

Learning from Student Browsing Data on E-Learning Platforms: Case Study

Sergen Cansiz
Morpa R&D Center
Istanbul, Turkey
sergen.cansiz@morpa.com.tr

Berk Sudan
Morpa R&D Center
Istanbul, Turkey
berk.sudan@morpa.com.tr

Emre Ogretici
Morpa R&D Center
Istanbul, Turkey
emre@morpa.com.tr

Mehmet S. Aktas
YTU Computer Engineering Dept
Istanbul, Turkey
aktas@yildiz.edu.tr

Abstract—Interpretation of the behaviors of students in e-learning platforms with machine learning models has become an emerging need in recent years. Increase in the number of registered students on e-learning platforms is one of the reasons for choosing machine learning models. Tracking, modeling and understanding student activities gets more complex when the number of students is increased. This study is focusing modeling student activities on e-learning platforms with Complex Event Processing (CEP), Association Rule Mining (ARM) and Clustering methods based on distributed software architecture. Within the scope of this study, different modules that work real-time have been developed. An admin panel has been also developed in order to control all modules and track the student actions. Performance results of modules also obtained and evaluated on distributed system architecture.

Index Terms—e-learning, complex event processing, association rule mining, clustering

I INTRODUCTION

E-LEARNING platforms are increasingly preferred and used by students. Students prefer the different learning models provided by e-learning platforms such as self-learning and teacher-assisted learning [1]. However, the integration of advancing technology and artificial intelligence applications into e-learning platforms also adds extra features to e-learning platforms and contributes to students' learning processes [2]. These features also make e-learning platforms more useful and effective on learning. With the artificial intelligence integration, students can be tracked on the basis of e-learning platforms, and various suggestions can be offered to students during training. There are researches made on prediction models by using various machine learning algorithms in order to present these suggestions to users (students) [3].

The purpose of most research and developed applications based on machine learning on e-learning platforms is to ease students' learning processes and to bring the learning performance to the highest level. The data derived from students' actions on the platform directly affects the machine learning models to produce accurate and consistent results. Because of that, it's very important to extract the students' personal behaviors on the platform by considering that the learning style of each student may differ from each other. In addition to predicting student success with machine learning algorithms using behavioral data; students can also be categorized according to the platform usage patterns and learning processes. Thus, system analysts, teachers and parents can evaluate students according to their categories and take actions that will improve the student's performance.

Apart from the e-learning platform, it is a known fact that a teacher pursues students in the class and makes suggestions according to their learning style, corrects their mistakes and guides the student directly. In e-learning platforms, these procedures are expected to be done by experts or teachers. It is almost impossible for experts to provide individual advice to students considering an e-learning platform with thousands of or maybe millions of users. In such cases, In e-learning platforms where teacher membership is available, teachers can follow the students they are responsible for through the system and take the necessary actions. In addition, since there is no physical classroom environment in e-learning platforms, it is impossible for a teacher to monitor and evaluate all students in real time. Therefore, the evaluation process is done by the teacher every other day or weekly after completing the student education. Such situations may also cause delay in actions that should be taken during the student's learning process.

It is possible to monitor whether the behaviors of the users on a system show certain patterns in real-time with the help of the tool developed on Complex Event Processing (CEP). Students' behaviors in an e-learning platform can be interpreted according to certain patterns and action can be taken instantly by using these tools. In this way, when the teachers cannot monitor and evaluate the students' behaviors, our tools can interpret the behaviors of the students instantly according to the directives determined by the experts or teachers.

Each student may have common behaviors as well as different behaviors during the learning process. As an example, a student may need information on "lecture A" while on the exercise of "subject B". In such cases, it is possible for students to return to the "lecture A" during or after the exercise of "subject B". In this way students can get the prior knowledge they need. Identifying these and similar actions automatically and showing them to expert analysts or teachers can form the basis for actions to be taken for other students. In addition, these actions can be added on the CEP by experts as a separate pattern (or rule) to automatically send recommendation notifications to students. Plus, students can be divided into clusters according to their behavior data in order to analyze students' situations and interpret their behavior cumulatively. Thus, the learning processes of the students can be examined by generalizing them according to their behaviors over different clusters.

Within the scope of this research, a demo e-learning platform has been developed on the CEP module that detects the students' actions according to certain rules (patterns), performs the predefined notification action according to the detected patterns and extracts the common behaviors (association rules) in the students' actions. In addition, students were divided into clusters according to various

behavioral patterns gathered from an e-learning site. It is aimed to facilitate the learning process by accurately monitoring and interpreting students' behaviors.

In this study, similar studies in the literature are discussed in Section 2. In Section 3, the complex methodology that is followed while developing Complex Event Processing, association rules and developing clustering modules is presented.

II. LITERATURE REVIEW

Complex Event Processing (CEP) is a very effective method for processing real-time streaming data and extracting certain patterns from these streaming data bounded or unbounded [4,5]. Simultaneously, with the help of the distributed system technologies, CEP can work integrated with machine learning models [6]. This method, which is frequently used within the scope of Business Performance Management (BPM), is used in companies with a large number of customers or in e-commerce sites to identify certain patterns based on users' behavior (user behaviors, shopping etc.) and to take necessary actions [7]. The CEP module used in e-commerce sites which are focused on purchasing can be also used in e-learning systems with a focus on student success.

