ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Applied Big Data Technique and Deep Learning for Massive Open Online Courses (MOOCs) Recommendation System

Siriporn Sakboonyarat[1] and Panjai Tantatsanawong[2]

ABSTRACT

As traditional recommendation techniques suffer from scalability problems resulting in poor-quality recommendations, they cannot be effectively used on big data. With the immense amount of emerging online learning resources nowadays, it has become harder for users to find and select their preferred content. Similarly, course recommendation systems also face an information overload problem. Most recommendation systems are created based on their own learning management systems and can only be used with those systems. Furthermore, the storage and processing of these systems cannot be updated, which makes them unsuitable for real-world problems, because data is continuously changing and emerging. Focusing on the aforementioned problem, in this study, we propose a novel online recommender system, namely, MCR-C-FGM. It runs on clusters and is trained with a fit-generator method which uses the Apache platform to distribute the processing of large datasets along with a clustering model created by a Deep Neural Network and Long Short-Term Memory. The network is trained with the fit-generator method. The test results with real MOOCs data from Harvard University and MIT, which were published in edX, show a high precision rate of 75%, an accuracy rate of 76%, and a recall rate of 78% in the evaluation processes. The time efficiency during the training process improves by 35% compared to the non-clustering model. Moreover, the MCR-C-FGM is capable of being scaled out, which allows it to efficiently support big data.

Article information:

Keywords: Big Data, Course Recommendation, Deep Learning, MOOCs.

Article history:
Received: October 11, 2021
Revised: December 6, 2021
Accepted: September 17, 2022
Published: October 15, 2022 (Online)

DOI: 10.37936/ecti-cit.2022164.245873

1. INTRODUCTION

Recommendation systems are created to help users select their preferred products. As they recommend a few filtered products among a given assortment, they will narrow down users' choices and consequently decrease the time used [1]. With the overwhelming amount of content nowadays, it is hard for users to manually search for what they want, preventing them from finding contents that suit their preferences [2]. Moreover, due to the COVID-19 pandemic, online learning has become increasingly common. Accordingly, the demand for online learning is predicted to continuously grow as the behavior of learners changes from classroom learning to online learning.

Recommendation systems have been popular applications. Examples include Spotify, Amazon, YouTube, and Netflix. Nowadays, Massive open online courses (MOOCs) have become popular at over 950 universities, with more than 16300 courses available, and around 180 million students who have enrolled in them in 2020 [3]. Due to the large number of choices, students become perplexed when choosing what they should study. In addition, the current course recommendation system still uses traditional recommendation algorithms that require scaling up to increase the data storage, which is not a viable long-term solution [4]. When new data is added, a new training set is required. That is not suitable for real-life situations, where new data is added continuously and rapidly. As a result, new technology and methods are required to handle big data, especially data in MOOCs, which are huge and grow exponentially with time.

This paper proposes a course recommendation sys-

[1,2] The authors are Department of Computing, Faculty of Science, Silpakorn University, Nakhon Pathom, Thailand., E-mail: sakboonyarat_s@silpakorn.edu and tantatsanawong_p@silpakorn.edu
[2]Corresponding author: tantatsanawong_p@silpakorn.edu

tem model for MOOCs which uses the Apache platform to distribute the processing of large datasets along with a clustering model created by a Deep Neural Network and Long Short-Term Memory (LSTM). The network is trained with the fit-generator method.

The research objectives of this study are:

1) To study proper methods for creating a course recommendation system model for MOOCs using big data.
2) To increase the efficiency of the course recommendation system model for big data. For example, efficiency can be improved by decreasing the processing time or increasing precision.

The main contributions of this study are:

1) To propose a big data architecture using Apache Hadoop technology for MOOCs, which supports scaling out.
2) To propose a novel online recommender system model for MOOCs using a Deep Neural Network and LSTM.
3) To highlight the educational benefits to be attained from adoption of our system, With it, learners can find courses specific to their needs in a more streamlined manner and educators can create custom-tailored courses in response to the modern online learning demand for more specific, individualized courses based on the learner's background.

2. LITERATURE REVIEW

2.1 Traditional recommendation algorithms

Traditional recommendation algorithms are usually classified into two types: collaborative recommendation and content-based recommendation [5]. Collaborative recommendation uses the past behaviors of several users to produce new recommendations. The limitation of this type is that a large history of past behavior of users is required to make a good recommendation. Moreover, scalability problems can be encountered because all preferences of all users must be collected, which results in sparse and excessive data. Examples of this algorithm are matrix factorization and the restricted Boltzmann machine.

Content-based recommendation analyzes the property of each item and makes a recommendation based on the user's profile and past behaviors. The limitation of this algorithm is that it cannot recommend new items apart from items related to the user's past behaviors. The examples of this algorithm are naïve Bayes, decision tree, k-nearest neighbor (KNN), and support vector machines. Collaborative recommendation and content-based recommendation both have unique benefits and disadvantages.

2.2 Deep learning for recommender systems

As stated in Section 2.1, traditional recommendation techniques have several limitations. During the past few years, deep learning has been in the spotlight, because of its ability to distinguish complex architecture in high-dimensional data [6]. Deep learning techniques are used as a primary tool in different areas, such as image processing [7], natural language processing [8], signal processing [9, 10], medical image analysis [11], drug discovery and toxicology [12], and financial fraud detection [13]. Deep learning approaches, have been widely adapted and are now used in some of the research on recommendation systems.

[14] proposed a multilayer perceptron (MLP) model used for makeup recommendation. In this work, MLP was used to create model labels and expert rules. This approach demonstrates the efficiency of applying expert knowledge in the learning process of highly accurate MLP framework guidance models, but requires substantial human involvement. [15] used recurrent neural networks (RNNs) to increase the precision of short-term recommendations by changing collaborative filtering methods to sequence prediction problems, and the results show that the efficiency rate is higher than traditional recommender systems and is well suited for high-density data. [16] proposed the usage of deep learning and content-based recommendation to match up user features to item features. Research shows that the model can be scaled to cover a large number of users. [17] proposed a video recommendation system on YouTube using deep learning to assist the recommendation process. This study views the recommendation problem as an extreme multi-class classification problem by viewing each video as a different class and searching for a class which is the most likely to be preferred by users. It uses deep neural networks to train the model and find user and video embeddings to obtain promising results.

2.3 Course recommendation system

Data mining techniques have been used for a long time in recommendation systems. For this reason, [18] proposed a data mining technique using a combination of clustering techniques simple K-means and the association rule algorithm Apriori They were applied for the recommendation of courses in Moodle (modular object-oriented dynamic learning environment) with collaborative filtering. The results show that the model can recommend relevant courses to new students who have recently enrolled.

