Proposing Big Data Architecture for Addressing Dropout Problem in MOOC Platforms

Anh-Khoa Nguyen, Dinh-Khoa Nguyen, Khoa Tan Vo, Thu Nguyen, Tu-Anh Nguyen-Hoang, Ngoc-Thanh Dinh and Hong-Tri Nguyen

January 10, 2025

Proposing big data architecture for addressing dropout problem in MOOC platforms

Anh-Khoa Nguyen[1,2], Dinh-Khoa Nguyen[1,2], Khoa Tan VO[1,2], Thu Nguyen[1,2],
Tu-Anh Nguyen-Hoang[1,2], Ngoc-Thanh Dinh[3], Hong-Tri Nguyen[4]
[1]Faculty of Information Science and Engineering, University of Information Technology, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
[3]The Industrial University of Ho Chi Minh City, Vietnam
[4]Aalto University, Finland

***Abstract*—Accessing academic knowledge on Massive Open Online Courses (MOOCs) has made learning more convenient with flexible schedules and a vast array of course options. The common weakness of these platforms, however, lies in the difficulty of controlling learners' behaviour. No one can know for certain whether the level of engagement during learning is sufficient for learners to fully grasp the knowledge, or whether learners may fail to complete the courses they have enrolled in, leading to dropout behaviour. In this study, we also applied an artificial intelligence model to predict whether current students can complete the course, enabling quick detection of dropout behaviour and timely preventive measures. Based on the proposed architecture, a real-time monitoring, analysis, and management application system for learner behavior can be developed. This empowers course managers to detect which learners might drop out of which courses, enabling timely alerts to learners for adjusting their study plans or, on a broader scale, restructuring the organization of courses with excessively high dropout rates. To maximize scalability with the increasing volume of MOOC data and applicability across different MOOC platforms, our architecture will be built on the Microsoft Azure Cloud computing service, utilizing modern and renowned big data technologies to perform tasks ranging from streaming data collection, batch data processing, distributed processing of large-scale data, to storing data in any format with the latest data lakehouse storage architecture, and real-time data visualization and anomaly detection through integrated models.**

***Index Terms*—big data architecture, clickstream data, MOOC, dropout prediction**

I. Introduction

In today's rapidly evolving digital era, the continuous development of digital technology has opened up numerous opportunities for learners to access vast knowledge resources of humanity from anywhere in the world at any time in a borderless manner. Massive Open Online Courses (MOOCs) are platforms that provide diverse knowledge in various fields and have become increasingly popular with the trend of online learning, with some well-known MOOC platforms including Coursera, edX, Udacity, XuetangX, and others. The effectiveness and benefits of these platforms for learners are undeniable. However, the major challenges lie in managing and monitoring learner engagement as well as the performance outcomes throughout a course. According to reports, up to 90% of learners cannot complete a course until the end, impacting the overall effectiveness of these platforms [1].

Numerous studies have utilized clickstream data on MOOC platforms to predict dropout behaviour, assisting course managers in identifying the risk of learners dropping out. This type of data includes personal learning log, capturing online interactions such as events like pausing, playing, seeking video content during an online lecture session, and more. However, with the rapid increase in the number of MOOC learners, traditional approaches face challenges related to the sheer volume of data generated by learners, posing issues regarding processing speed and model training and prediction speed. Above all, the crucial task is to detect early signs of potential risks associated with suspicious behaviours leading to dropout, allowing course managers to intervene promptly and issue warnings about the risk of not completing the course to learners. This helps learners adjust their study plans for more effective and successful learning.

Therefore, it is essential to build an early dropout behaviour detection system. We propose constructing a large-scale data system on the Microsoft Azure Cloud computing platform to leverage the inherent potential of cloud computing services. This system aims to collect online learner interaction behaviour data using Kafka and Azure, integrating platforms for data processing, analysis, and visualization like Databricks, a web-based technology platform for working with Spark, providing cluster management capabilities and IPython notebook support. We also suggest applying a modern storage architecture known as Lakehouse to store diverse types of data collected from MOOC platforms. Databricks plays a role in loading and transforming data before feeding it into the prediction model. The prediction results will be stored in the Lakehouse to serve the end user.