As machine learning techniques have improved, there has been an increase in the amount of applications on student education. Since the early 2000s, students' behaviors have been tried to be modeled by machine learning algorithms in various studies [8,9]. Various model approaches which are used to predict the performance of students were developed on machine learning models [10,11]. One of the most important factors in predicting future success by measuring students' performance and modeling their behavior on e-learning platforms is the data collected from students. In a study, success prediction was made by collecting the student's processing time, idle time, total activities, key-strokes data [12]. The machine learning model developed based on the obtained results makes successful predictions. This situation reveals the importance of the data received from the student. With the help of the CEP tools, students' actions can be stored instantly by a rule-based pre-processing module. These actions can be used on other machine learning models later. Similarly, in a study, CEP and IoT data were collected with predefined rules and these data were used by a machine learning model [13].

Predicting the success of students by using machine learning models and making recommendations out of them is a widely used method in e-learning platforms. However, in some cases it is possible to make recommendations by analyzing student actions with a real-time and rule-based system without the need for a machine learning model. In a study [14], the actions of students in the learning process were tracked based on some rules. In this rule-based system, teachers were given the option of adding and setting rules. Thus, it is aimed to improve the learning processes of the students. The rules are defined based on the order of course lectures to prevent the students from attending one lecture without attending the previous one(s). However, in this study, the rules are not determined by the Complex Event Processing module. In another study, a module makes recommendations to students using association rule mining. In this module, the user actions are read from the server log file and the actions of the users are taken sequentially. Sequential actions have been analyzed and, using data mining

methods, a module which sends recommendation notifications to students who exhibit similar behavior has been developed [15]. In a study on language learning, a module which tracks user actions with predefined rules developed using complex event processing methods [16]. This research shows that Complex Event Processing (CEP) can be used on e-learning platforms.

In an e-learning platform where rule-based operations are performed with the complex Event Processing module, rules can be created by experts or by using machine learning methods. By finding common behaviors of students on the platform, significant and successful behaviors can be added as a rule. Rules can be extracted based on common behaviors using Association Rule Mining (ARM) methods, as in [17]. There are studies in which Association Rule Mining is applied in the field of health, finance and e-commerce, as in [18]. However, no research has been found in the field of e-learning.

We observe the use of complex event processing techniques in different domains. This technique is used to detect patterns in different domains such as social media [19-22], internet of things [23-24], cloud computing [25], e-commerce [26] and real time streaming based applications [27]. Our study focuses on detecting predefined patterns within the data collected from the clickstream data generated by user-system interactions. In recent years, in order to keep track of the events in a system, provenance based systems have been utilized [28-32]. In this study, our main focus is to keep track of the traces of user actions on the e-learning platform and detect predefined patterns. We also observe studies in the field of service oriented architecture based systems with a focus on high performance and scalability [33-41]. These systems are designed and implemented based distributed system software architecture. In this study, we also utilize distributed system based open-source software systems such as Apache Flink. However, our main focus is the use of complex event processing for detecting predefined patterns on the user-system interaction data collected from log files.

III. METHODOLOGY

In this research, a demo e-learning platform and admin panel with different integrated modules were developed. Demo e-learning platform contains lectures, exercises and exam options in various courses and lectures. The modules where the students' behaviors are tracked, modeled and presented with visualization tools can be controlled by the admin panel. Thus, the actions of students, both private and general, can be tracked.

Two different methods based on instant actions and reporting have been developed in order to increase student success by monitoring the actions of students on the platform. Instant actions are based on recommendations when a student is on the platform. Reporting is intended to be evaluated by the teacher or specialist after students completed courses. Instant action transforms the student actions into recommendation actions based on certain rules. In the reporting section, the actions of all students are evaluated and divided into certain clusters. With these clusters, which are expected to be checked by teachers and specialists, it is possible to determine which cluster the student is included in. Thus, the opportunity to be evaluated with the success-based clustering module is provided to the students. The following

modules have been developed to implement the planned system.

- Web server log parser module for tracking user actions on the platform
- Rule engine which rules are managed, and their validity are checked
- Complex Event Processing module to detected using predefined rules
- Association Rule Mining module to get new usage patterns (rules) from students' actions
- Action executor module to take actions over defined rules
- Clustering module which divide students into clusters (segments) according to their usage behavior

The real-time log parser module is allowed to read log files from Web Server. It takes each request on the web server and analyzes page visits or materials activities of the user by separating them according to the session ID and user ID. The student activities obtained by log parser are sent to different modules over the messaging service instead of database. Thus, the transmission of user activities to different modules is done with high speed performance.

The student activities sent by the log parser are taken by the Complex Event Processing (CEP) module to determine if it recognizes any certain pattern rules. The rules that CEP looks for are stored in the database which is controlled by the rule engine. These rules can be created by experts considering factors of affecting students' success. This rule engine also integrated into the admin panel and rules can be updated via an interface. Experts can also deactivate or activate the rules and check their expiry dates.