Similarly, [19] used data mining techniques and Apache Spark to develop a course recommendation system, called, MCRS using the Apriori algorithm to find the connection between former students' information to create the subject recommended for each student in the future. Latest comparison results show that Apriori algorithms on Apache Spark have a better efficiency rate than Apriori algorithms on Apache Hadoop. Nevertheless, this finding is only applicable to a few recommender systems for a few students.

Table 1: *Comparison of course recommendation systems in the literature*

Ref	Purpose	Techniques	Dataset:Environment	Results
[18]	Machine learning algorithms for the recommendation of courses in E-learning systems based on historical data	K-means clustering, Apriori algorithm	N/A : Moodle	The model can recommend courses for new students.
[19]	Course recommendation system for MOOCs	Data mining, Apache Spark, and Apriori algorithm	[22] : MOOCs	The model can recommend a course to some students.
[20]	Resource recommendation model for use in MOOC environments	Deep belief networks (DBNs)	starC MOOC:MOOCs	RMSE $<$ 67.48%
[21]	Recommendation for MOOC with learner neighbors' K-means and learning series	Recommendation on improved features predicted with learner neighbors and learning series	Mic-video Platform of ECNU : MOOCs	Precision = approximately 34% Recall = approximately 37% F1-score = approximately 35%

One limitation of data mining is that it cannot process high-dimensional information or information that has huge volume and grows exponentially with time. [20] used deep learning called deep belief networks (DBNs) to recommend courses in MOOCs, namely, MOOCRC. This method uses learner features, course content attribute features, and learner behavior features to form a "course content attribute features" vector. Using the test from starC MOOC of Central China Normal University, the results show that when comparing MOOCRC to other techniques, such as content-based, KNN, singular value decomposition, and restricted Boltzmann machines, MOOCRC has the least root mean square error (RMSE) with a value of less than 67.48%, which is considered highly efficient.

[21] proposed a recommendation system that focuses on adjusting features predicted using learner neighbors and learning series called RLNLS to reduce the dropout rate of students. The recommendation accuracy of RLNLS can be shown via the decrease in the dropout rate since some of the students' dropout reasons were the fact that the subjects do not match with their interests. The conclusions of the above-mentioned papers' course recommendation systems are presented in Table 1.

2.4 Importance of big data and education

Due to the increasing amount of data, educational works need to use technology as a tool to assist educational processes. Accordingly, big data technology has been used in many areas. For example, [23] presented an analysis of students in MOOCs using computational methods to determine self-regulation learning patterns and discovered that successful learners have more interactions than unsuccessful learners, which helps teachers to keep track of their processes. [24] used data mining to predict dropouts from past behavioral records of learners and found that by applying the gradient boosting decision tree, the prediction reaches an accuracy of 88%, which helps reduce the absence rate. [25] examined the relation between results and participation of learners by analyzing texts and found that participation in Moodle supports learners academically. [26] examined videos with behavioral data from the analysis of big data in an online learning environment and found that using imVideo, a tool to evaluate learners, helps to significantly increase the interaction among learners and between learners and peers.

3. METHODOLOGY

3.1 Analysis

Collaborative recommendation and content-based recommendation have their own restrictions. Accordingly, this study determines what hybrid method maximizes the advantages and minimizes the drawbacks of both methods [27]. As the event logs in MOOCs are large, fluctuating, and from different sources, it is necessary to have a fitting method to store and process data. Here, the Apache platform, which is designed to store large amounts of data by saving it in a Hadoop Distributed File System (HDFS), is used. The Apache platform is universally recognized and used by many applications, such as Facebook, Yahoo, Twitter, Adobe, machine translation, bioinformatics, scientific computing, image processing, and government [28]. This platform is also used with TensorFlow, Keras, Python, and GeForce RTX 2080 Ti graphics cards to improve system performance. In this study, we developed a Hadoop cluster in four virtual machines as a platform to test the system. Among the four machines, one of them is a master node while the rest are slave nodes. We used Intel(R) Xeon(R) E5-2670 v2 @2.5GHz CPUs as processing units. The details are presented in Table 2.

Table 2: *Configuration of a Hadoop cluster*

Node	CPU	Memory	Disk
Master	Intel ®Xeon ®E5-2650 v2 2.60 GHz	16 GB	150 GB
Slave1–Slave3	CPU 2 cores	8 GB	100 GB

The event log used in the course recommendation system on MOOCs is sequential. Therefore, we used a Deep Neural Network with LSTM to make it suitable for data processing. The summary of the technology and procedure for the course recommendation model is shown in Table 3.

Table 3: *Technology and procedure used in the course recommendation system model*

Objective	Technology	Reason
Data storage	Apache Hadoop	Replication, fault tolerance, scalability, data integrity, data locality, and high throughput [29, 30]
Development tool	TensorFlow, Keras, and Python	1) TensorFlow and Keras can efficiently run a model. 2) Python is widely used for machine learning.
Processing unit	GeForce RTX 2080 Ti graphics card	Used to accelerate runtime
Data processing procedure	Deep Neural Network with LSTM	1) A Deep Neural Network can find a hidden pattern in high-dimensional data. 2) LSTM is suitable for processing sequential data [31-33].

3.2 Dataset

This research used a dataset from Harvard University and MIT which was published the edX learning data in 2012–2013 [22] to train and test the model. The dataset is separated into a total of 13 subjects, for a total of 641138 records.

3.2.1 Exploratory Data Analysis (EDA)

Exploratory Data Analysis refers to the critical process of performing initial investigations on data to discover patterns and to spot anomalies. It is a good way to understand the data first and try to gather as many insights as possible from it. EDA is all about making sense of the data in hand, before getting dirty with it.

Table 4: *Correlation results of variables*

	LR	YB	GN	LE	CI	IS	CT	Gr	Vi	EP	NC	ST
LR	1.000	-0.073	-0.078	-0.002	0.117	-0.159	-0.030	-0.042	-0.079	-0.044	-0.069	-0.006
YB	-0.073	1.000	-0.607	-0.419	0.035	-0.047	-0.028	-0.033	-0.033	-0.040	-0.063	-0.017
GN	-0.078	-0.607	1.000	0.379	-0.161	0.182	0.015	0.017	0.068	0.037	0.061	0.054
LE	-0.002	-0.419	0.379	1.000	-0.055	0.069	0.022	0.024	0.040	0.027	0.047	0.021
CI	0.117	0.035	-0.161	-0.055	1.000	-0.907	-0.053	-0.059	-0.106	-0.012	-0.022	-0.030
IS	-0.159	-0.047	0.182	0.069	-0.907	1.000	0.053	0.064	0.112	0.004	0.046	0.029
CT	-0.030	-0.028	0.015	0.022	-0.053	0.053	1.000	0.936	0.131	0.628	0.628	-0.006
Gr	-0.042	-0.033	0.017	0.024	-0.059	0.064	0.936	1.000	0.166	0.691	0.695	-0.007
Vi	-0.079	-0.033	0.068	0.040	-0.106	0.112	0.131	0.166	1.000	0.199	0.426	-0.087
EP	-0.044	-0.040	0.037	0.027	-0.012	0.004	0.628	0.691	0.199	1.000	0.790	-0.010
NC	-0.069	-0.063	0.061	0.047	-0.022	0.046	0.628	0.695	0.426	0.790	1.000	-0.017
ST	-0.006	-0.017	0.054	0.021	-0.030	0.029	-0.006	-0.007	-0.087	-0.010	-0.017	1.000