The focus of this architecture is not only to address specific problems, such as predicting dropout likelihood to issue early warnings for potential dropouts, but also to have a promising future in extending its capabilities for managing learning tasks and supporting decision-making based on diverse data from MOOC learning platforms. This includes monitoring learners' academic performance, recommending online courses, analyzing the effectiveness of course outcomes, and applying natural language processing techniques to analyze the relationship between learner discussions on learning forums and their course outcomes.

In the following sections of this study, Section II will review various dropout prediction methods primarily based on click-stream data. In Section III, we will delve into our proposed big data architecture tailored specifically for MOOCs, highlighting the pivotal Azure Cloud components that underpin this system. Section IV will explore the dropout issue in MOOCs in greater detail, including previous studies and our approach. Section V will showcase the performance of our method, and Section VI will provide a summary of the key findings and tasks.

II. Related works

With the increasing demand for online learning, the amount of data generated by learners on MOOC platforms is also growing rapidly. As a result, this data is becoming increasingly difficult to control and analyze immediately. Traditional dropout prediction models rely on manually extracted features, often computing general statistics over a weekly time unit, resulting in a small-scale feature set that may not fully leverage the detail in the data for effective predictions. The FSPred framework is introduced, encompassing feature creation, feature selection, and dropout prediction stages [7]. It employs fine-grained feature generation over multiple days to create features. The feature selection is then performed using a unified method, selecting valuable features for prediction using logistic regression. To address the computational cost due to the detailed features, a feature selection method is utilized to rank and choose relevant features. Finally, an advanced forward search method based on logistic regression is employed to select the optimal subset of features for the prediction model.

Other hand, Context-aware Feature Interaction Network (CFIN) model, utilizing context-smoothing techniques, is applied to build a dropout prediction model, considering the high correlation between learners' activities and individual context factors [8]. CFIN employs context-smoothing to reduce noise in feature values based on various contexts, involving feature enhancement, embedding, and feature fusion. It utilizes an attention mechanism to incorporate the relationships between learners and course information into the modeling framework. The CFIN model exhibits high prediction effectiveness on KDDCUP and XuetangX datasets, and its results are deployed in an AI assistant on the XuetangX online learning platform to enhance learner retention. Another approach on combine two model Hybrid Algorithm Combining Decision Tree and Extreme Learning Machine (DT-ELM). The DT-ELM algorithm is proposed to predict whether learners will continue to participate in the next week of a course [9]. To expedite the training process, a hybrid approach combining feature selection with a decision tree (DT) and utilizing an extreme learning machine (ELM) for prediction is introduced. The decision layer uses DT for feature selection based on the maximum information gain ratio, and its results are utilized to enhance the impact of selected features on leaf nodes. The enhancement layer improves the impact of selected features on leaf nodes based on the classification abilities of each chosen feature. The mapping layer maps the DT to ELM, enhancing ELM with fewer connections. Experimental results show that the DT-ELM model outperforms traditional machine learning and deep learning models in terms of accuracy, AUC, and F1-score, with faster training times. Ensemble classifiers are utilized to mitigate the variance of prediction errors. Incorporating ensemble classifiers alongside neural networks can augment accuracy and F1 score without succumbing to overfitting. The integration of these methodologies leads to more precise week-by-week dropout predictions. the initial step involves preprocessing the dataset to generate a correlation matrix spanning thirty days for each learner. This approach facilitates early dropout prediction by the conclusion of the first week. Subsequently, six novel models are introduced, employing ensemble classification techniques in conjunction with Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM). CNN is applied for automated feature extraction, while LSTM takes into account the temporal dynamics of the data, enhancing the early prediction performance [5]. These studies above have not mentioned in a deeply way for preventing ratio dropout percentage yet. Therefore, in this study, we took advantage of the big data architecture model to early detect the risk of dropping out of MOOC course thanks to streaming data processing. As a result, course managers can promptly detect students at high risk of not being able to complete the course and issue warnings to students to reschedule their study plans.

