# A Hybrid Approach for Usability Evaluation of Learning Management Systems Using Machine Learning Algorithms

Richard Torres-Molina
*Department of Computer Science*
*Virginia Polytechnic Institute and State University*
Blacksburg, USA
richardat@vt.edu

Mohammed Seyam
*Department of Computer Science*
*Virginia Polytechnic Institute and State University*
Blacksburg, USA
seyam@vt.edu

***Abstract*—This research full paper describes a hybrid approach based on data, questionnaire answers, and machine learning algorithms to predict usability scores in Learning Management Systems (LMSs) to improve student learning and satisfaction. Students need to achieve their learning goals by interacting with LMSs. To attain these goals, usability evaluation ensures effectiveness (task completion), efficiency (time measurement), and satisfaction (positive attitude). Usability evaluation usually follows questionnaires, user testing of the LMS, and expert reviews. Although these methods are widely used due to several benefits, they face challenges related to trying these software systems multiple times until the system satisfies student needs, human subjectivity perception, and lack of software system adaptability. We propose this hybrid approach to face these challenges, promote student engagement with the system, and create a better design in the LMS courses. The aim is to identify features extracted from the LMS to predict usability scores with machine learning techniques. We evaluated this strategy through a case study with data collected from undergraduate students at a public university in the United States. The students' tasks were answering a quiz, posting in a forum, and uploading an assignment. These activities in the LMS allow the extraction of ten features into the machine learning algorithms. These attributes are time quiz, time forum, time assignment, grade quiz, word count message post, file size, file type, clicks module quiz, clicks module forum, and clicks module assignment. The four targets are from scores of the System Usability Scale and UseLearn questionnaires. Random Forest produces the best performance of average mean square error and root mean square error among machine learning algorithms. The results are promising, though there are alternatives for improvements. Our proposed approach contributes to the engineering and computing education field by providing a predictive tool for usability scores to improve the student learning experience and the components of the LMS.**

***Index Terms*—usability evaluation, machine learning, learning management systems**

## I. Introduction

The International Organization for Standardization (ISO) established the usability definition in ISO 9241-11 [1]. The concept refers to user goals achievement with a product considering different factors: task completion (effectiveness), time measurement (efficiency), and positive attitude (satisfaction). Jakob Nielsen [2] extended these factors to easy to learn (learnability), easy to remember (memorability), and low error rate (errors). Considering this perspective, usability evaluation (UE) methods have emerged to evaluate the usability of software systems to guarantee quality [3]. UE methods in software systems - including Learning Management Systems (LMSs) - usually follow a traditional subjective approach through questionnaires [4], user testing [5], heuristics [6]. LMSs [7] are electronic platforms with services such as course creation, course management, communication, and assessment like Moodle, Canvas, and Blackboard. The subjective approach helps find usability issues. However, they present gaps for improvement. These gaps are the time needed to test a software system multiple times until the system satisfies user needs and human subjectivity perception. Also, they lack the capability for real-time configurations to customize different software systems. An objective quantitative analysis is required to provide better software systems that satisfy user needs in the long term. This analysis is a narrow research area with user logging data and Machine Learning (ML). There is no standard automatic objective identification tool for UE with ML where the score predictions will help the software developers and User Experience (UX) designers to improve LMS software over time after corrections. ML embedded in UE emerged from the lack of analytic evidence in usability items' relevance with limited works [8]–[10].

Therefore, we propose a hybrid approach based on user logging data, usability questionnaire answers, and ML techniques to predict usability scores for the Moodle LMS to promote student engagement and satisfaction. Identifying the features that influence predictive capability in LMSs with ML techniques is crucial for enhancing UX and maximizing user goal achievement. We evaluated this approach in a case study, with data collected from undergraduate students in a public university in the United States. The students follow three tasks in the Moodle LMS, specifically a quiz, discussion forum, and assignment on topics relevant to software engineering. The preliminary results are promising, showing that the best performance of average mean square error and root mean square error among ML algorithms is Random Forest. Our

approach provides a potential strategy for UE, which would help researchers, software developers, and UX designers create LMS software systems that satisfy their users' needs.

## II. Related Work

The subjective approach in UE methods in software methods is mostly through questionnaires, user testing, and heuristics. Questionnaires are the most popular strategy such SUS (System Usability Scale) [11]–[14], After-Scenario Questionnaire (ASQ), Post-Study System Usability Questionnaire (PSSUQ) [15], User Experience Questionnaire (UEQ) [13], USE (Usefulness, Satisfaction, and Ease of Use) [16], [17], Computer System Usability Questionnaire (CSUQ) [18], Usability Metric for User Experience Lite (UMUX-Lite) [19], UseLearn checklist [8], among others. Non-standard questionnaires have been created to address researcher's needs for different studies such as the e-learning platform Blackboard [20] on user student experience, UniStudium [21] (e-learning interface for learning analytics), Context-Aware Mobile Learning System (CAMLS) [22], academic websites [23], and an e-learning platform [24] for a programming course. SUS has been adopted in different e-learning systems [11]–[13] as well as higher education levels [14]. Other questionnaires such as UseLearn checklist [8], USE [17], ASQ [15], PSSUQ [15], CSUQ [18], and UMUX-Lite [19] have been tested on adaptive [25], and non-adaptive platforms (Coursera, OpenLearning, Moodle, and Open Education).

In user testing [5], users follow a set of tasks specific to the software system. User testing has been used in e-learning systems from various universities around the world [16], [23], [24]. User testing [11], [14], [25] takes place with different users following tasks according to the software system. In AdaptLearn LMS [26], undergraduate students followed the task of reviewing a module (adaptive or non-adaptive condition). Other tasks, downloading course materials, searching course schedules, setting up registration, and creating class schedules were accomplished by participants on myCourseVille LMS [15]. To help students figure out solutions, usability testing is also done on the user interface (UI) [18] design on Massive Open Online Course (MOOC) platforms to promote user satisfaction [13], [19]. By bridging the gap between LMS design and usability evaluation, the major objective is to promote students' learning and fix usability problems.