As mentioned before, the rules used to identify complex actions that affect students' success can be created by experts as well as by analyzing the student's common actions within the system. Therefore, the Association Rule Mining (ARM) module has been developed in order to detect common behaviors of the users based on their actions in the platform. Considering that thousands of students can be enrolled on the platform, distributed system architecture has been used for faster performance on the ARM module. FP Growth algorithm has been used for detecting students' associated behaviors. These common behaviors can also be adjusted according to students' exam scores. Thus, the behavior of successful students and unsuccessful students on the platform can be compared. For instance, with FP Growth algorithm, it can be found that most of the students who are successful on subject X, first enter subject X's lecture and then enter the subject Y's lecture. In such cases, it can be concluded that subject Y's lecture has a positive effect on subject X. These types of student behaviors can be found with FP Growth algorithm on the ARM module and create new rules (patterns) on the rule engine in order to be detected with the CEP module. However, the rules that have been found by the ARM module can be seen and managed by teachers and experts from the admin panel. They can add these rules to the rule engine and activate them. The developed CEP module works while students are studying on the platform. The ARM module is used every other day to obtain different rules. In this way, student behaviors can be controlled real time and differences in their behavior can be tracked.

It is very important to take actions instantly to increase the success of students based on the results obtained with the

CEP and ARM modules. Hence, the options to reach students via Pop-up, SMS and E-mail have been considered. If detected rules are supposed to reach the student instantly Pop-up option can be considered. Otherwise, email option can be considered. The module called Action Executor has features to handle different notification options or transferring data across the modules. This module works with CEP and Rule Engine. System is triggered by the detection of the pattern and sends feedback to students based on the rules defined in the rule engine. In some cases, it saves data for machine learning methods.

In addition to instant actions, students are divided into segments with various clustering techniques to be presented to the evaluation of teachers and experts. The Clustering Module was originally developed through student behaviors from a demo e-learning platform and designed as improvable with new student actions. K-Means, Bisecting K-Means and Gaussian Mixture algorithms are used for clustering. Thus, student clusters (segments) can be evaluated with different algorithms. User actions on the platform are stored in the database and used to create new clustering models every other day. Clustering results obtained are stored in the database and made accessible through the admin panel. The results are displayed to experts and teachers with different visualization and reporting options.

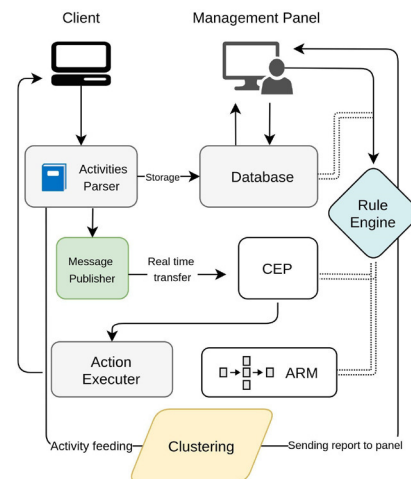


Fig. 1. System Architecture

Speed performance of ARM and Clustering modules can vary depending on the number of students enrolled in e-learning platforms. Along with new data from students, the data used for ARM and Clustering algorithms are updated every other day. This will cause the data size to increase and the algorithms to run slowly over time. For this reason, distributed system architecture is used and performance of ARM and Clustering modules on single node and multi node machines were evaluated as speed oriented. The system architecture shown in Figure 1 has been designed considering latency and scalability. Performance reports of the architecture on single node and double node are discussed in Section 5.

IV. PROTOTYPE

In order to implement the proposed system architecture and performance evaluation specified in the methodology section, the e-learning platform and admin panel prototypes in which different modules can be controlled have been developed. The e-learning platform has lectures on various topics, exercises and exam options as well as different course

options. While the e-learning platform was developed with PHP on Apache Server, the admin panel was developed using ReactJS framework on Node.js. These two different platforms can be managed under a single domain via Nginx Proxy Server. In addition, MongoDB was used to manage the developed modules and store the processed student actions. Data transfers between the demo e-learning platform and the admin panel are provided through the NodeJS API services through different modules.

Apache Server Log files are parsed and stored with parameters such as session ID and user ID so that student actions can be used on different modules without slowing down the system. The log parser running in real time was developed using Python Programming Language. All requests on the server are saved in a customized format. The most important part in parsing the log file where all the requests are stored is that the student actions can be parsed individually. Therefore, the Log Parser works only with certain requests and rules that save certain data which comes with the requests. These rules are modified so that they can be created from the admin panel. Kafka service was used to send the data parsed by the log parser to other modules in real time. In addition, user actions are stored with the NodeJS API service so that the data in MongoDB can be accessed from the admin panel. While the data is transmitted to the modules through Kafka topics, it can be monitored on the admin panel with the help of MongoDB. The architecture of this system is shown in Figure 2.