Since its introduction, self-attention (also known as a critical component in Transformer) has been widely applied not only in the field of NLP. In dropout prediction tasks, numerous studies have employed this mechanism, among which [19] utilized self-attention and masked-attention to uncover temporal relationships in data. This was complemented by CNN layers for feature compression and a CRF layer for final predictions.

III. Proposed Architecture

A. Architecture overview

The theoretical framework for this architecture is based on the 4 V's of big data, reflecting the diverse data types in MOOCs platforms: unstructured (images, videos, text, forum discussions), semi-structured (learner logs, system records, grading sheets), and structured (learner profiles, instructor details, course specifications). This variety demonstrates the breadth of data. The rapid increase in learners and courses highlights the high Velocity of data, while the large volume of lecture videos and interaction data underscores the Volume. The analytical value of this data benefits learners, instructors, and researchers.

The proposed data architecture, shown in Fig. 1, uses Microsoft Azure Cloud Service for efficient data collection and processing. It leverages cloud computing advantages like automated workflows, scalability, and cost savings compared to on-premise systems.

B. Platform data analysis

Regarding the experimental data utilized for the proposed architecture in this study, we presently focus on the user

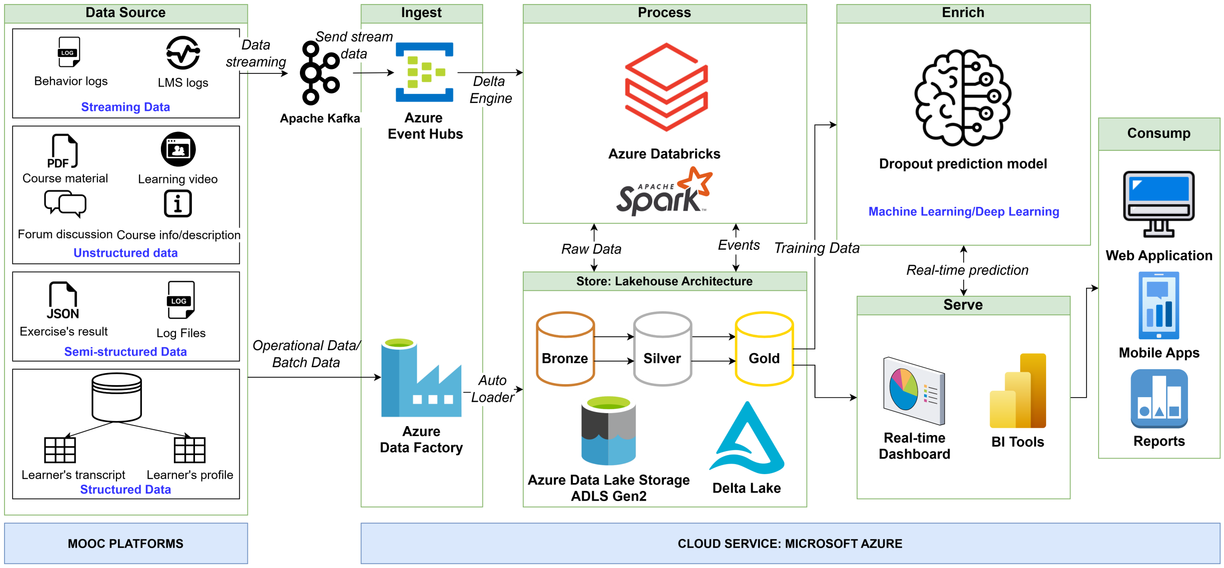

Fig. 1: The proposed architecture for MOOC data.