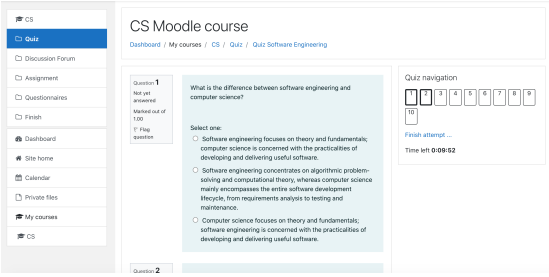
Heuristics [27]–[30] are a set of guidelines where experts evaluate software's usability. The different guidelines are Nielsen's heuristics, Kujala's heuristics, Shneiderman's "Eight Golden Rules", Gerhardt-Powals' cognitive engineering principles, etc. Heuristics have been employed in software systems like digital learning technology prototypes for monitoring intracranial pressure [31], the University of Hong Kong Libraries' mobile website [32], and an e-learning system [33] from a Nigerian case study. In this manner, heuristics [31] find usability issues in the e-learning system such as the color header and footer of the screen. Heuristics also detect [32] usability issues in regards to 1) consistency and standards, 2) flexibility and efficiency, and 3) a better approach for error recovery. The results [33] are capable of better improvement in student learning optimal support.

Mostly questionnaires, user testing, and heuristics have been used in LMSs. Although these UE methods have been broadly utilized, there are challenges related to embedded bias, time investment, and lack LMSs adaptability after live updates. A potential alternative is an objective evaluation limited to logging user data, analytical tools, and ML. For example, Harrati et al. [11] gathered estimation metrics from Moodle (clicks, task duration, completion rate, etc). Factors like session time and learnability score were applied in Fenu et al. [34] work as an analytics tool for usability evaluation interface. ML embedded in UE is bound to a few works: ML evaluation for LMS Moodle [8], a semi-automated framework for social networks [9], and usability-user experience (UUX) issues prediction [10]. The first ML method among UE was from Oztekin et al. [8]. They collected data from a biology course in Moodle and UseLearn checklist answers. The methodology considers different ML algorithms - Neural Networks, Support Vector Machine (SVM), Linear Regression, and Decision Trees (DT) - and severity index. The input was quantitative UseLearn answers and the output was the overall usability. The researchers found that the predicted overall usability and severity index helped to improve the Moodle course and find inconsistencies in the platform - lack of information and personalized experience. Souza Santos et al. [9] proposed a framework with user logging data and ML. The users follow tasks such as "sign up" on social network websites with usability issues: Perspective, Social-Network, and Love-Social. In this context, the user logs help to extract the attributes: URL, duration, event type, Cascading Style Sheets (CSS) path, total duration, event-type specific, and Document Object Model (DOM) object. These attributes were the input into the ML models (DT, Logistic Regression, SVM, and Random Forest) to predict usability smells. The findings were that binary classification between smells (task and action smells) has predictive potential, and multi-class classification was not possible (small data set and class imbalance). Finally, Bakiu and Guzman [10] classify UUX issues from software and video game user reviews. They applied prepossessing techniques on the sentences' reviews to be the input into SVM. This helps to predict specific UUX issues to improve the software. However, there are still open questions about ML embedded in usability evaluation in software systems. These include: which interactions (features) between a user and a LMSs are required to predict positive overall usability and which issues in software systems usability can be predicted with ML. An approach that we proposed considering subjective quantitative answers from questionnaires and objective user logging data embedded in ML as a potential automatic detection tool for UE.

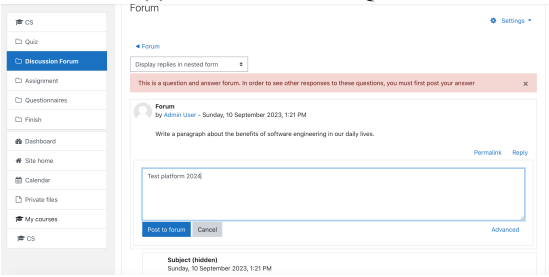
## III. Methods

### A. Moodle course

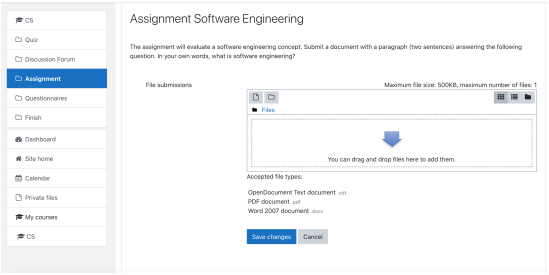
Moodle [35] is a LMS for course management, assessment, and learning for K-12 and college students. It was developed




(a) User interface for Quiz.




(b) User interface for Discussion forum.




(c) User interface for Assignment.

Fig. 1: Moodle course tasks.

in PHP by Martin Dougiamas on 2002, and is composed of different components including Quiz, Forum, Assignment, Questionnaire, etc. The three tasks as seen in Figure 1 are: submit a Quiz, send a paragraph in a simple Forum, and add a file for the Assignment. At the end of the three tasks, two questionnaires were applied to measure the usability of this software system, the System Usability Scale (SUS), and some questions from UseLearn. The three tasks and the two questionnaires are sequential. The student needs to finish each activity to unlock the next one.

*1) Quiz:* The quiz is composed by 10 shuffled questions about software engineering questions. The quiz is multiple choice with three possible answers as radio buttons. The navigation method is sequential with one attempt allowed. The time constraint is 10 minutes.

*2) Discussion Forum:* The discussion forum is open to the students based on a question/answer forum basis. The post answer should be relevant to the following statement: "Write a paragraph about the benefits of software engineering in our daily lives". The student can reply once to this question.

*3) Assignment:* The assignment evaluates a software engineering concept. The student is required to submit a document with a paragraph answering the following question: "What is software engineering?". The student can submit a maximum of one file and the feedback type is deactivated.