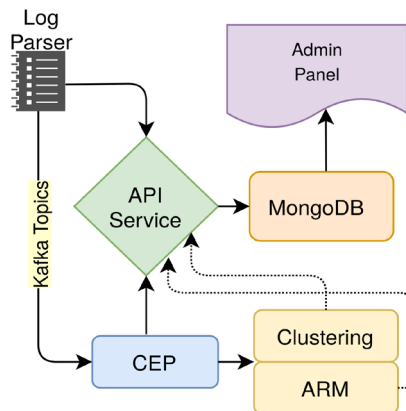


Fig. 2. API Service Diagram

Student activities obtained by the Log Parser with defined rules are saved with certain codes. In this way, various actions can be defined in the modules and processing can be performed according to these actions. Table 1 shows the activities and sample data in these activities. In addition to the data specified in Table I, “user ID” and “session ID” are also collected for each action.

TABLE I. ACTIVITIES AND DATA IN ACTIVITIES

ACTIVITY	DATA
VISIT.LOGIN	Login Data
ACTION.LOGIN	Login Data
VISIT.MATERIAL	Material ID, Lecture ID
EXIT.MATERIAL	Material ID, Lecture ID, Participation Rate
ACTION.MATERIAL	Material ID, Lecture ID, Action Result Data
START.EXAM	Material ID, Lecture ID
FINISH.EXAM	Material ID, Lecture ID, Exam Result Data

A. Complex Event Processing Module

CEP module developed on Apache Flink is fed by student actions extracted by the Log parser via Kafka topic. These actions are sent in JSON format. The aim of the CEP module is to recognize certain patterns of student activities. These patterns can be managed with the rule engine and accessed from the admin panel. New rules in the rule engine can be created from the admin panel, and the created rules, the expiry date and activity of the rules can be managed and updated. In this way, it is ensured that all pattern rules can be managed efficiently. The Rule Engine interface is shown in Figure 3.

It is aimed to increase student success with the rule patterns in Apache Flink. For instance, patterns as an expected scenario of a course can be defined in the rule file. When the student acts in accordance with the defined scenario, Apache Flink accepts the pattern and sends the related notification to the Action Executor module with details. The student is then notified by the Action Executor.

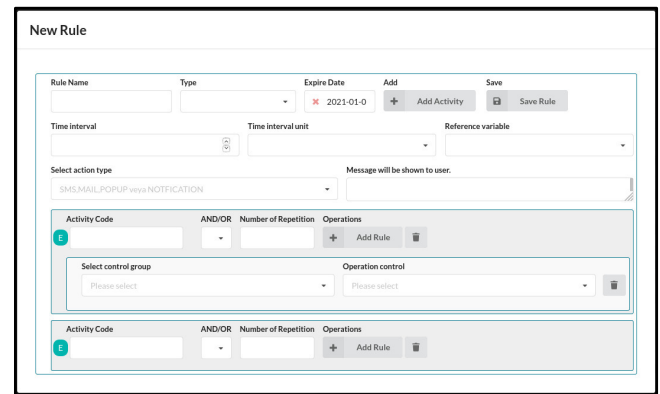


Fig. 3. Rule Creation and Management Screen in Rule Engine

B. Association Rule Mining Module

The patterns specified in the Rule Engine for the Complex Event Processing module are determined by education specialists. Although specialists can manually analyze student success from their actions, big amounts of data, various student actions don't allow specialists to analyze all data. In addition, it is possible for each student to follow different learning paths. For this reason, it is very important to monitor the behavior of the students in the platform and to find the common behaviors of all students. The common behaviors can be analyzed by showing them to the education specialists and these behaviors can be defined in rules to increase the success. In this way, the common behaviors of the students in the platform can be determined with the help of the ARM module.

The most important part of the ARM module is the way students' common behaviors are extracted. In this study, since the aim is to increase student success, the association rules were extracted based on the end-of-lecture exams. In this way, students' common behaviors can be analyzed according to their exam success. Predefined sequential actions of a student can be tracked through pattern rules via the CEP module. Thus, the CEP module was also used to obtain data for the ARM module. The CEP module determines all the activities that a student does from the first time he/she enters a subject until the subject's exam. Students can enter other lectures between a lecture and its exam. With the help of the association rules, related lectures of the lectures can be obtained. The sequential actions detected by the CEP module are shown in Figure 4.

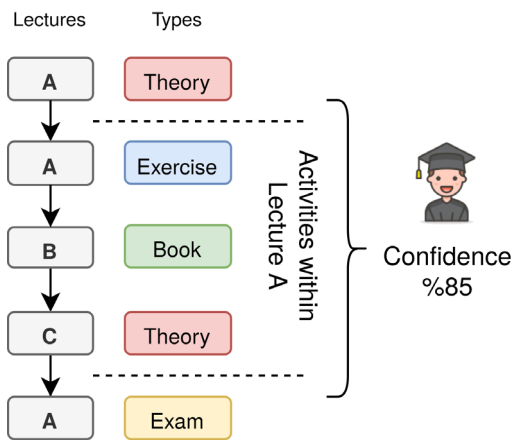


Fig. 4. Extracting ARM Data via CEP Module

Figure 4 shows the activities of a student from the beginning of the lecture A to the subject until he / she takes the exam. While a student was in lecture A, this student also entered the exercise of lecture A, the book of lecture B and the lecture C. This sequence of actions (pattern) is recognized and processed by the CEP module. In this way, with the help of the ARM module, frequent sequences are determined by processing the activities of all students. Assuming that the scenario shown in Figure 4 occurs in the same way on 85 students out of 100 students, this indicates that a student who started the lecture A and solved his/her exercise needs book B and the animation of lecture C with a probability of 0.85. In this way, specific lectures on the e-learning platform can be added to the Rule Engine by extracting the association rules from the materials. Thus, with the help of the CEP module, a student who enters lecture A and solves the exercise of lecture A can be recommended to take a look at lecture B and the lecture C.