interaction aspect with course videos, which constitutes log-streaming data collected during learners' course engagement. This dataset concurrently serves as input for the dropout prediction model, which will be one of the deployed application models in the system. In Fig. 2, the visualization depicts the learning trends and dropout behaviors of learners over the course weeks, using user activity data from the XuetangX MOOC platform. The volume of interaction records from learners reaches tens of millions of lines each week, encompassing hundreds of thousands of learners, suggesting potential fluctuations in their engagement patterns over the initial weeks. In the identified dropout group, the volume of interactions with course videos is significantly higher than in the non-dropout group in the first week. However, their interaction volume diminishes considerably in subsequent weeks and lacks stability compared to the non-dropout group. We define the number of dropouts in a given week of the course by examining whether the preceding week was the last week they interacted with the course. Consequently, we discern that although the dropout group exhibits extensive interactions with videos in the initial week of the course, the dropout rate peaks in the subsequent week. This observation could be elucidated by two hypotheses: firstly, individuals inclined to dropout may lack interest and fail to concentrate on the lecture video content, leading to an escalation in interaction behaviors such as fast-forwarding, skipping, or clicking on other course components. Secondly, it is plausible that the instructional content in the lecture videos may not be suitably aligned with their learning abilities, prompting them to pause the video to seek supplementary materials, eventually resulting in discontinuation. This underscores the significance of situating clickstream data within the context of streaming data in addressing the dropout phenomenon in MOOC platforms, as the collection, analysis, and prediction based on this data should be swiftly and timely executed during the early weeks of the course. As analyzed, this period witnesses the most dynamic shifts in learning trends and dropout probabilities.

C. *Data pipeline scenario*

The proposed architecture aims beyond just handling click-stream data, striving to accommodate all types of data across MOOC platforms for future extended research, with each component fulfilling a unique and crucial role. This architecture fosters collaboration and caters to diverse audiences, including analysts, data administrators, data scientists, and course managers. It entails intricate data pipeline scenarios and delineates the roles of these stakeholders within the framework. Streaming data from MOOCs is gathered and relayed via Event Hubs and Kafka to Azure Databricks, leveraging its Delta Engine for efficient data processing. Data Factory pipelines, whether scheduled or triggered, extract raw data from various sources, with Azure Databricks administrators handling data processing upon its arrival. Subsequently, the administrators organize and compress the data into Delta Lake tables or folders within the Bronze layer of Data Lake Storage using Azure Databricks. Azure Databricks jobs, regardless of being streaming, scheduled, or triggered, fetch new transactions from the Bronze layer of Data Lake Storage. These jobs execute data integration, cleansing, transformation, and aggregation before employing ACID transactions to load refined datasets into the Silver and Gold layers of Data Lake Storage. All MOOC data are stored in Delta Lake within Data Lake Storage, ensuring standardized formatting for consistency across services. This architecture revolves around a shared data lake adopting the open Delta Lake format. Raw data from various batch and streaming sources is ingested to construct a unified data platform, catering to downstream applications. Data analysts can visualize, generate reports, and issue notifications to course

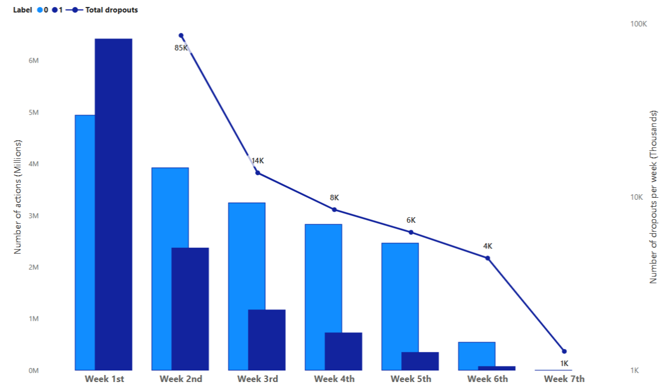

Fig. 2: Weekly analysis of learning interaction trends and student dropout behavior in courses.

managers in case of anomalies, while data scientists leverage this data to develop models addressing specific challenges.