As shown in Table 4, the Final_cc_cname (LR), LoE (LE), YoB (YB) and Gender (GN) attributes generally exhibit negative correlations (shown in red) which can be used to represent the User profile data group. On the other hand the course_id (CI), Institution (IS), viewed (Vi), explored (EP), certified (CT), grade (Gr), start_time (ST), nchapters (NC) attributes generally exhibit positive correlations (shown in green) which can be used to represent the Course learning data group. As a result, we have selected 4 User profile attributes and 8 Course learning attributes (12 attributes in total) to represent the data.

3.2.2 Data Preparation

The data preparation steps are:

1) Input: 4 User profile attributes and 8 Course Learning attributes as per Tables 5 and 6.
2) Cleaning Data: The YoB, grade, nevents, ndays_act, nplay_video, nchapters, and nforum_posts attributes will default to "0" if the data value is missing. The missing gender attribute values will default to "3" to represent "gender not specified" and missing LoE attribute values will default to "6" to represent "level of education not specified". The number of data records after cleaning remains the same as prior to cleaning.
3) Set Course_id attribute as the Class Label for the model.
4) Perform Min-Max Normalization so that data values are in the range $(0 - 1)$ as shown in Equation 1.
5) As per common practice in large datasets, the dataset is divided into 80% for Training (512910 records) and 20% for Testing (128228 records).
6) Upon further observation, the course enrollment count per subject differs by a large amount across each subject. Therefore, we have applied Data Augmentation by adjusting the number of records for each subject to equal to the course with the most frequent enrollment records and each added record is randomized. After augmentation, the record count per subject becomes 135697 records for Training set and 33924 for the Test set.
7) After Data Augmentation, there are a total of 2072799 records for the Training set (80%) and 542784 records for the Test set (20%).

$$v' = \frac{v - min_A}{max_A - min_A} N \tag{1}$$
$$N = (new\_max_A - new\_min_A) + new\_min_A$$

v is the original value. v' is the normalized value. min_A and max_A are the minimum and maximum values from the original dataset. new_min_A and new_max_A are the desired new minimum and maximum values for the normalized dataset.

3.3 System architecture diagram

We present an overview of our proposed course recommendation system architecture containing data sources, data ingestion, processing, and insight, in Fig.1.

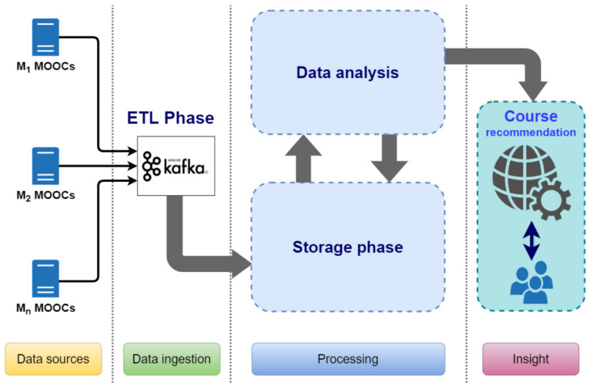

Fig.1: *Overview of the course recommendation system architecture*

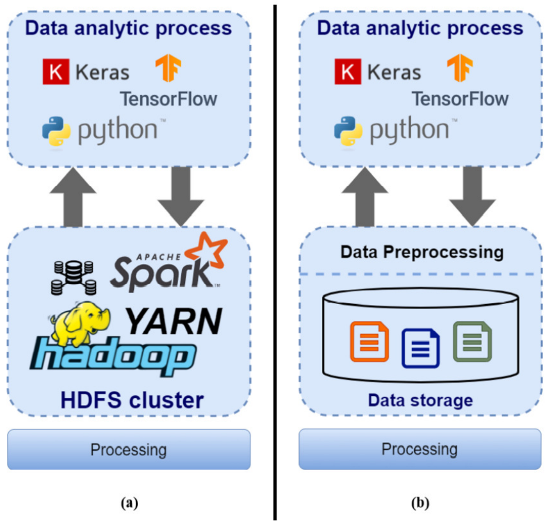

Fig.2: *"Processing" working process according to the overview of the course recommendation system architecture shown in Fig.1. This consists of (a) Clustering and (b) Non-Clustering so that we can compare the efficiency of the models*

3.3.1 Data sources

Data from the MOOCs sources are named M1, M2,..., Mn. The Architecture of the course recommendation system supports event logs from multiple MOOCs with various data structure formats, including CSV, JSON, and HDFS.

3.3.2 Data ingestion

The data is transferred from source MOOCs to Hadoop clusters using Apache Kafka to extract, load, and transform it so that big data can be transmitted continuously and with high quality.

3.3.3 Processing

As shown in the architecture of the course recommendation system shown in Fig.1, we separate the experiment into two working processes: clustering (Fig.2(a)) and non-clustering (Fig.2(b)). Accordingly, the time efficiency and accuracy of each model can be compared. As shown in Section 3.3.2, the data ingestion process receives data from MOOCs and sends it to the next stage. In Fig.2(a), Apache Kafka transfers the data over to the Hadoop cluster, which consists of connectors called Hadoop HDFS, Hadoop YARN, and Apache Spark. In Fig.2(b), Apache Kafka transfers the data to data storage, which collects the data in form of CSV files.

3.3.4 Insight

The course recommendation system model will suggest courses to users.

3.4 Proposed model

As shown in to the overview of the course recommendation system architecture shown in Fig.1, a Deep Neural Network along with LSTM is used to recommend courses to users. The model is shown in Fig.3.

Input consists of four attributes of user profile and eight attributes of course learning, as shown in Tables 5 and 6, respectively.