### *B. Subjective approach*

The Moodle course contained questions related to the questionnaires SUS and UseLearn as a subjective approach. The Moodle component was "Questionnaire" and the students can respond to this questionnaire once. They give answers to the SUS [36] for low-level usability dimensions (efficiency, effectiveness, and satisfaction) and some questions from the UseLearn checklist [37] for high-level usability dimensions (error prevention, consistency & functionality, and course management). A change in high usability dimensions will affect the low-level dimensions in the LMS. For example, a button that is not working will directly affect the user perception in terms of efficiency, effectiveness, and satisfaction. SUS is just an overall score about the system's usability without the reasons behind it. Therefore, a more detailed insight is needed in the high-level dimensions to achieve an efficient and reliable LMS.

*1) System Usability Scale (SUS):* SUS is a questionnaire with total score ranging from 0 to 100. It is composed of 10 statements with positive and negative questions on a 5-point Likert scale. Scores from each statement will range from 1 (strongly disagree) to 5 (strongly agree). SUS is calculated differently if the questions are even or odd. The score contribution for odd questions is the scale position minus 1. Five minus the score contribution is for even questions. The sum from all the scores is multiplied by 2.5 to obtain the overall value of SUS.

*2) UseLearn checklist:* The UseLearn checklist evaluates specific usability dimensions of the Moodle course. The dimensions of UseLearn are based on 36 questions on a 5-point Likert scale. Scores from each statement will range from 1 (strongly disagree) to 5 (strongly agree). The highest score 5 means that this dimension does not have issues. In this study, three dimensions were selected: 1) error prevention, 2) consistency and functionality, and 3) course management. Each dimension is composed of three questions.

### *C. Objective approach*

UE based on an objective approach evaluation considers quantifiable metrics on software systems and ML. The metrics are from user interaction with the system (clicks, time, keystrokes, etc.). The user is unaware of these interactions. These interactions and quantitative subjective answers from questionnaires are relevant to ML learning techniques for usability score predictions as an automatic detection tool.

*1) Log analysis:* Log analysis refers to records collected while the user is using the system. These records [11], [34], [38] collect metrics such as task time, clicks, keystrokes, cursor distance, and completion rate.

*2) Linear Regression:* Linear regression can be univariate or multivariate [39]. In the simple case scenario, univariate means having an input $x$ that fits into a straight line $y$. The representation is $y = w_1x + w_0$, where $w_0$ and $w_1$ are regression

coefficients. The term $y$ changes by changing the regression coefficients. In a multivariate linear regression problem, the example $x_j$ is an $n$-element vector.

*3) Decision Tree:* A decision tree [40] is an algorithm developed by Leo Breiman known as the Classification and Regression Tree (CART). A decision tree [39] is composed of nodes and branches through recursive partitioning. The input attribute corresponds to an internal node ($A_i$) and the attribute values are in the branches of each node ($A_i$=$v_{ik}$). The leaves (end branches) are the decision of the tree. The decision tree chooses the split to minimize the outcome impurity within each sub-partition. The impurity is measured by Gini impurity (classification) [41] or squared deviations from the mean (regression).

*4) Random Forest:* Random forest [42] is a class of ensemble methods (combining multiple individual models) designed for decision trees. Each tree training is through a random sample of the training data with replacement (bagging). A random subset of features is applied for the tree splitting. This algorithm [40] chooses the variable and split point by minimizing a criterion such as Gini impurity or squared deviations from the mean. The final prediction is found by averaging the predictions of the individual trees in the forest.

*5) Artificial Neural Networks:* Artificial neural networks [39] are based on the idea of how neurons are connected in the human brain. A set of neurons creates a graph network, where each node is a neuron connected through a link. The features from the dataset serve as the input in the neural networks to make predictions for classification or regression problems. The architectures are single and multilayer feed-forward neural networks.

## IV. Experimental Set-up

### A. Data collection and pre-processing

We recruited 183 undergraduate students' emails through Google Forms. A total of 88 students logged into Moodle. Seventy students were included as part of the study. Eighteen students were excluded for several reasons: logging into the system multiple times, exceeding 40 minutes in the study, uploading unrelated files to the task, posting irrelevant answers to the forum, and not answering the usability questionnaires.

*1) Feature extraction:* The data was saved in MySQL, a database from Moodle. The features extracted as input to the supervised ML algorithms are presented in Table II. The features were chosen based on prior literature review, actions, and targets of the respective components, except the session completion component "core" as seen in Table I.

*2) Target extraction:* The targets are four usability scores as seen in Table III: SUS score, error prevention (EP) score, consistency and functionality (CF) score, and course management (CM) score.

*3) Data standardisation:* Standardization is a common step in preprocessing data for ML algorithms. This technique produces superior performance in these algorithms by reducing inconsistencies related to different scales of the features. The standardization technique was applied to scale nine features.

TABLE I: Components action and target description.

| Component | Action Start | Action End | Target Start | Target End |
|---|---|---|---|---|
| mod_quiz | started | submitted | attempt | attempt |
| mod_forum | viewed | created | course_ module | post |
| mod_assign | viewed | submitted | course_ module | assessable |
| core | loggedin | loggedout | user | user |

TABLE II: Features description.

| Feature | Type | Description |
|---|---|---|
| time quiz | Numerical | seconds to finish quiz task |
| time forum | Numerical | seconds to finish forum task |
| time assignment | Numerical | seconds to finish assignment task |
| grade quiz | Numerical | quiz points (1 correct answer, 0 incorrect answer) |
| word count per forum response | Numerical | number of words in the post |
| file size | Numerical | kilobytes in the assignment file |
| file type | Numerical | 1 (PDF file) or 0 (other file type) |
| clicks module quiz visited | Numerical | number of clicks to finish quiz task |
| clicks module forum visited | Numerical | number of clicks to finish forum task |
| clicks module assignment visited | Numerical | number of clicks to finish assignment task |

The features include: time quiz, time forum, time assign, grade quiz, word count per forum response, file size, clicks module quiz, clicks module forum, and clicks module assignment. The formula to rescale each feature is based on the mean and standard deviation. The formula is:

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

where $z$ is the standardized feature, $x$ is the original feature, $\mu$ is the mean of the feature, and $\sigma$ is the standard deviation of the feature.