IV. Dropout prediction models

A. Dataset

Regarding the training data for the model in this study, we utilize a dataset from XuetangX, one of the largest MOOC platforms in China. The dataset is derived from the KDD Cup 2015, a competition with the theme of predicting whether learners will drop out of a course based on data from XuetangX. In overview, the KDD Cup 2015 dataset comprises 39 courses, 200,904 course registrations, and a total of 8,157,277 records of learner behaviour on the website. There are 7 different events recorded in the dataset, including *access, discussion, navigate, page_close, problem, video, and wiki*. Registrations marked as dropping out of the course are labeled as 1 (41,681 samples), while the opposite is labeled as 0 (159,223 samples). The dataset appears to be quite imbalanced with dropout labels accounting for up to 80% of the total labels.

The User Activity dataset was released following KDDCup 15 and supplemented with a large amount of log data from XuetangX, comprising nearly 89 million learners behaviours and 685,387 enrollments on the platform. Out of these enrollments, 225,642 are labeled. Similar to KDD Cup 2015, this dataset also suffers from imbalance issues, with dropouts dominating the label ratios. Additionally, the enrollment_id across both datasets is inconsistent, posing a significant challenge in merging the two datasets. Therefore, we have decided to utilize the larger User Activity dataset for training models.

B. Methodology

There are three different definitions in studies predicting dropout behaviour on MOOC platforms [9]:

- Predicting dropout by determining whether learners continue participating in the course until the last week [10], [11], [12].

- Predicting whether the current week is the last active week for learners [13], [14], [15].

- In contrast to the first two definitions that predict the final state of learners and cannot determine whether learners drop out until the end of the course [16], [17]. The third definition is to predict whether learners will continue to participate in the next week of the course, which is related to the continuous state of learners. The dropout label can be determined based on the current week's behaviour, which can help instructors take timely intervention measures.

Predicting user dropout rates in real life doesn't necessarily need to be done in real-time, meaning that predictions can be made after 1 day, 1 week, or 1 month after the course starts. Therefore, the accuracy of the model will be more important than the scale and size of the model.

C. Data preprocessing

An important aspect that has not been addressed by the studies [5], [18] is the handling of imbalanced data before investigating their methods. To accurately assess the performance of each model, one necessary step is to balance the data. Our approach involves downsampling, where enrollments labeled as 1 are reduced to achieve a balanced dataset between the two labels. Since the two datasets differ in the number of behaviour types, with User Activity having 22 behaviours while KDD Cup 2015 only has 7 distinct behaviours, we will remap the behaviours of the User Activity dataset to 7 to facilitate experimentation on the KDD Cup 2015 dataset. Despite having more behaviour types, the distribution of these behaviours varies significantly.

The pre-processing method we use is mostly similar to the approach proposed in [3], with some additional elements, such as incorporating a vector representing the total time for each behaviour. The course duration recorded in the dataset has a maximum of 30 days, and predicting dropout is done at various time intervals. For each registration with an ID Q on day $t \leq T$, there will be two vectors representing the learner's behaviour. The first and second vectors will take the form:

$$x_1 = [a_t^{(1)}, a_t^{(2)}, a_t^{(3)}, \ldots, a_t^{(7)}]$$

$$x_2 = [b_t^{(1)}, b_t^{(2)}, b_t^{(3)}, \ldots, b_t^{(7)}]$$

These vectors will have the same dimension, which is 7, corresponding to the number of behaviour types a and b represent the frequency and total time, respectively, of each behaviour recorded on that day. T is the number of days in weeks; for example, if the number of weeks used for prediction is 3 weeks, then $T = 21$, so $T = \{7, 14, 21, 28, 30\}$.