The ARM module was developed on Apache Spark using the FP Growth algorithm. Action sequences (patterns) recognized by the CEP module are stored in HDFS. Simultaneously, data is also stored in MongoDB in order to be accessible from the admin panel. With the help of the ARM module, the data in HDFS are obtained every other day on the platform by using Apache Spark. The results are displayed in the admin panel along with the "confidence" values. Association rules displayed in the admin panel can be added as new rules to the Rule Engine managed by experts.

C. Clustering Module

On the platform, a clustering module has been developed to examine the students' behavior in different segments. With

this clustering module, students are divided into clusters according to their behavior on the platform, such as city, school, classroom and lessons. These behaviors of the students have been extracted by considering their performance in various materials within lectures. The attributes used when creating clusters are mainly: The participation rate of lecture material, the success rate in the exercises, and the overall participation rate.

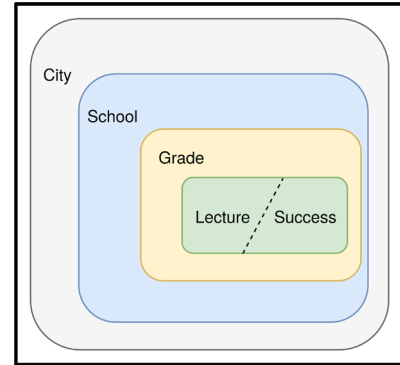


Fig. 5. Macro and Micro Pivots

E-learning platforms can be used for personal use as well as institutional use by public and private schools. Therefore, segments are evaluated with certain pivots instead of all students in the e-learning platform. The segments handled in different pivots are two types: "micro" and "macro". Macro segmentation is done by considering all students in a city or all students enrolled in a school. Micro segmentation is done over the grade levels of the students. In this way, students in different locations, schools and grade levels can be evaluated according to the separate clusters. In addition, all pivots can be created by associating them with the lecture and success rate. Students can be separated by pivots according to their success in certain topics, and differences in behavior between successful and unsuccessful students can be found. Macro and Micro structure, lecture and success divisions are summarized in Figure 5.

Student actions data used in the clustering module is obtained using CEP module. After the lecture exam, the rate of completion of the lecture, the success rate in the exercises, the participation rate of all materials and the success rate in the lecture exam are determined with the data transferred from the Log Parser to the CEP module. The detected behavior data stored on HDFS and MongoDB. Student behavior data on HDFS are processed every other day with the Clustering Module and divided into clusters with pivots.

Considering the data collected every day, it is highly probable that both the increase in data size and the change in data values day by day. Clustering Module has been developed in distributed system architecture to prevent performance from being negatively affected by data increase. Performance results of the Clustering Module running on HDFS and Apache Spark is discussed in the "Performance Evaluation" section. Considering the changes in data amount, different clustering techniques and optimum cluster number finding methods were used. The cluster module gets results with various clustering techniques in distributed system architecture and stores these results in MongoDB. The results stored in MongoDB are visualized in the admin panel.

K-Means, Gaussian Mixture and Bisecting K-Means algorithms are used in the Clustering Module. These algorithms can yield different results in the same data.

Therefore, by evaluating different results, it can be determined clearly which cluster students belong to. Another important point in the clustering module is that the optimum cluster number and optimum seed values can be determined. Accordingly, optimum cluster numbers and optimum seed values are obtained with Elbow and Silhouette methods.

Matplotlib library was used to visualize cluster results. Micro segmentation was performed by using the "Success" and "Course ID" attributes as pivots. A separate clustering was made according to each different value of the "Success" and "Course ID" attributes. The results that have been obtained from the K-Means algorithm by using K-Means & Elbow methods to find optimum seed and optimal cluster number is shown in Figure 6. The different colors in the images show different clusters, and the cross symbols show the centers (centroids) of these clusters. The behavior of students who are successful for a specific lesson can be easily interpreted on the platform from this type of visualization.