TABLE III: Target description.

| Target | Type | Description |
|---|---|---|
| SUS score | Numerical | usability score between 0 and 100 |
| EP score | Numerical | usability score between 1 and 5 |
| CF score | Numerical | usability score between 1 and 5 |
| CM score | Numerical | usability score between 1 and 5 |

*4) Data normalization:* Normalization is a preprocessing technique in ML to ensure consistency and numerical stability between the features (scale between 0 and 1). It enhances overall performance in these algorithms, where the targets are normalized. The formula is:

$$x_{\text{norm}} = \frac{x - \min(X)}{\max(X) - \min(X)} \tag{2}$$

where $x_{\text{norm}}$ is the normalized target, $x$ is the original target value, $\min(X)$ is the minimum value of the target in the dataset, and $\max(X)$ is the maximum value of the target in the dataset.

### B. Supervised Machine Learning algorithms

We propose a methodology to predict usability based on ML techniques. The number of features extracted from the three tasks are used as input in the ML algorithms. The targets are four usability scores: SUS score, EP score, CF score, and CM score. The small dataset collected was applied to ML-supervised algorithms using a three-fold cross-validation procedure to predict the usability scores and avoid overfitting problems. The supervised ML algorithms chosen are linear regression, decision trees, random forest, and neural networks. The hyperparameters in the ML algorithms were found through the grid search algorithm [43].

### C. Evaluation Metrics

The evaluation metrics are Mean Square Error (MSE) and Root Mean Square Error (RMSE) to verify the performance and fitness in regression models. MSE considers the error as the square difference between the predicted and the actual target. RMSE is the square root of MSE, where the unit of measurement is the same as the target. The formulas are depicted as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}(\hat{Y}_{ij} - Y_{ij})^2 \tag{3}$$

$$\text{RMSE} = \sqrt{MSE} \tag{4}$$

where $n$ is the number of records, $m$ is the number of outputs, $\hat{Y}_{ij}$ is the predicted value for the $i$-th record and $j$-th output, and $Y_{ij}$ is the target for the $i$-th record and $j$-th output.

## V. Results

We proposed three experiments with different combinations of the ten features to predict usability scores as a hybrid approach between logging user data, ML, and questionnaires. The insight of these experiments is to show the capacity of supervised ML algorithms to predict usability scores in LMSs and the features involved. Each experiment follows the three cross-validation [44] procedure as a valid strategy to ensure performance in regression problems by the mean and standard deviation of the selected metrics MSE and RMSE [45], [46]. Cross-validation [47] in a small dataset is a robust metric to estimate model performance. Another consideration is that the research is not a comparative study but a study about which features can predict usability evaluation scores through ML algorithms. The current research about usability and ML uses performance evaluation metrics depending on classification (accuracy and F1 score) or regression (MSE and RMSE) [8]–[10]. To demonstrate the ability of ML models to predict usability scores, we considered MSE and RMSE as performance metrics [48].

TABLE IV: Parameter settings for ML algorithms.

| ML algorithm | Configuration |
|---|---|
| Linear Regression | fit intercept: True |
| Decision Trees | maximum depth: None, minimum samples leaf: 10, minimum samples split: 2 |
| Random Forest | maximum depth: None, minimum samples leaf: 10, minimum samples split: 2, estimators: 100 |
| Neural Networks | activation: relu, alpha: 0.01, hidden layer sizes: (100, 50, 25) learning rate: 0.001, solver: sgd |

### A. Experiments with 6 features

In the first experiment, the best ML configuration found by grid search is presented in Table IV. Based on domain knowledge, we selected six features applicable for usability detection: time quiz, time forum, time assignment, clicks module quiz, clicks module forum, and clicks module assignment.

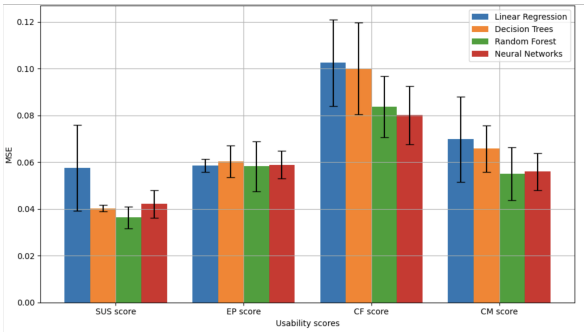



Fig. 2: MSE from the usability scores with different Machine Learning Algorithms with 6 features selected.

The model prediction results are seen in Figure 2. The lowest average MSE among the ML algorithms is the SUS score (0.0441 $\pm$ 0.0076) and the highest is the CF score (0.0916 $\pm$ 0.0159). SUS gives an insight into the overall usability of the software system without specific usability dimensions for LMS. For this reason, the other three scores - CM, EP, and CF - provide usability dimensions specifically for LMS. The UseLearn scores ordered by the average from the ML algorithms are EP score (0.0590 $\pm$ 0.0065), CM score (0.0617 $\pm$ 0.0119), and CF score (0.0916 $\pm$ 0.0159).

Random Forest is the best strategy for usability score predictions with overall MSE 0.0583 $\pm$ 0.0099 and RMSE 0.2381 $\pm$ 0.0200. The lowest MSE is for Random Forest on the SUS score. Among usability prediction scores, Neural Networks demonstrate notable performance. Linear Regression is the highest MSE and RMSE, meaning that there is no linear relationship between the predictor variables and the target. RMSE proves that in experiment one, Random Forest is the strategy to fit the data to a specific model as shown in Figure 3. On the other hand, Linear Regression continues to exhibit underperformance, particularly in terms of the CF score.