Table 5: *User profile attribute description*

No.	Attribute name	Description
1	Local Register	Contains the location of the student
2	Year of Birth	Year of birth
3	Gender	Possible values are male (m), female (f), and others (o)
4	Level of Education	Highest level of education completed

Table 6: *Course learning attribute description*

No.	Attribute name	Description
1	Course ID	Course name
2	Institution	Name of the institution that owns the course
3	Certified	The value is 1 if the student has got the certificate, otherwise 0
4	Grade	Grade of a student in the course
5	Viewed	The value is 1 if the student has accessed the course tab (videos, exams, and problem sets), 0 otherwise
6	Explored of Education	The value is 1 if the student has explored at least half of the chapters of course where they registered, 0 otherwise
7	N chapters	Number of chapters (within the courseware) with which the student interacted
8	Start time	Timestamp of course registration

As shown in Fig.3, the course recommendation system utilizes a Deep Neural Network in conjunction with LSTM where the input data consists of User profile data and Course learning data. The layer implementation details are as follows:

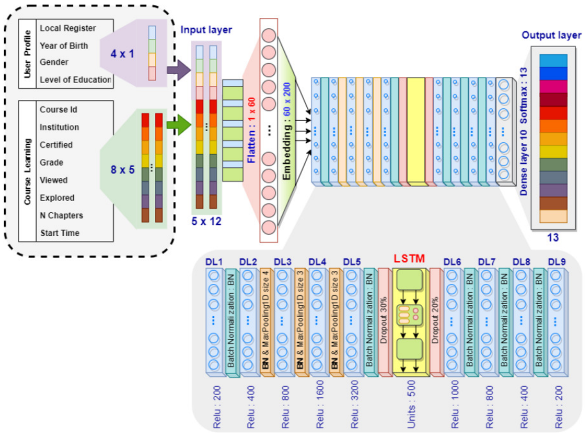

Fig.3: *Course recommendation model created using a Deep Neural Network and LSTM*

1) Input Layer consists of 4 User profile attributes and 8 Course learning attributes, (12 attributes in total)
2) 10 Dense Layer, DL1 - DL10
3) Output Layer produces 13 classes
4) Nadam Optimization Algorithm is used to adjust weights and biases of node connectivity to improve regression towards the desired output
5) Batch Normalization is additionally used in a Hidden layer to improve efficiency and stabilize the training process
6) Hyper parameters: Optimizer = Nadam, loss function = SparseCategoricalCrossentropy, learning rate = 1e-6 and epsilon = 1e-17
7) Activation Function: ReLU for Dense Layers and Softmax for the final layer

Dense layer 1 – 10 details :

- Dense layer 1 ReLU 200
- Batch normalization
- Dense layer 2 ReLU 400
- Batch normalization and Max Pooling 1D size 4
- Dense layer 3 ReLU 800
- Batch normalization and Max Pooling 1D size 3
- Dense layer 4 ReLU 1600
- Batch normalization and Max Pooling 1D size 3
- Dense layer 5 ReLU 3200
- Batch normalization
- Dropout 30
- LSTM unit = 500
- Dropout 20
- Dense layer 6 ReLU 1000
- Batch normalization
- Dense layer 7 ReLU 800
- Batch normalization
- Dense layer 8 ReLU 400
- Batch normalization
- Dense layer 9 ReLU 200
- Batch normalization
- Dense layer 10 Softmax

The model training process uses TensorFlow, Keras, Python, and GeForce RTX 2080 Ti graphics cards. The experiments can be separated into three methods: 1) Non-Clustering, which is trained using the fit method called MCR-NC-FM shown in Fig.4(a); 2) Non-Clustering, which is trained using the fit-generator method called MCR-NC-FGM shown in Fig.4(b); 3) Clustering, which is trained using the fit-generator method called MCR-C-FGM shown in Fig.4(c). We will compare the time efficiency and precision these three models.

3.4.1 Non-Clustering trained using the fit method called MCR-NC-FM

As shown in Fig.4(a), the MCR-NC-FM model trained using a fit method starts with "data preprocessing," where it receives and prepares training data from the data storage. Then the training data is sent to "RAM." "Model building" activates the "fit method" to start the work and repeats (3) until it reaches the training amount (total of 500 epochs). The "fit method" always uses the same set of training data from "RAM" to train the model.

3.4.2 Non-Clustering trained using the fit-generator method called MCR-NC-FGM

As shown in Fig.4(b), MCR-NC-FGM is trained using a fit-generator method. First, "model building" uses the "generator data module" to prepare training datasets as the "generator data module" will receive data from "data storage" and transfer it to "RAM." Then, "model building" activates the "fit-generator method" to start the training. Processes 1–4 are be repeated until it reaches the training amount (total of 500 epochs). The "fit-generator method" shuffles the training data order so that it is different in every training epoch.

3.4.3 Clustering trained using a fit-generator method called MCR-C-FGM

As shown in Fig.4(c), MCR-C-FGM is trained using a fit-generator method. First, "model building" uses the "generator data module" to prepare training datasets as the "generator data module" will receive data from the "Hadoop cluster" and transfer it to "RAM." Then, "model building" activates the "fit-generator method" to start the training. Processes 1–4 are repeated until it reaches the training amount (total of 500 epochs). The "fit-generator method" shuffles the training data order so that it is different in every training epoch.

3.5 Performance evaluation of the model

Model precision is measured and evaluated using the testing set data by determining the precision from evaluating the predicted results against the actual re-

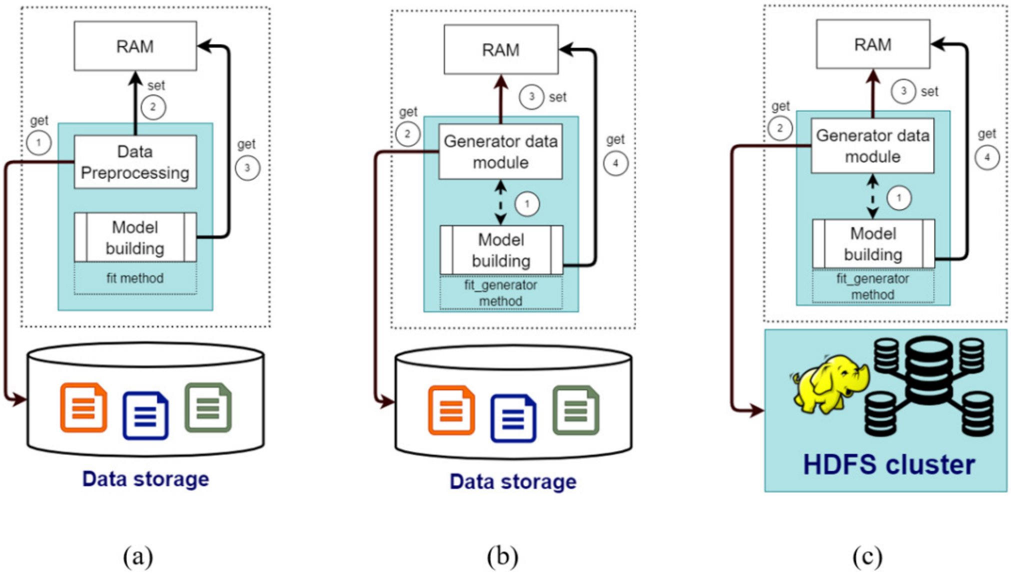

Fig.4: *Course recommendation system process separated into three methods: a) MCR-NC-FM, b) MCR-NC-FGM, c) and MCR-C-FGM*

sults, as shown in Equation 2.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Accuracy is a metric that generally describes how the model performs across all classes. It is calculated as the ratio of the number of correct predictions to the total number of predictions, as shown in Equation 3.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (3)$$

The recall is calculated as the ratio of the number of Positive samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect Positive samples, as shown in Equation 4.

$$Recall = \frac{TP}{(TP + FN)} \quad (4)$$

True Positive (TP) represents the number of results that correctly indicate the presence of a condition or characteristic.