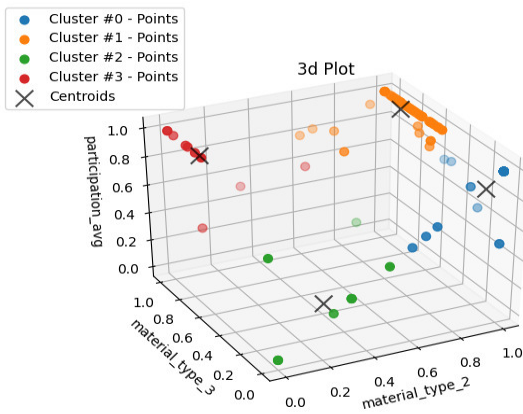


Fig. 6. Example Clustering Result for 3 Features

V. PERFORMANCE EVALUATION

The speed performances of the Clustering Module and ARM modules developed on distributed system architecture have been tested with different data sizes. Performance results of the modules developed using Spark and HDFS were compared on single node and double node machines. Experimental setup of single node and double node is given in Figure 7. System information of master and slave nodes used in performance evaluation is shown in Table 2.

The results of the performance test carried out with the both single and double node HDFS and Spark installation for the ARM module are shown in Figure 8. The data size was increased exponentially from 1 MB to 10 GB, and the speed of completely extracting the association rules was tested through the student behavior data of the ARM module. When the obtained results are examined, 1 MB of data is processed in an average of 9.54 seconds, while 10 GB of data is processed in an average of 519.03 seconds. Considering that the ARM module operates every other day, it can be said that the developed module works with enough performance.

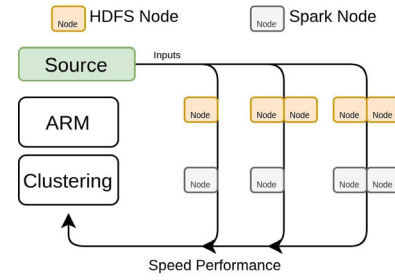


Fig. 7. Experimental Setup of Performance Evaluation

TABLE II. SYSTEM INFORMATION OF NODES

	Master Node	Slave Nodes 1 & 2
Operating System	Ubuntu 18.04.4	
CPU	Intel(R)Core (TM) i5-7200U CPU @ 2.50GHz	
Number of Cores	2	1
Virtual / Physical	Physical	Virtual
Memory	10240 MB	2048 MB
Memory Used	"Not Necessary"	1024 MB

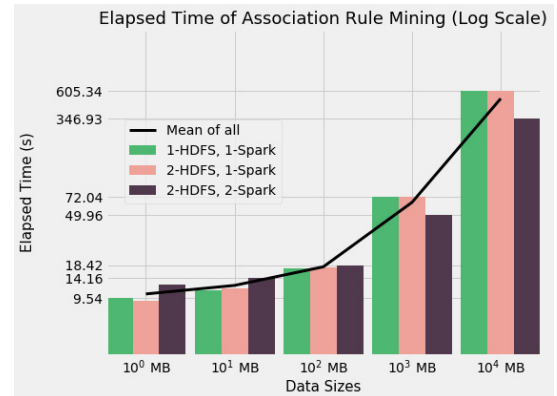


Fig. 8. Performance Results for ARM module on Single and Double Node

As shown in Figure 8, as data grows, the increase in the amount of spark nodes becomes more essential. When data is small, e.g. 1 MB, there is not much difference between single and double nodes. Since HDFS is used for storing data only, changes in the number of nodes running HDFS don't affect the results much. However, if our data is huge and we need stability, then multi-node HDFS would be a good solution. Spark, on the other hand, affects the results significantly. As data grows, its effect can be seen more clearly.

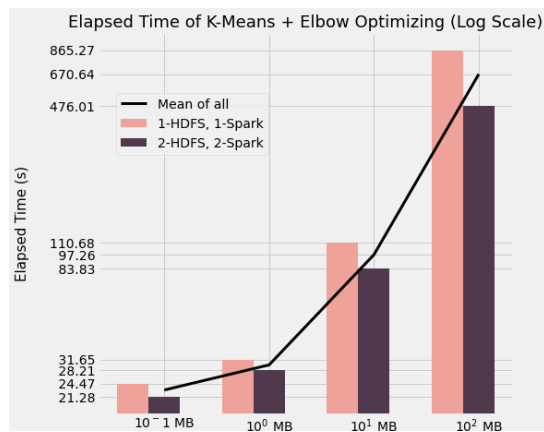


Fig. 9. Performance Results for Clustering Module on Single and Double Node

K-Means Clustering result with datasets 100KB, 1MB, 10MB, 100MB is shown in Figure 9. Figure clearly indicates that as data grows, multi-node processing is getting much more important. Also, when data size is 100MB, although the number of nodes was doubled, the elapsed time decreased by more than twice. This shows the effectiveness of distributed structure.

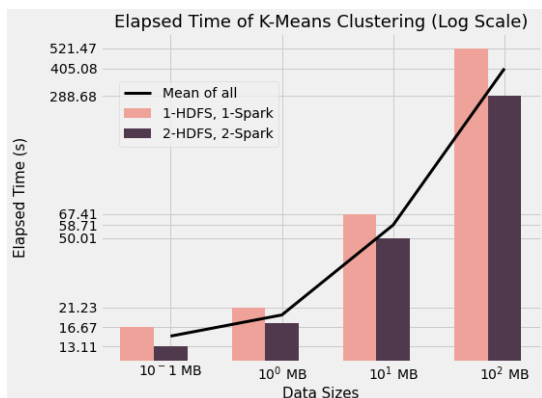


Fig. 10. Performance Results for Optimizing (Finding Optimum Cluster & Seed Values) Module on Single and Double Node

K-Means and Elbow methods are used together to calculate the number of optimum k (number of clusters) and value of initial seeds. Since this is an unsupervised problem and we don't know how many segments are present in our data, optimizing the seeds and k was important. As shown in Figure 10, the rate of data growth directly affects the performance. Also, the cost of optimizing is more than the cost of clustering, because optimizing includes clustering multiple times. Optimizing module has parameters like: "number of seed values trying", "range of amount of k to be tried". When these parameters change, performance results are changed accordingly. In this case, "number of seed values trying" chosen as 3 and "range of amount of k to be tried" chosen as the range between 2 and 5.