Aggregating the two vectors according to T will result in two matrices $X_i \in R^{7 \times T}$, Finally, when concatenating these matrices, we obtain a matrix $X \in R^{14 \times T}$

V. Experiments

A. Experimental setup and metrics

We will use a configuration similar to that of [5] with both CNN and LSTM models. For the CNN, a kernel size of 3×3 will be utilized, with 2 convolutional layers corresponding to 8 and 16 filters. Max pooling with a size of 2×2 will be

TABLE I: Comparison table of the performance of models and methods each week

Method/Model	Analysis metric		Selection metric		Analysis metric		Selection metric		Analysis metric		Selection metric	
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
	First week				Second week				Third week			
DT	0.6605	0.6196	0.6605	0.6394	0.6807	0.6431	0.6807	0.6614	0.7015	0.6612	0.7015	0.6808
LR	0.6797	0.6370	**0.8250**	0.7189	0.7131	0.6635	0.8566	0.7478	0.7359	0.6837	**0.8706**	0.7659
MLP	0.6947	0.7133	0.6947	0.7039	0.7281	0.7558	0.7281	0.7417	0.7052	0.7794	0.7052	0.7405
CNN_2v	0.7093	0.6731	0.7826	0.7181	0.7485	0.7084	0.8221	0.7559	0.7592	0.7147	0.8390	0.7670
CNN_1v [5]	0.7151	0.6843	0.7942	0.7290	0.7552	0.7281	0.8106	0.7612	0.7678	0.7309	0.8418	0.7773
Bagging_CNN_CNN_2v	**0.7205**	**0.7451**	0.7074	0.7258	0.7572	**0.7902**	0.7388	0.7636	0.7729	0.8183	0.7480	0.7816
Bagging_CNN_LSTM_1v [5]	0.7166	0.7280	0.7090	0.7183	**0.7577**	0.7873	0.7409	0.7634	**0.7734**	0.8122	0.7514	0.7806
CLSA [18]	0.7077	0.7250	0.6979	0.7112	0.7492	0.7022	0.8598	**0.7726**	0.7631	**0.8523**	0.7212	0.7813
LSTM_1v [5]	0.7178	0.6859	0.7967	**0.7368**	0.7492	0.7086	0.8468	0.7711	0.7708	0.7306	0.8540	0.7869
Self-Attention [19]	0.7169	0.6910	0.7771	0.7316	0.7532	0.7259	0.8080	0.7647	0.7713	0.7400	0.8313	0.7830
LSTM_2v (Our method)	0.7161	0.6856	0.7906	0.7344	0.7508	0.7022	**0.8598**	**0.7726**	0.7717	0.7319	0.8535	**0.7875**

applied after each convolutional layer. Finally, padding will be implemented to maintain the matrix size before and after each convolutional layer unchanged, preventing the loss of information at the matrix edges. Additionally, we will employ a model constructed using the bagging ensemble method, following the structure outlined from [5]. The performance of these models will be compared in section V-B. In the binary classification task, we will use metrics such as accuracy, precision, recall, and F1-score to evaluate the models. Precision measures the model's ability to correctly predict dropouts among those predicted as dropouts. Recall measures the model's ability to not miss any dropout cases in the dataset. F1-Score is a combined measure of precision and recall, providing an overall and comprehensive view of the model. The cloud platform we used for deployment is Azure, with the main components being EventHub, Azure Data Lake Gen 2, and Databricks. EventHub acts as the input for incoming data and gathers streams for distribution to other components. ADLS Gen 2 is used to store all the raw data simulated by local Kafka producer then sent to EventHub as well as the processed data from Databricks. The capture feature of EventHub is designed to send a batch of data every 5 minutes. In Databricks, we use a cluster with one node of the type Standard_DS3_v2, specifically configured with 14 GiB of memory and 4 vCPU cores.

B. Model experiment results

The character "2v" in the model name will learn from the matrix composed of 2 vectors representing learner behaviour introduced in section IV-B. "1v" will represent the model from [5].

The dataset will be divided into three parts: a training set (70%), a validation set (15%), and a test set (15%). Early stopping will be employed, and evaluations will be conducted over a 3-week period.

The classification threshold we use is 0.5. In Table I, it can be observed that after balancing the data, the performance of the models has significantly decreased compared to the results from [5] and [18]. With the integration of multiple sub-models, Ensemble models [5] can yield the most comprehensive results, showcasing the highest Accuracy and Precision scores over three weeks. This approach leverages the strengths of each constituent model to enhance overall performance and robustness in predictions. However, for data with temporal dynamics, LSTM models exhibit good balance in predicting labels, as evidenced by the F1 score.