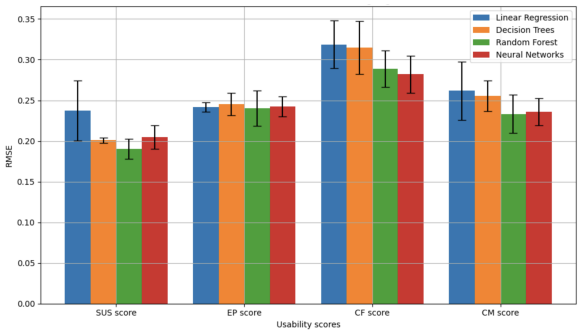



Fig. 3: RMSE from different Machine Learning Algorithms with 6 features selected.

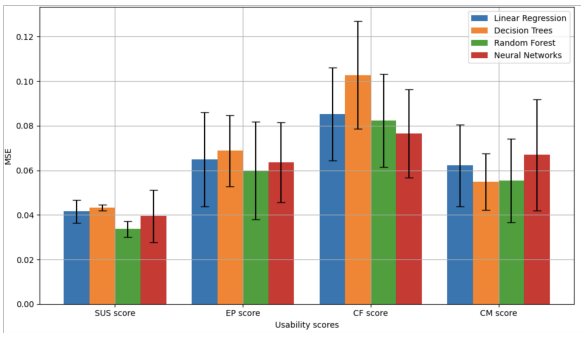



Fig. 4: MSE from different Machine Learning Algorithms with 7 features selected.

### *B. Experiments with 7 features*

In the second experiment, the best ML configuration found by grid search is presented in Table V. We deleted the features related to module clicks and added additional features from the specific tasks to verify if the performance improves. The seven features are: time quiz, time forum, time assignment, grade quiz, word count per forum response, file size, and file type.

TABLE V: Parameter settings for ML algorithms.

| ML algorithm | Configuration |
|---|---|
| Linear Regression | fit intercept: True |
| Decision Trees | maximum depth: None, minimum samples leaf: 10, minimum samples split: 2 |
| Random Forest | maximum depth: 5, minimum samples leaf: 2, minimum samples split: 5, estimators: 50 |
| Neural Networks | activation: tanh, alpha: 0.01, hidden layer sizes: (100) learning rate: 0.01, solver: sgd |

The results from this experiment are seen in Figure 4. The lowest average MSE among the ML algorithms is the SUS score (0.0395 $\pm$ 0.0055) and the highest is the CF score (0.0867 $\pm$ 0.0215). This exhibits the same relevance in those terms as experiment one. The UseLearn scores ordered by the average from the ML algorithms are CM score (0.0599 $\pm$ 0.0187), EP score (0.0643 $\pm$ 0.0193), and CF score (0.0867 $\pm$ 0.0215).

In experiment two, Random Forest obtains the lowest overall MSE (0.0579 $\pm$ 0.0162) and RMSE (0.2353 $\pm$ 0.032) from all scores. The lowest MSE on Random Forest is 0.0337 $\pm$ 0.0036 on the SUS score. Neural Networks are able to predict usability prediction scores as well. On this occasion, Decision Trees demonstrate poor fit to the data, evidenced by their lowest MSE when compared to other techniques. Figure 5 depicts RMSE from all the ML techniques, where Random Forest has the lowest RMSE results in terms of SUS score (0.1833 $\pm$ 0.0099), CM score (0.2323 $\pm$ 0.0381), and EP score (0.2410 $\pm$ 0.0428).

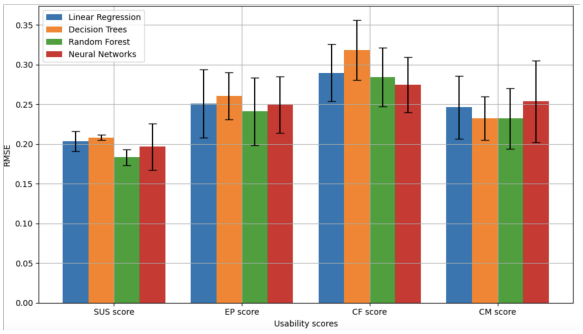



Fig. 5: RMSE from different Machine Learning Algorithms with 7 features selected.

### *C. Experiments with 10 features*

In the last experiment, we added the deleted features from experiment two. The ten features in total are time quiz, time forum, time assignment, grade quiz, word count per forum response, file size, file type, clicks module quiz, clicks module forum, and clicks module assignment. These features were selected as input in each algorithm with the configuration presented in Table VI.

TABLE VI: Parameter settings for ML algorithms.

| ML algorithm | Configuration |
|---|---|
| Linear Regression | fit intercept: True |
| Decision Trees | maximum depth: None, minimum samples leaf: 10, minimum samples split: 2 |
| Random Forest | maximum depth: None, minimum samples leaf: 2, minimum samples split: 2, estimators: 50 |
| Neural Networks | activation: relu, alpha: 0.01, hidden layer sizes: (50, 25, 10) learning rate: 0.01, solver: sgd |

The model prediction results are seen in Figure 6. The lowest average MSE among the ML algorithms is the SUS score (0.0456 $\pm$ 0.0077) and the highest is the CF score (0.0897 $\pm$ 0.0183). The order among the UseLearn scores considering the average from the ML algorithms is CM score (0.0604 $\pm$ 0.0176), EP score (0.0691 $\pm$ 0.0161), and CF score (0.0897 $\pm$ 0.0183).

Random Forest is the best strategy for usability predictions with overall MSE 0.0577 $\pm$ 0.0151 and RMSE 0.2356 $\pm$ 0.0299. The lowest MSE on Random Forest is 0.0364 $\pm$ 0.0047 on the SUS score. Neural Networks excel for usability prediction scores as Random Forest. In this experiment, Linear regression does not fit the data well as experiment one. Figure 7 depicts RMSE from all the ML techniques, where Random Forest has the lowest RMSE results in terms of SUS score (0.1905 $\pm$ 0.0121), CM score (0.2297 $\pm$ 0.0358), and EP score (0.2355 $\pm$ 0.0393).