True Negative (TN) represents the number of results that correctly indicate the absence of a condition or characteristic.

False Positive (FP) represents the number of results which wrongly indicate that a particular condition or attribute is present.

False Negative (FN) represents the number of results which wrongly indicates that a particular condition or attribute is absent.

4. EXPERIMENT AND RESULTS

4.1 Efficiency of the course recommendation system architecture model

Based on the architecture of the course recommendation system shown in Fig.1, developers separate the experiment into two methods: clustering (Fig.2(a)) and non-clustering (Fig.2(b)). The architecture of the course recommendation system with the clustering function is suitable and has good performance. This is due to the Apache platform's ability to handle big data. When the architecture is combined with a model developed with a Deep Neural Network and LSTM, the performance of the course recommendation system model improves in terms of time and values. It has a better efficiency. The course recommendation system model using deep learning is shown in Fig.5.

4.2 Efficiency of the course recommendation system

For the testing processes of MCR-NC-FM, MCR-NC-FGM, and MCR-C-FGM on epochs 301–350, the precision of the model hardly changes despite the training process precision, which still improves, as shown in Fig.6, 7, and 8. This is the reason why we decided to use epochs 301–350 to compare the precision and time efficiency of the model, as shown in Tables 7 and 8.

4.2.1 Time efficiency and precision of the MCR-NC-FM model

The working process of the MCR-NC-FM model is shown in Fig.4(a). The "model building" process using the "fit method" always uses the same data

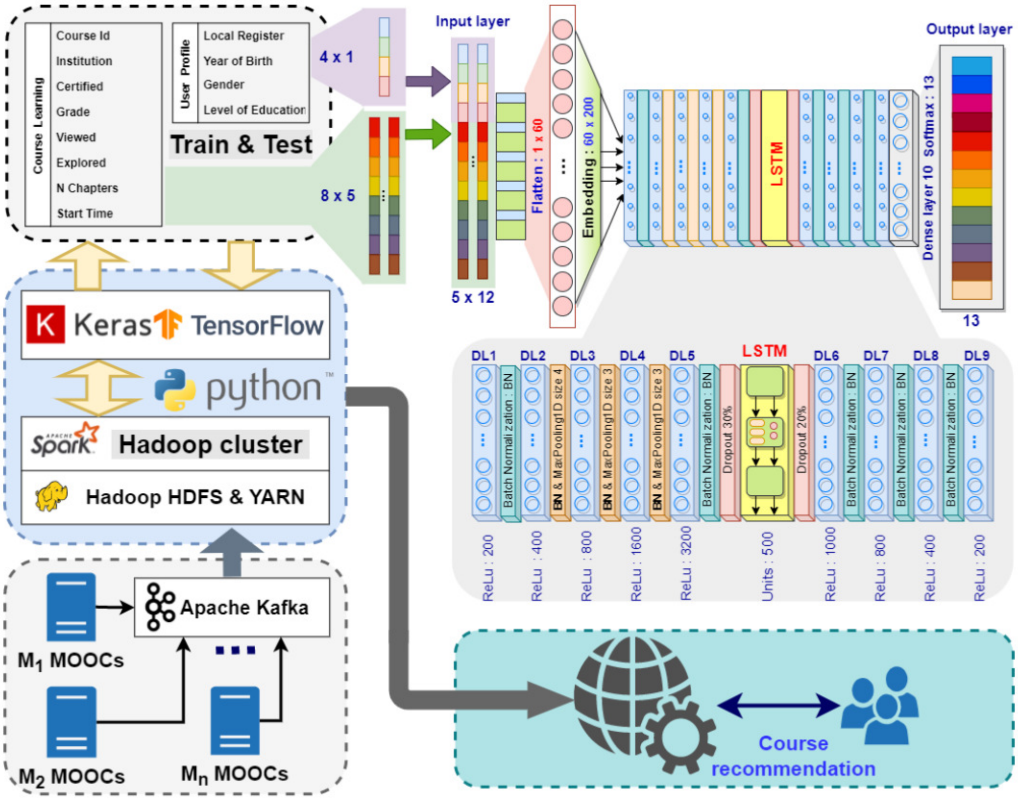

Fig.5: *System architecture for course recommendation system in MOOCs*

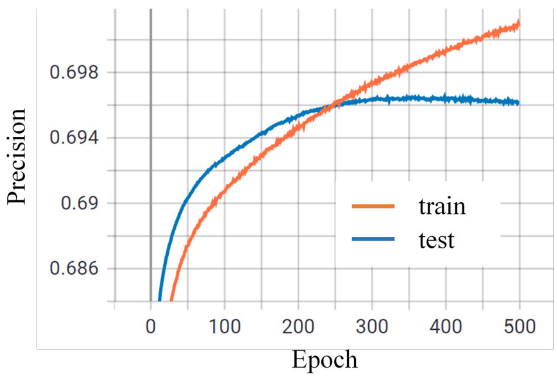

Fig.6: *Time efficiency and precision of the MCR-NC-FM model*

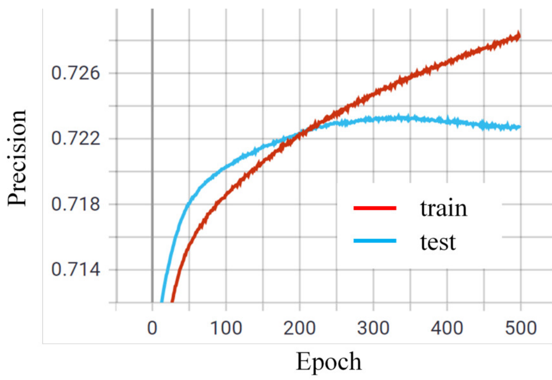

Fig.7: *Time efficiency and precision of the MCR-NC-FGM model*

from "RAM" to train the model (total of 500 epochs) by repeating the third process until it reaches the number of training rounds. This process results in low precision, as shown in Fig.6. The precision of the training dataset (blue line) is higher than precision of the testing dataset (orange line) from the start until around 250 epochs. After 250 epochs, the precision is more constant.