VI. CONCLUSIONS AND FUTURE STUDIES

In this study, it is aimed to increase student success and improve their learning experiences by using CEP, ARM and Clustering modules. Accordingly, tests were applied on a demo e-learning platform using state-of-the-art technologies. It has developed modules in which students' actions on the system can be monitored and interpreted in real time. Features have been added to the demo e-learning platform for necessary feedback actions such as SMS, E-mail and Pop-up notification. In addition, with ARM and Clustering modules,

such tools have been developed that make success-oriented inferences. Thus, students can be analyzed individually.

Besides the developed modules, an admin panel has been developed which the entire e-learning platform and the different features integrated can be controlled. In this way, teachers and experts are provided with the opportunity to review and make changes through an available interface. Thanks to this admin panel, education experts are provided with all possibilities, from the ability to create the patterns obtained in the CEP module to the message to be displayed to the user.

In addition to this study where modules that can detect student behavior data according to certain patterns, create different patterns with the ARM module and divide students into segments, an anomaly detection module can be added in future studies. Thus, it can be determined whether the students exhibit behavior different than expected in their learning processes and action can be taken according to the different situations obtained. Students can also be evaluated through their unexpected behavior during the learning process.

ACKNOWLEDGMENT

This study was supported by Tubitak Teydeb Project ID: 3189164 Grant. We thank Morpa R&D Center for providing the computational facilities and datasets that made this study possible.

REFERENCES

- [1] S. Liaw, H. Huang, and G. Chen, "Surveying instructor and learner attitudes toward e-learning," *Computers & Education*, vol. 49, no. 4, pp. 1066–1080, 2007.
- [2] F. Hussain, "E-learning 3.0 = E-learning 2.0 + Web 3.0?," IADIS International Conference on Cognition and Exploratory Learning in Digital Age, CELDA (Report Paper), 2012.
- [3] S. B. Aher, L.M.R.J. Lobo, "Best Combination of Machine Learning Algorithms for Course Recommendation System in E-learning," *International Journal of Computer Applications*, 41(6), 2012.
- [4] D. C. Luckham, "The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems", Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 2001.
- [5] G. Cugola and A. Margara, "Processing flows of information," *ACM Computing Surveys*, vol. 44, no. 3, pp. 1–62, 2012.
- [6] N. Mehdiyev, J. Krumeich, D. Enke, D. Werth, and P. Loos, "Determination of Rule Patterns in Complex Event Processing Using Machine Learning Techniques," *Procedia Computer Science*, vol. 61, pp. 395–401, 2015.
- [7] S. Chen, J. Jeng, and H. Chang, "Complex Event Processing using Simple Rule-based Event Correlation Engines for Business Performance Management," *The 8th IEEE International Conference on E-Commerce Technology and The 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services (CEC/EEE'06)*, 2006.
- [8] F. Esposito, O. Licchelli, G. Semeraro, "Discovering Student Models in e-learning Systems", *Journal of Universal Computer Science*, 10 (1), pp. 47–57, 2004.
- [9] R. Sison, M. Shimura, "Student Modeling and Machine Learning", *International Journal of Artificial Intelligence in Education (IJAIED)*, 1998, 9, pp.128-158, 2008.
- [10] N. Thai-Nghe, T. Horvath, L. Schmidt-Thieme, "Factorization models for forecasting student performance", *Proceedings of the 4th international conference on educational data mining*, Eindhoven, The Netherlands, July 62011
- [11] K. Pliakos, S.-H. Joo, J. Y. Park, F. Cornillie, C. Vens, and W. V. D. Noortgate, "Integrating machine learning into item response theory for addressing the cold start problem in adaptive learning systems," *Computers & Education*, vol. 137, pp. 91–103, 2019.
- [12] M. Hussain, W. Zhu, W. Zhang, S. M. R. Abidi, and S. Ali, "Using machine learning to predict student difficulties from learning session data," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 381–407, 2018.
- [13] J. Roldán, J. Boubeta-Puig, J. L. Martínez, and G. Ortiz, "Integrating complex event processing and machine learning: An intelligent architecture for detecting IoT security attacks," *Expert Systems with Applications*, vol. 149, p. 113251, 2020.
- [14] M. Cesarini, M. Monga, and R. Tedesco, "Carrying on the e-learning process with a workflow management engine," *Proceedings of the 2004 ACM symposium on Applied computing - SAC '04*, 2004.
- [15] O. Zaiane, "Building a recommender agent for e-learning systems," *International Conference on Computers in Education*, 2002. Proceedings., 2002.