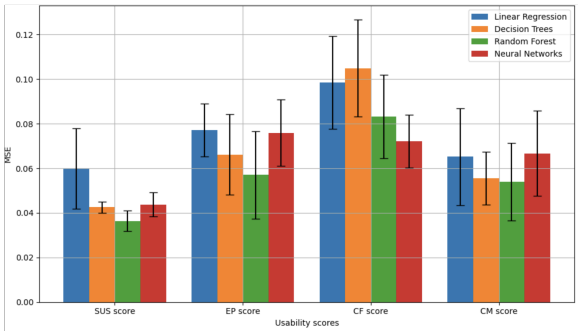



Fig. 6: MSE from different Machine Learning Algorithms with 10 features selected.

The results of MSE and RMSE show that the seven features provide better results in the SUS score. In this context, there is a potential for usability prediction scores based on quantitative questionnaire answers, user interactions, and ML algorithms.

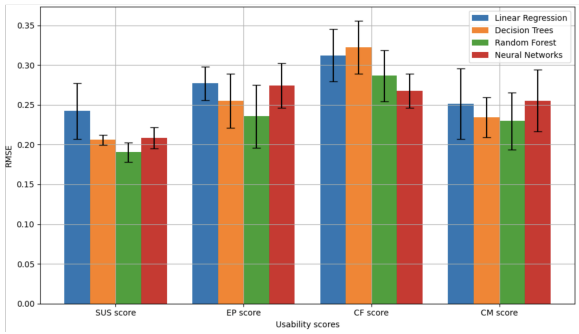



Fig. 7: RMSE from different Machine Learning Algorithms with 10 features selected.

## VI. Discussion

The results of this study show a prospective adaptable UE alternative for usability score prediction in LMSs. In the three experiments, ML algorithms obtained the lowest MSE and RMSE of the SUS scores in contrast with the other metrics. SUS score offers software developers and UX designers insight into the system's usability for users (undergraduate students). ML models fit the features more suitable for SUS score prediction (low-level usability dimensions) than the UseLearn checklist (high-level usability dimensions). CF scores perform poorly by the highest RMSE and MSE among the three experiments. This shows that the features selected are not relevant for predicting consistency and functionality between elements in the user interface of the Moodle course, such as consistency between the titles, headers, and icons. The order of UseLearn dimensions is the same in the last two experiments: CM score, EP score, and CF score. ML algorithms have some capability of predicting the CM score, which is relevant for indicating the Moodle course's provision of resources to support online learning. The EP score verifies error prevention measures, ensuring tasks are effortlessly completed. Random Forest achieves the best performance among the experiments. The second experiment with seven features produces the best performance of the SUS score. The features are time quiz, time forum, time assignment, grade quiz, word count message post, file size, and file type. The module clicks features from each task were eliminated. This demonstrates that utilizing module clicks is not the best strategy for feature selection in the SUS score prediction.

### A. ML techniques performance

Linear regression struggles to find a linear relation between the features and the targets. The worst strategy is in the third experiment, where ten features are selected. Its SUS score improves when module clicks are eliminated, and features of each task are chosen, showing some linear relationship between the inputs and the targets. It is still one of the lowest performances among the three other models.

Decision trees enhance RMSE and MSE in experiments one and three. The hyperparameters in the three experiments produce the same values by following a pre-pruning technique (removing unnecessary branches). This depth restricts the number of tree splits from the root node to the leaf nodes. The depth found was "None" meaning that nodes expand until the leaves are pure or the leaves contain less than the minimum samples to split. The value was two samples to split an internal node to capture patterns correlating with the small dataset. The minimum sample leaf is ten to be a leaf node to prevent overfitting with few samples. Decision Trees in experiment two do not produce favorable usability score prediction with the highest RMSE and MSE in some scores.

Random Forest is the best algorithm for usability prediction in the three experiments. Random Forest is an ensemble method that aggregates predictions of multiple decision trees as estimators. The rest of the hyperparameters follow the same criteria as an individual tree with the maximum depth, minimum samples split, and minimum samples leaf. In experiment two, the prediction of multiple trees followed criteria with the maximum depth being five. The number of branches decreased, showing the minimum samples for split is five and the minimum samples for leaf is two. As a result, multiple trees produce superior capabilities of RMSE usability prediction for SUS score (0.1833 $\pm$ 0.0099), CM score (0.2323 $\pm$ 0.0381), and EP score (0.2410 $\pm$ 0.0428).

Neural networks can predict usability scores in all the experiments. The second experiment can predict CF scores in contrast with the other ML models. The hidden layer is

just one layer with 100 neurons. Therefore, neural networks find hidden patterns in the seven features. The learning rates in the experiments range from 0.001 to 0.01. The lowest is in the first experiment where the optimization process converges slowly.

### B. Performance improvement

The ML algorithms in this research explain a modest performance, albeit not entirely satisfactory. MSE and RMSE in the usability scores show higher values of $0.0337 \pm 0.0036$ and $0.1833 \pm 0.0099$ in all the experiments. A possibility to improve performance is the feature engineering and selection to avoid overfitting (no generalization of unseen samples). The features were chosen based on domain knowledge from previous research where user logging data was used [8]–[10], [34]. In this context, one common feature is the time of each task which aligns with the results from the experiment two. However, in the previous state-of-the-art works, there is no hybrid methodology where user logging data, quantitative questionnaire answers, and ML are employed for usability score prediction. Another factor is how the module clicks were extracted to be considered as a feature with MySQL queries and Table I. A plug-in can be developed and installed on Moodle to extract the module clicks of each task. The features specific to each task in the quiz (grade), forum (word count), and assignment (file size, and file type) are applicable just for usability score predictions on LMS. Features such as erroneous clicks to complete a task can be picked as an alternative.

Erroneous clicks mean which other components the student selected outside the one from the specific task. Linear Regression and Decision Trees make it easier to interpret how a usability score is determined. Random Forests and Neural Networks, although often considered "black boxes" due to their increased complexity, excel at capturing and characterizing nonlinear relationships and interactions in data. Random Forest produces the highest performance in all the experiments given that multiple trees produce an average between them in terms of RMSE and MSE. Neural Networks show progress in performance, which means that the data forms a nonlinear relationship in the data.