4.2.2 Time efficiency and precision of the MCR-NC-FGM model

The working process of the MCR-NC-FGM model is shown in Fig.4(b). First, "model building" contacts the "generator data module" to prepare training datasets. Then, the "generator data module" receives data from the "data storage" and stores it in "RAM." "Model building" activates the "fit-generator method" and processes 1–4 are repeated until it reaches the training amount (total of 500 epochs). The "fit-generator method" shuffles data to make it different in every training session. The collection of data in the Hadoop cluster results in better management of data within the cluster and enables horizontal scaling without affecting the system. As shown in Fig.7, the precision of the model testing procedure (red line) is higher than that of the model

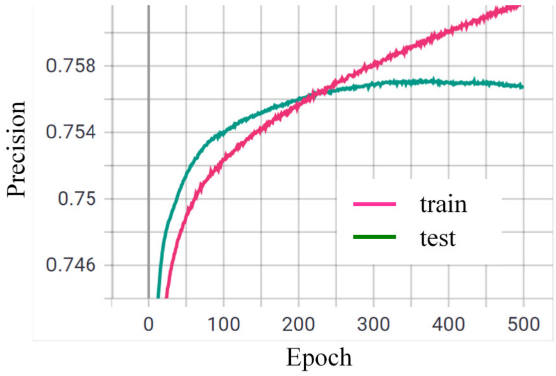

Fig.8: *Time efficiency and precision of the MCR-C-FGM model*

training process (blue line) in the early testing. However, From 200 epochs onward, the precision in the model testing phase is stable. Hence, the MCR-NC-FGM model has better precision in the testing and training phases than the MCR-NC-FM model.

4.2.3 Time efficiency and precision of the MCR-C-FGM model

The working process of the MCR-C-FGM model is shown in Fig.4(c). First, "model building" contacts the "generator data module" to prepare training datasets while the "generator data module" receives data from the "Hadoop cluster" and stores it in "RAM." "Model building" activates the "fit-generator method" as processes 1–4 until it reaches the training amount (total of 500 epochs). The "fit-generator method" shuffles data to make it different in every training session. The collection of data in the Hadoop cluster results in a better management of data within the cluster and enables horizontal scaling without affecting the system. As shown in Fig.8, the precision of the model testing procedure (pink line) is higher than that of the model training process (green line) in the early testing cycles. However, From 200 epochs onward, the precision in the model testing phase is stable. Hence, the MCR-C-FGM model has better precision in the testing and training phases than the MCR-NC-FM and MCR-NC-FGM models.

As shown in Table 7, when considering the precision of the model, during epochs 301–350, the precision of the MCR-NC-FM, MCR-NC-FGM, and MCR-C-FGM models are 69.785, 72.526, and 75.861, respectively. The MCR-C-FGM model has the best precision rate of 75.861 as MCR-C-FGM is trained with a fit-generator method in which training data is shuffled for every test using the "Hadoop cluster." This is the reason why the efficiency of the MCR-C-FGM model is higher than that of the MCR-NC-FM and MCR-NC-FGM models.

As shown in Table 8, when considering the time efficiency of the model, in epochs 301–350, the MCR-NC-FM, MCR-NC-FGM, and MCR-C-FGM models have total training times of 18:57:24 hours, 17:49:54 hours, and 11:33:12 hours, respectively.

The MCR-C-FGM model, which has the best time efficiency during the testing process, is trained with the least time during the training process. The time efficiency comparison between the MCR-C-FGM, the MCR-NC-FM, and the MCR-NC-FGM models shows that the MCR-C-FGM model requires 444 fewer minutes than the MCR-NC-FM model and 376 fewer minutes than the MCR-NC-FGM model.

MOOCs presents a technical challenge in Big Data space in terms of both the volume and variety of data. Furthermore, they require processing to be done at high velocity. One Example scenario is collecting and processing real-time video player data (play, pause, stop). Traditional data collection and processing methods are not well suited for analyzing all the information efficiently due to the high rate at which data accumulates, which leads to system bottlenecks. Selecting the range of tools and methodologies to use for such a scenario requires extensive knowledge of and experience with the tools and methodologies involved to completely grasp the advantages and the disadvantages (also known as engineering tradeoffs once compared between competing methodologies) that will be introduced to any given system with specific requirements, workload, and needs.

Throughout this study of designing and building a course recommendation for MOOCs using Big Data, the authors have been introduced to several techniques with their own tradeoffs and have directed the study's focus to three main areas: 1) Association rules, 2) Deep Neural Networks, and 3) Deep Neural Networks with LSTM. Of no less importance, designing an architecture for processing large amount of data requires appropriate technology that scales well in response to the influx of data. This is further elaborated in Table 10.

The authors chose the Apache Platform, which is a software framework for handling large datasets by leveraging clustering techniques for distributed processing. Information flowing from MOOCs are stored in blocks within HDFS such that pieces of information can be shard across different computers (or "nodes") and appropriate distributed algorithms are used to guarantee that the distribution happens correctly and handle data errors that occur. Based on the proposed architecture, new nodes can also be added in such a way that the operations of existing nodes are not effected (scale out pattern). Apache Kafka is also used as a distributed streaming platform and Message Broker that is able to handle continuous, large streams of data from multiple sources.

The MCR-C-FGM model uses Apache Kafka technologies to send and retrieve real-time data, which allows the system to be stable. It can also be connected by structured and unstructured data, which differs from MCRS [19]. MCRS uses Apache Sqoop to con-