The Attention method we will experiment with, as per [19], excludes the use of Position Embedding due to findings indicating its ineffectiveness. Experimental results reveal that applying Self-Attention to student behavioral datasets does not yield better outcomes compared to CNN or LSTM models. This can be attributed to the distinct nature of the applied data. Student study days exhibit real-world features that differ significantly from textual passages. MOOC platforms offer courses with varying start and end dates, while each student's study commencement also varies. Student activity is influenced by numerous factors such as work schedules and holidays, resulting in minimal correlations between study days. Moreover, prediction timeframes are limited (up to 21 days), posing challenges for Self-Attention models to grasp the context of entire student behaviors and thereby impacting effectiveness.

We will focus on not missing those who are likely to dropout from the course; hence, the metrics we emphasize to select the most suitable model are Recall and F1. Therefore, the two models we will consider are **LSTM 2v** and Logistic Regression. In practice, we aim for the best efficiency in accurately predicting both dropout and retention labels, so the **LSTM 2v** model is our proposed model due to its stability, even though LR yields very good results in terms of Accuracy.

The Fig. 3 represents the outcome illustrating the data processing of our proposed architecture. Details such as user count, dropout rates, and daily course accesses not only provide reference value for course creators and MOOC platform managers but also serve various purposes such as enhancing learning experiences and platform quality.

VI. Conclusion

This research proposes the construction of a large-scale data application architecture based on state-of-the-art technologies such as cloud, lakehouse architecture, and distributed processing. The main focus is on addressing the dropout issue on MOOCs platforms by thoroughly monitoring learner behaviour and providing timely warnings for those at risk of dropping out. We have proposed the **LSTM 2v** method, which performs well in predicting dropouts by extracting additional data about the time between behaviours and maintaining temporal consistency in clickstream data. The model predicts the number of

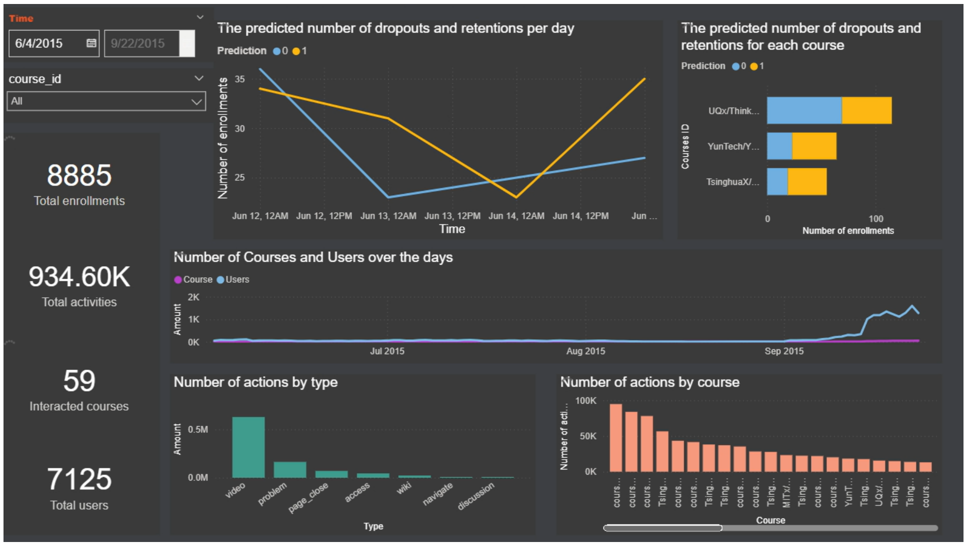

Fig. 3: The dashboard displays the current status of the platform for the day.

users dropping out in the second and third weeks well, with F1 scores of **0.7726** and **0.7875**, respectively. Moreover, it emphasizes scalability, as there is potential for leveraging the system for other applications on MOOC platforms, such as predicting learning performance, learning outcomes, proposing online courses, and even other applications like natural language processing based on learner forum discussions. The feasibility of training a large amount of data, the ability to apply trained models to process a vast amount of comments, and generate real-time results are all relevant aspects. The best-performing model is integrated into Spark Structured Streaming on the Databricks platform, making the most of Databricks' advantages in data processing efficiency, workflow automation, and monitoring and scheduling for job tasks. A large-scale data framework combined with the Lakehouse storage architecture is used to create a system capable of handling massive amounts of continuously collected data from social networks, transmitted through Kafka to Eventhub, to provide real-time hate speech detection results. Subsequently, the results are statistically analyzed and displayed through a real-time dashboard.