### C. Limitations

The limitations of this work are the relatively small size of the dataset used, the case study experiment was virtual, and the generalization to unseen tasks. New data collected from students will increase the performance of the ML algorithms and neural networks could find hidden patterns with its predictive abilities. A controlled environment in person for the research will be beneficial to analyze student concentration on the tasks without any external stimulus.

## VII. Conclusion and Future work

### A. Conclusion

We presented a hybrid approach based on user logging data, quantitative usability questionnaire answers, and ML techniques to predict usability scores for an LMS to promote student engagement and satisfaction. We evaluated this approach in a remote case study, with data collected from undergraduate students. The 70 students follow three tasks in the Moodle LMS, specifically a quiz, discussion forum, and assignment on topics relevant to software engineering. We applied different features in the ML algorithms Linear Regression, Decision Trees, Random Forest, and Neural Networks in three experiments. The second experiment with seven features produces the best performance of average MSE and RMSE in the SUS score. Random Forest achieves the best performance among the experiments. The feature module clicks are not suitable for usability SUS score prediction. The results are promising with alternatives for improvements for better performance of SUS and UseLearn scores prediction. The SUS score prediction will provide a general insight into the LMS's usefulness. The UE specifically for LMS through UseLearn prediction scores will give information about how well the Moodle course is usable in terms of learning resources (CM), error prevention and task accomplishment (EP), and consistency between the visual elements (CF). This approach contributes to the computing education field by providing a predictive tool for usability scores to improve the student learning experience by potential LMS customization.

### B. Future work

The module clicks can be extracted differently. An alternative is to develop and install a plug-in inside Moodle and extract the module clicks of each task. Another feature can be picked, for example, erroneous clicks. Additional data collection, new feature extraction, hyperparameters diversification, and new tests for unseen tasks would improve the ML performance. Unseen tasks related to accessing course materials, engagement, and participation in the LMSs.

At the same time, the methodology can be extended to other LMSs, for instance, Blackboard or Canvas. The Moodle course initially centered on software engineering concepts, yet can be expanded to other disciplines where LMSs are employed. Usability score prediction would provide customized LMSs according to the student's needs.

## References


[1] N. Bevan, J. Carter, and S. Harker, "ISO 9241-11 Revised: What Have We Learnt About Usability Since 1998?" in *Proceedings of the 17th International Human-Computer Interaction Conference*, Los Angeles, CA, USA, 2015, pp. 143–151.

[2] J. Nielsen, *Usability Engineering*, 1st ed. San Francisco, CA, USA: Morgan Kaufmann, 1994.

[3] ISO/IEC, "25010:2023 systems and software quality requirements and evaluation," accessed: 2023-09-30. [Online]. Available: https://www.iso.org/standard/78176.html

[4] S. Riihiaho, "Usability Testing," *The Wiley Handbook of Human Computer Interaction*, vol. 1, pp. 255–275, 2018.

[5] F. Paz and J. A. Pow-Sang, "Usability Evaluation Methods for Software Development: A Systematic Mapping Review," in *Proceedings of the 8th International Conference on Advanced Software Engineering and its Applications (ASEA)*, Jeju, South Korea, 2015, pp. 1–4.

[6] K. Ishaq, F. Rosdi, N. Zin, and A. Abid, "Heuristics and Think-aloud Method for Evaluating the Usability of Game-based Language Learning," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 11, pp. 311–324, 2021.

[7] W. Nakamura, E. De Oliveira, and T. Conte, "Usability and user experience evaluation of learning management systems a systematic mapping study," in *Proceedings of the 19th International Conference on Enterprise Information Systems*, Porto, Portugal, 2017, pp. 97–108.

[8] A. Oztekin, D. Delen, A. Turkyilmaz, and S. Zaim, "A machine learning-based usability evaluation method for eLearning systems," *Decision Support Systems*, vol. 56, pp. 63–73, 2013.

[9] F. de Souza Santos, M. Vinícius Treviso, S. P. Gama, and R. P. de Mattos Fortes, "A framework to semi-automated usability evaluations processing considering users' emotional aspects," in *Proceedings of the 24th Human-Computer Interaction International Conference*, Virtual Event, 2022, p. 419–438.

[10] E. Bakiu and E. Guzman, "Which feature is unusable? detecting usability and user experience issues from user reviews," in *Proceedings of the 25th International Requirements Engineering Conference Workshops (REW)*, Lisbon, Portugal, 2017, pp. 182–187.

[11] N. Harrati, I. Bouchrika, A. Tari, and A. Ladjailia, "Exploring user satisfaction for e-learning systems via usage-based metrics and system usability scale analysis," *Computers in Human Behavior*, vol. 61, pp. 463–471, 2016.

[12] A. C.-P. Anabel Martin-Gonzalez and V. Uc-Cetina, "Usability evaluation of an augmented reality system for teaching euclidean vectors," *Innovations in Education and Teaching International*, vol. 53, no. 6, pp. 627–636, 2016.

[13] N. P. I. R. Devy, S. Wibirama, and P. I. Santosa, "Evaluating user experience of English learning interface using user experience questionnaire and system usability scale," in *Proceedings of the 1st International Conference on Informatics and Computational Sciences*, Semarang, Indonesia, 2017, pp. 101–106.

[14] K. Abuhlfaia and E. de Quincey, "Evaluating the usability of an e-learning platform within higher education from a student perspective," in *Proceedings of the 3rd International Conference on Education and E-Learning*, New York, NY, USA, 2020, p. 1–7.

[15] N. Phongphaew and A. Jiamsanguanwong, "Usability Evaluation on Learning Management System," in *Proceedings of the 2017 International Conference on Usability and User Experience*, Los Angeles, CA, USA, 2018, pp. 39–48.

[16] H. Pangestu and M. Karsen, "Evaluation of usability in online learning," in *Proceedings of the International Conference on Information Management and Technology*, Bandung, Indonesia, 2017, pp. 267–271.