Table 7: *Training precision and testing precision*

Epoch	Training precision			Testing precision		
	MCR-NC-FM	MRC-NC-FGM	MRC-C-FGM	MCR-NC-FM	MRC-NC-FGM	MRC-C-FGM
001–050	67.969 ± 1.172	70.712 ± 1.427	73.566 ± 2.952	68.549 ± 0.654	71.236 ± 0.877	74.310 ± 1.487
051–100	68.935 ± 0.095	71.724 ± 0.089	75.091 ± 0.095	69.176 ± 0.070	71.937 ± 0.063	75.303 ± 0.074
101–150	69.190 ± 0.063	71.964 ± 0.058	75.334 ± 0.056	69.355 ± 0.043	72.093 ± 0.035	75.466 ± 0.034
151–200	69.381 ± 0.049	72.145 ± 0.047	75.496 ± 0.042	69.485 ± 0.032	72.191 ± 0.024	75.561 ± 0.023
201–250	69.539 ± 0.043	72.291 ± 0.040	75.632 ± 0.037	69.571 ± 0.018	72.269 ± 0.019	75.631 ± 0.015
251–300	69.672 ± 0.037	72.415 ± 0.035	75.754 ± 0.034	69.619 ± 0.011	72.305 ± 0.010	75.671 ± 0.012
301–350	69.785 ± 0.031	72.526 ± 0.031	75.861 ± 0.032	69.638 ± 0.005	72.324 ± 0.006	75.698 ± 0.006
351–400	69.885 ± 0.028	72.622 ± 0.025	75.962 ± 0.029	69.640 ± 0.005	72.315 ± 0.008	75.704 ± 0.006
401–450	69.976 ± 0.024	72.711 ± 0.025	76.052 ± 0.026	69.634 ± 0.006	72.295 ± 0.009	75.697 ± 0.005
451–500	70.056 ± 0.021	72.791 ± 0.024	76.140 ± 0.024	69.622 ± 0.006	72.274 ± 0.008	75.687 ± 0.008

Table 8: *Time efficiency in the training process*

Epoch	Training precision			Training time (hh:mm:ss)		
	MCR-NC-FM	MRC-NC-FGM	MRC-C-FGM	MCR-NC-FM	MRC-NC-FGM	MRC-C-FGM
001–050	67.969 ± 1.172	70.712 ± 1.427	73.566 ± 2.952	02:41:03	02:33:36	01:38:00
051–100	68.935 ± 0.095	71.724 ± 0.089	75.091 ± 0.095	05:24:48	05:06:44	03:16:20
101–150	69.190 ± 0.063	71.964 ± 0.058	75.334 ± 0.056	08:08:26	07:38:15	04:55:37
151–200	69.381 ± 0.049	72.145 ± 0.047	75.496 ± 0.042	10:50:38	10:09:48	06:34:51
201–250	69.539 ± 0.043	72.291 ± 0.040	75.632 ± 0.037	13:31:39	12:44:21	08:14:30
251–300	69.672 ± 0.037	72.415 ± 0.035	75.754 ± 0.034	16:14:19	15:17:34	09:54:13
301–350	69.785 ± 0.031	72.526 ± 0.031	75.861 ± 0.032	18:57:24	17:49:54	11:33:12
351–400	69.885 ± 0.028	72.622 ± 0.025	75.962 ± 0.029	21:40:26	20:21:31	13:11:49
401–450	69.976 ± 0.024	72.711 ± 0.025	76.052 ± 0.026	00:21:27	22:54:02	14:50:21
451–500	70.056 ± 0.021	72.791 ± 0.024	76.140 ± 0.024	02:59:24	01:24:43	16:27:30

Table 9: *MCR-C-FGM performance results*

Model/Evaluation	Precision	Accuracy	Recall
MCR-C-FGM	75.70%	76.00%	78.85%

nect the learning management system and HDFS, and that has the limitation that the data should only be structured with batch processing. Even if MOOCRC [20] first uses deep learning techniques called DBNs for MOOC recommendation systems, it uses as many as 40 attributes of learner and course content features to make course characteristic vectors. Some of the features do not help with the precision. By contrast, MCR-C-FGM uses only 12 attribute features but manages to heighten the precision to more than 75%. Accuracy is 76% and Recall is more than 78% in Table 9.

4.2.4 Applications of the proposed system architecture and proposed recommendation system

1) The proposed architecture and recommendation system can be used within learning platforms such as Open EdX platform, Moodle, or other learning platform with similar collection of data attributes as suggested in the study, which includes recommendation results for up to 13 subjects. Our model proposed can be used directly without altering the input layer or the output layer.

2) Recommendation systems for other classes of results such as hotel recommendations, movie recommendation, and tourist spot recommendation can leverage similar techniques as present in this study. User Profile data will be present in these other systems as well, and other attributes related to the field can be used in place of course learning attributes. Then the number of output classes can be adjusted accordingly without having to modify the deep layers.

5. CONCLUSIONS AND FUTURE WORKS

In this study, the developers propose an architecture to collect big data in MOOCs environments using the ability of the Apache platform along with a course recommendation system model for MOOCs. Moreover, Apache Kafka is used for real-time data transfers. The clustering models are created using a Deep Neural Network and LSTM trained with the fit-generator method. The experimental results show that the MCR-C-FGM model has the best time efficiency and precision compared to the MCR-NC-FM and MCR-NC-FGM models. Using the real data retrieved from Harvard University and MIT, the precision is as high as 75%, with an accuracy rate of 76% and a recall rate of 78% in the evaluation processes.

Table 10: *Comparison of course recommendation system techniques*

Technique	Advantages	Disadvantages
Association rule [19]	- Easy to develop - Simple environment to set up	- Difficult to scale - Does not support very large datasets
DNN [20]	- Supports scaling out - Able to process large datasets	- Data may be overfit as part of training the model - Model limitations in terms of how well past information can be retained within the model. A scenario where the model is biased towards newer data may occur more often.
DNN+LSTM (MCR-C-FGM)	- Supports scaling out - Able to process large datasets	- High initial development cost as the technique requires more processing power

For time efficiency, training the MCR-C-FGM model required 444 fewer minutes than training the MCR-NC-FM model. Training the MCR-C-FGM model required 376 fewer minutes than training the MCR-NC-FGM model. With the ability of LSTM, old data can be used to predict future events as LSTM reuses the results from the previous node and applies them to the next node as an input. Moreover, using the fit-generator method and Hadoop cluster allows the dataset to be shuffled in every training session. This method results in a higher efficiency for the MCR-C-FGM model compared with the other models. Accordingly, the MCR-C-FGM model is a good method for developing recommendation systems. Furthermore, it supports scaling out or increasing the size of the storage space in a horizontal format (scale out) and can handle big data.

In future work, we are determined to improve our method using LSTM and other features, such as Sentiment, and to optimize of the model to enhance the efficiency of the model. We also aim to study our method diligently for the sake of improving the education system.

ACKNOWLEDGMENT

This research was supported by Mahidol Wittayanusorn School, Thailand.

References

[1] A. K. Sahu, and P. Dwivedi, “User profile as a bridge in cross-domain recommender systems for sparsity reduction,” *Applied Intelligence*, vol. 49, no. 7, pp. 2461-2481, 2019/07/01, 2019.

[2] E. Ashraf, S. Manickam, and S. Karuppayah, “A COMPREHENSIVE REVIEW OF COURSE RECOMMENDER SYSTEMS IN E-LEARNING,” *Journal of Educators Online*, vol. 18, no. 1, pp. 23-35, 01//, 2021.