- 16 D. Jin, S. Shi, Y. Zhang, H. Abbas, and T.-T. Goh, "A complex event processing framework for an adaptive language learning system," *Future Generation Computer Systems*, vol. 92, pp. 857–867, 2019.
- 17 Jung, J., Park H., 2020. SAX-ARM: Deviant event pattern discovery from multivariate time series using symbolic aggregate approximation and association rule mining, *Expert Systems with Applications*, 141, <https://doi.org/10.1016/j.eswa.2019.112950>
- 18 K. Vougas, T. Sakellariopoulos, A. Kotsinas, G.-R. P. Foukas, A. Ntargaras, F. Koinis, A. Polyzos, V. Myrianthopoulos, H. Zhou, S. Narang, V. Georgoulis, L. Alexopoulos, I. Aifantis, P. A. Townsend, P. Sfikakis, R. Fitzgerald, D. Thanos, J. Bartek, R. Petty, A. Tsigiros, and V. G. Gorgoulis, "Machine learning and data mining frameworks for predicting drug response in cancer: An overview and a novel in silico screening process based on association rule mining," *Pharmacology & Therapeutics*, vol. 203, p. 107395, 2019.
- 19 Baeth, M.J. et al. (2019). Detecting misinformation in social networks using provenance data, *CONCURR COMP-PRACT E*, 31(3).
- 20 Baeth M. J. et al. (2018) An approach to custom privacy policy violation detection problems using big social provenance data, *CONCURR COMP-PRACT E*, 30(21).
- 21 Baeth, M.J. et al. (2017). Detecting misinformation in social networks using provenance data, *SKG-17*.
- 22 Baeth, M.J. et al. (2015). On the Detection of Information Pollution and Violation of Copyrights in the Social Web, *SOCA-15*.
- 23 Dundar, B. et al. (2016) A Big Data Processing Framework for Self Healing Internet of Things Applications, *SKG-16*.
- 24 Aktas, M.S. et al. (2019), Provenance aware run-time verification of things for selfhealing Internet of Things applications, *CONCURR COMP-PRACT E*, DOI: 10.1002/cpe.4263.
- 25 Aktas M.S., (2018) Hybrid cloud computing monitoring software architecture, *CONCURR COMP-PRACT E*, 30(21).
- 26 Aktas M.S. (2019) Detecting Complex Events With Real Time Monitoring Infrastructure On Event-Based Systems, *Pamukkale Univ Muh Bilim Derg.* 2019; 25(2): 199-207.
- 27 Fox, G. et al. (2006). *Real Time Streaming Data Grid Applications. Distributed Cooperative Laboratories: Networking, Instrumentation, and Measurements*. Editors: Davoli, F. Plazzo, S, Zappatore, S., pp. 253-267.
- 28 Riveni, M. et al. (2019). Application of provenance in social computing: A case study, *CONCURR COMP-PRACT E*, 31(3).
- 29 Tas, Y. et al. (2016) An Approach to Standalone Provenance Systems for Big Provenance Data, *SKG-16*.
- 30 Aktas M.S. et al. (2010). High performance hybrid information service architecture, *CONCURR COMP-PRACT E*, 22(15).
- 31 Aktas M.S., et al. (2008) XML metadata services, *CONCURR COMP-PRACT E*, 20 (7).
- 32 Aktas, M.S. et al. (2007). Fault tolerant high-performance Information Services for dynamic collections of Grid and Web services, *FUTURE GENER COMP SY*, 23(3).
- 33 Pierce, M.E. et al. (2008). The QuakeSim project: Web services for managing geophysical data and applications, *PURE APPL GEOPHYS*, 165(3-4).
- 34 Aydin, G. et al. (2005). SERVOGrid complexity computational environments (CCE) integrated performance analysis, *GRID-05*.
- 35 Aktas, M. et al. (2006). iSERVO: Implementing the International Solid Earth Research Virtual Observatory by Integrating Computational Grid and Geographical Information Web Services, *PURE APPL GEOPHYS*, 163(11-12).
- 36 Aydin, G. et al (2008). Building and applying geographical information system Grids, *CONCURR COMP-PRACT E*, 20 (14).
- 37 Oh, S. et al. (2010) Mobile Web Service Architecture Using Contextstore, *KSII T Internet Info*, Volume 4.
- 38 Nacar, M.A. et al. (2007) VLab: collaborative Grid services and portals to support computational material science, *CONCURR COMP-PRACT E*, 19 (12).
- 39 Aktas, M.S. et al. (2004). A web based conversational case-based recommender system for ontology aided metadata discovery, *GRID-04*, pp:69-75.
- 40 Aktas, M. (2007) A Federated Approach to Information Management in Grids. *INT J WEB SERV RES*, 7(1).
- 41 Fox, G. et al, (2006) Grids for real time data applications. *Parallel Processing and Applied Mathematics*, Vol:3911, Book Editors: Wyrzykowski, R and Dongarra, J and Meye, N and Wasniewski, J.