Acknowledgment

This research is funded by Vietnam National University HoChiMinh City (VNU-HCM) under grant number DS2024-26-02.

References

[1] Jingjing Zhang and Ming Gao and Jiang Zhang. "The learning behaviours of dropouts in MOOCs: A collective attention network perspective" Computers & Education (2021).
[2] Armbrust, Michael, et al. "Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics." Proceedings of CIDR. Vol. 8. 2021.
[3] Etaati, Leila, and Leila Etaati. "Azure databricks." Machine Learning with Microsoft Technologies: Selecting the Right Architecture and Tools for Your Project (2019): 159-171.
[4] Armbrust, Michael, et al. "Delta lake: high-performance ACID table storage over cloud object stores." Proceedings of the VLDB Endowment 13.12 (2020): 3411-3424.
[5] Talebi, Kowsar, Zeinab Torabi, and Negin Daneshpour. "Ensemble models based on CNN and LSTM for dropout prediction in MOOC." Expert Systems with Applications 235 (2024): 121187.
[6] NVIDIA, NCCL. "NVIDIA Collective Communications Library (NCCL)." (2017).
[7] Qiu, Lin, Yanshen Liu, and Yi Liu. "An integrated framework with feature selection for dropout prediction in massive open online courses." IEEE Access 6 (2018): 71474-71484.
[8] Feng, Wenzheng, Jie Tang, and Tracy Xiao Liu. "Understanding dropouts in MOOCs." Proceedings of the AAAI conference on artificial intelligence. Vol. 33. No. 01. 2019.
[9] Chen, Jing, et al. "MOOC dropout prediction using a hybrid algorithm based on decision tree and extreme learning machine." Mathematical Problems in Engineering 2019 (2019).
[10] Ramesh, Arti, et al. "Learning latent engagement patterns of students in online courses." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 28. No. 1. 2014.
[11] He, Jiazhen, et al. "Identifying at-risk students in massive open online courses." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 29. No. 1. 2015.
[12] Yang, Diyi, et al. "Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses." Proceedings of the 2013 NIPS Data-driven education workshop. Vol. 11. 2013.
[13] Taylor, Colin, Kalyan Veeramachaneni, and Una-May O'Reilly. "Likely to stop? predicting stopout in massive open online courses." arXiv preprint arXiv:1408.3382 (2014).
[14] Kloft, Marius, et al. "Predicting MOOC dropout over weeks using machine learning methods." Proceedings of the EMNLP 2014 workshop on analysis of large scale social interaction in MOOCs. 2014.
[15] Sharkey, Mike and Robert F. Sanders. "A Process for Predicting MOOC Attrition." Conference on Empirical Methods in Natural Language Processing (2014).
[16] Fei, Mi, and Dit-Yan Yeung. "Temporal models for predicting student dropout in massive open online courses." 2015 IEEE international conference on data mining workshop (ICDMW). IEEE, 2015.
[17] Wang, Feng, and Li Chen. "A Nonlinear State Space Model for Identifying At-Risk Students in Open Online Courses." International Educational Data Mining Society (2016).
[18] Qian Fu, Zhanghao Gao, Junyi Zhou, Yafeng Zheng. "CLSA: A novel deep learning model for MOOC dropout prediction" Computers & Electrical Engineering (2021).
[19] Yin, S., Lei, L., Wang, H., & Chen, W. (2020). Power of attention in MOOC dropout prediction. IEEE Access, 8, 202993-203002.