[3] D. Shah. “By The Numbers: MOOCs in 2020,” `https://www.classcentral.com/report/mooc-stats-2020/`.

[4] M. Gheisari, G. Wang, and M. Z. A. Bhuiyan, “A Survey on Deep Learning in Big Data.” pp. 173-180.

[5] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” *Knowledge-Based Systems*, vol. 46, pp. 109-132, 2013/07/01/, 2013.

[6] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, “A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning,” *Archives of Computational Methods in Engineering*, vol. 27, no. 4, pp. 1071-1092, 2020/09/01, 2020.

[7] H. Fujiyoshi, T. Hirakawa, and T. Yamashita, “Deep learning-based image recognition for autonomous driving,” *IATSS research*, vol. 43, no. 4, pp. 244-252, 2019.

[8] D. W. Otter, J. R. Medina, and J. K. Kalita, “A Survey of the Usages of Deep Learning for Natural Language Processing,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604-624, 2021.

[9] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436-444, 2015/05/01, 2015.

[10] Y. Li, and S. Kang, “Artificial bandwidth extension using deep neural network-based spectral envelope estimation and enhanced excitation estimation,” *IET Signal Processing*, vol. 10, no. 4, pp. 422-427, 2016.

[11] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez, “A survey on deep learning in medical image analysis,” *Medical Image Analysis*, vol. 42, pp. 60-88, 2017/12/01/, 2017.

[12] M. Mroczek, A. Desouky, and W. Sirry, “Imaging Transcriptomics in Neurodegenerative Diseases,” *Journal of Neuroimaging*, vol. 31, no. 2, pp. 244-250, 2021.

[13] C. Kleanthous, and S. Chatzis, “Gated Mixture Variational Autoencoders for Value Added Tax audit case selection,” *Knowledge-Based Systems*, vol. 188, pp. 105048, 2020/01/05/, 2020.

[14] T. Alashkar, S. Jiang, S. Wang, and Y. Fu, “Examples-rules guided deep neural network for makeup recommendation,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, California, USA, 2017, pp. 941–947.

[15] R. Devooght, and H. Bersini, “Long and Short-Term Recommendations with Recurrent Neural

Networks," in *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, Bratislava, Slovakia, 2017, pp. 13–21.

[16] A. M. Elkahky, Y. Song, and X. He, "A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems," in *Proceedings of the 24th International Conference on World Wide Web*, Florence, Italy, 2015, pp. 278–288.

[17] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations." pp. 191-198.

[18] S. B. Aher, and L. M. R. J. Lobo, "Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data," *Knowledge-Based Systems*, vol. 51, pp. 1-14, 2013/10/01/, 2013.

[19] H. Zhang, T. Huang, Z. Lv, S. Liu, and Z. Zhou, "MCRS: A course recommendation system for MOOCs," *Multimedia Tools and Applications*, vol. 77, no. 6, pp. 7051-7069, 2018/03/01, 2018.

[20] H. Zhang, T. Huang, Z. Lv, S. Liu, and H. Yang, "MOOCRC: A Highly Accurate Resource Recommendation Model for Use in MOOC Environments," *Mob. Netw. Appl.*, vol. 24, no. 1, pp. 34–46, 2019.

[21] Y. Pang, C. Liao, W. Tan, Y. Wu, and C. Zhou, "Recommendation for MOOC with Learner Neighbors and Learning Series," *Web Information Systems Engineering – WISE 2018*. pp. 379-394.

[22] HarvardX, "HarvardX Person-Course Academic Year 2013 De-Identified dataset, version 3.0," *Harvard Dataverse*, 2014.

[23] J. Maldonado-Mahauad, M. Pérez-Sanagustín, R. F. Kizilcec, N. Morales, and J. Munoz-Gama, "Mining theory-based patterns from Big data: Identifying self-regulated learning strategies in Massive Open Online Courses," *Computers in Human Behavior*, vol. 80, pp. 179-196, 2018/03/01/, 2018.

[24] J. Liang, J. Yang, Y. Wu, C. Li, and L. Zheng, "Big Data Application in Education: Dropout Prediction in Edx MOOCs." pp. 440-443.

[25] K. D. Strang, "Beyond engagement analytics: which online mixed-data factors predict student learning outcomes?," *Education and Information Technologies*, vol. 22, no. 3, pp. 917-937, 2017/05/01, 2017.

[26] S. P. Wang, and W. Kelly, "Video-based Big Data Analytics in Cyberlearning," *Journal of learning Analytics*, vol. 4, pp. 36-46, 2017.

[27] A. Marchand, and P. Marx, "Automated Product Recommendations with Preference-Based Explanations," *Journal of Retailing*, vol. 96, no. 3, pp. 328-343, 2020/09/01/, 2020.

[28] L. Duan, and Y. Xiong, "Big data analytics and business analytics," *Journal of Management Analytics*, vol. 2, no. 1, pp. 1-21, 2015/01/02, 2015.

[29] K. Chen, J. Powers, S. Guo, and F. Tian, "CRESP: Towards Optimal Resource Provisioning for MapReduce Computing in Public Clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 6, pp. 1403-1412, 2014.

[30] F. Shabestari, A. M. Rahmani, N. J. Navimipour, and S. Jabbehdari, "A taxonomy of software-based and hardware-based approaches for energy efficiency management in the Hadoop," *Journal of Network and Computer Applications*, vol. 126, pp. 162-177, 2019/01/15/, 2019.

[31] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, pp. 132306, 2020/03/01/, 2020.

[32] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1D & 2D CNN LSTM networks," *Biomedical Signal Processing and Control*, vol. 47, pp. 312-323, 2019.

[33] J. Zhao, F. Deng, Y. Cai, and J. Chen, "Long short-term memory-Fully connected (LSTM-FC) neural network for PM2. 5 concentration prediction," *Chemosphere*, vol. 220, pp. 486-492, 2019.

Siriporn Sakboonyarat is a teacher in the Department of Mathematics and Computing Science, Mahidol Wittayanusorn School, Thailand. She earned her B.Sc. in Computer Science from Nakhon Pathom Rajabhat University, Thailand, in 2001 and her M.S. in Computer Science from Silpakorn University in 2006. Currently, she is a Ph.D. candidate in Information Technology from Silpakorn University, Thailand. Her research interests are data mining and machine learning.

Panjai Tantatsanawong is an Associate professor at Computing Department, Faculty of Science, Silpakorn University. He received Doctor of Technology in Computer Science, Asian Institute of Technology (AIT), Thailand in 2000. He received Master of Science in Computer Science, Chulalongkorn University, Thailand in 1992 and Bachelor of Science, Mahidol university, Thailand in 1984. His research interests are data and information processing, internet of things, computer network and artificial intelligent.