÁM SÁT

Đoàn Văn Thanh Phong, Lê Khắc Toản, Lê Văn Vinh, Daniel Ambach, Trần Văn Lăng

TÓM TẮT — Học trực tuyến mang lại nhiều lợi ích và ngày càng trở nên phổ biến trong những năm gần đây. Tuy nhiên, tỷ lệ bỏ học trong các môi trường học tập này rất cao và trở thành một vấn đề lớn đối với các cơ sở giáo dục. Việc xác định sớm những sinh viên có nguy cơ bỏ học giúp giảng viên hoặc tổ chức đào tạo đưa ra các biện pháp hỗ trợ phù hợp và tăng cường mức độ tham gia của người học. Một số phương pháp hiện nay cho bài toán dự đoán việc bỏ học của sinh viên dựa trên các kỹ thuật học máy truyền thống hoặc học sâu. Tuy nhiên, tính riêng biệt của dữ liệu từ các khóa học trực tuyến đặt ra nhiều thách thức đáng kể trong việc dự đoán nguy cơ bỏ học. Nghiên cứu này đề xuất một phương pháp dựa trên học sâu, đặt tên là DP-SCL, cho bài toán dự đoán sinh viên bỏ học. Phương pháp đề xuất áp dụng học tương phản có giám sát, trong đó một bộ mã hóa

dùng chung với các lớp Long Short-Term Memory và cơ chế Multi-Head Attention được huấn luyện bằng hàm mất mát tương phản có giám sát. Kết quả thực nghiệm trên hai bộ dữ liệu từ các nền tảng giáo dục khác nhau cho thấy điểm mạnh của mô hình đề xuất khi so sánh với mười hai phương pháp cơ sở. Mã nguồn của DP-SCL có thể được tải trực tiếp từ trang GitHub của chúng tôi: https://github.com/dvt-phong/DP-SCL.

Từ khóa — MOOC, dự đoán bỏ học của sinh viên, học tương phản có giám sát.

Doan Van Thanh Phong graduated with a Bachelor's degree in Information Technology in 2018 and a Master's degree in Computer Science in 2023 from Ho Chi Minh City University of Technology and Engineering. He is currently a lecturer at Ho Chi Minh City Open University and does research at Ho Chi Minh City University of Technology and Engineering. His research interests include Artificial Intelligence, Deep learning, and student dropout prediction in online courses.
Email: phong.dvt@ou.edu.vn.
ORCID: https://orcid.org/0009-0009-6085-8424

Le Khac Toan graduated with his Bachelor's degree in Information Technology from the University of Science, Ho Chi Minh City, in 2005, and his Master's degree in Computer Science from Vietnam National University, Ho Chi Minh City, in 2009. Currently, he is working at the Hanoi University of Civil Engineering - Ho Chi Minh Campus. He is also a PhD candidate in Computer Science at the Faculty of Information Technology, Ho Chi Minh City University of Technology and Engineering. His research interests include Artificial Intelligence, Machine learning and Educational data mining.
Email: toanlk26.ncs@hcmute.edu.vn.
ORCID: https://orcid.org/0009-0006-4137-4940

Le Van Vinh received the bachelor's degree in Information Technology, and MSc degree in Computer Science from University of Science (Vietnam National University, Ho Chi Minh City) in 2005 and 2009, respectively. He received the PhD degree in Computer Science from HCMC University of Technology (Vietnam National University, Ho Chi Minh City) in 2017. Currently, he is working at the faculty of Information Technology, HCMC University of Technology and Engineering Vietnam. His research interests include Artificial Intelligence, Deep learning, Bioinformatics, Computer Vision.
Email: vinhlv@hcmute.edu.vn.
ORCID: https://orcid.org/0000-0001-5218-3089

Daniel Ambach received his Bachelor's degree in Business Administration in 2008 and his Master's degree in International Business Administration with a focus on Quantitative Methods and Finance in 2011, both from European-University Viadrina, Frankfurt (Oder), Germany. He received his PhD (Dr. rer. pol.) in Statistics from European-University Viadrina in 2016. Since 2021, he has been a Professor of Data Science & Computational Statistics at the DBU Digital Business University of Applied Sciences, Berlin. In parallel, he works as Head of Data Intelligence at accantec information solutions GmbH. His research interests include time series forecasting, machine learning, generative AI, and the integration of AI in higher education.
Email: daniel.ambach@dbuas.de.
ORCID: https://orcid.org/0009-0007-7932-5244

Tran Van Lang is an Associate Professor of Information Technology. He graduated with a Bachelor's degree in Applied Mathematics from the University of Natural Sciences, Ho Chi Minh City in 1982, and received his PhD in Mathematics and Physics from the same university in 1995. During his career, he has held many important positions, including Director of the Ho Chi Minh City Institute of Information Technology, Deputy Director of the Institute of Applied Mechanics and Informatics under the Vietnam Academy of Science and Technology (VAST), Head of the Faculty of Information Technology at Lac Hong University and Nguyen Tat Thanh University, and Chairman of the Science and Training Council of the Ho Chi Minh City University of Foreign Languages - Information Technology. His research interests include Bioinformatics, Cheminformatics, Parallel and Distributed Computing, Scientific Computing, and Computational Intelligence. He is proficient in several programming languages such as FORTRAN, C/C++, Java, Python, and Perl. He has made significant contributions to the training of IT human resources in Vietnam, especially in the southern region. He has successfully served as the principal supervisor for 7 PhD students who defended their theses in Mathematical Foundations for Informatics and Computer Science. More than 100 graduate students have received their master's degrees in IT under his supervision. He also serves as General Chair of the Program Committee of the National Conference on Fundamental and Applied IT Research (FAIR).More detailed information about him can be found at: https://fair.conf.vn/~lang.