[17] D. Hariyanto, M. Triyono, and T. Köhler, "Usability evaluation of personalized adaptive e-learning system using USE questionnaire," *Knowledge Management and E-Learning*, vol. 12, no. 1, pp. 85–105, 2020.

[18] H. Azami and R. Ibrahim, "Development and evaluation of massive open online course (MOOC) as a supplementary learning tool: An initial study," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 7, pp. 532–537, 2019.

[19] O. Korableva, T. Durand, O. Kalimullina, and I. Stepanova, "Usability testing of MOOC: Identifying user interface problems," in *Proceedings of the 21st International Conference on Enterprise Information Systems*, Crete, Greece, 2019, pp. 468–475.

[20] A. Alshehri, M. Rutter, and S. Smith, "Assessing the relative importance of an e-learning system's usability design characteristics based on students' preferences," *European Journal of Educational Research*, vol. 8, no. 3, pp. 839–855, 2019.

[21] V. Franzoni, A. Milani, P. Mengoni, and F. Piccinato, "Artificial intelligence visual metaphors in e-learning interfaces for learning analytics," *Applied Sciences*, vol. 10, no. 20, pp. 1–25, 2020.

[22] A. Pensabe-Rodriguez, E. Lopez-Dominguez, Y. Hernandez-Velazquez, S. Dominguez-Isidro, and J. De-la Calleja, "Context-aware mobile learning system: Usability assessment based on a field study," *Telematics and Informatics*, vol. 48, 2020.

[23] A. Muhammad, A. Siddique, Q. Naveed, U. Khaliq, A. Aseere, M. Hasan, M. Qureshi, and B. Shehzad, "Evaluating usability of academic websites through a fuzzy analytical hierarchical process," *Sustainability*, vol. 13, no. 4, pp. 1–22, 2021.

[24] D. Nariman, "Impact of the interactive e-learning instructions on effectiveness of a programming course," in *Proceedings of the 14th International Conference on Complex, Intelligent and Software Intensive Systems*, Lodz, Poland, 2021, pp. 588–597.

[25] M. Alshammari, R. Anane, and R. Hendley, "Design and usability evaluation of adaptive e-learning systems based on learner knowledge and learning style," in *Proceedings of the 15th International Conference INTERACT*, Bamberg, Germany, 2015, pp. 584–591.

[26] M. Alshammari and R. Anane, "Usability and effectiveness evaluation of adaptivity in e-learning systems," in *Proceedings of the International Conference on Human Factors in Computing Systems*, San Jose, CA, USA, 2016, pp. 2984–2991.

[27] K. Ishaq, F. Rosdi, N. Zin, and A. Abid, "Heuristics and Think-aloud Method for Evaluating the Usability of Game-based Language Learning," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 11, pp. 311–324, 2021.

[28] B. A. Zardari, Z. Hussain, A. A. Arain, W. H. Rizvi, and M. S. Vighio, "QUEST e-learning portal: applying heuristic evaluation, usability testing and eye tracking," *Universal Access in the Information Society*, vol. 20, no. 3, 2021.

[29] B. A. Kumar and M. S. Goundar, "Usability heuristics for mobile learning applications," *Education and Information Technologies*, vol. 24, no. 2, pp. 1819–1833, 2019.

[30] D. Quiñones, C. Rusu, and V. Rusu, "A methodology to develop usability/user experience heuristics," *Computer Standards & Interfaces*, vol. 59, pp. 109–129, 2018.

[31] L. de Carvalho, Y. Évora, and S. Zem-Mascarenhas, "Assessment of the usability of a digital learning technology prototype for monitoring intracranial pressure," *Revista Latino-Americana de Enfermagem*, vol. 24, 2016.

[32] R. Fung, D. Chiu, E. Ko, K. Ho, and P. Lo, "Heuristic usability evaluation of university of hong kong libraries' mobile website," *Journal of Academic Librarianship*, vol. 42, no. 5, pp. 581–594, 2016.

[33] O. Daramola, O. Oladipupo, I. Afolabi, and A. Olopade, "Heuristic evaluation of an institutional e-learning system: A Nigerian case," *International Journal of Emerging Technologies in Learning*, vol. 12, no. 3, pp. 26–42, 2017.

[34] G. Fenu, M. Marras, and M. Meles, "A learning analytics tool for usability assessment in Moodle environments," *Journal of E-Learning and Knowledge Society*, vol. 13, no. 3, pp. 23–34, 2017.

[35] M. Dougiamas, "Moodle," accessed: 2024-01-01. [Online]. Available: https://moodle.org/

[36] J. Brooke, "SUS: A quick and dirty usability scale," *Usability Evaluation in Industry*, vol. 189, pp. 189 – 194, 1996.

[37] A. Oztekin, Z. J. Kong, and O. Uysal, "UseLearn: A novel checklist and usability evaluation method for eLearning systems by criticality metric analysis," *International Journal of Industrial Ergonomics*, vol. 40, no. 4, pp. 455–469, 2010.

[38] J. W. Castro, I. Garnica, and L. A. Rojas, "Automated Tools for Usability Evaluation: A Systematic Mapping Study," in *Proceedings of the 24th Human-Computer Interaction International Conference*, Virtual Event, 2022, pp. 28–46.

[39] S. J. Russell, P. Norvig, and E. Davis, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2010.

[40] P. Bruce, A. Bruce, and P. Gedeck, *Practical Statistics for Data Scientists*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2020.

[41] S. Suthaharan, *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*, 1st ed. New York, NY, USA: Springer, 2016.

[42] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining*, 2nd ed. New York, NY, USA: Pearson, 2018.

[43] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 1st ed. MIT Press, 2016.

[44] I. Tougui, A. Jilbab, and J. E. Mhamdi, "Impact of the choice of cross-validation techniques on the results of machine learning-based diagnostic applications," *Healthcare Informatics Research*, vol. 27, pp. 189 —- 199, 2021.

[45] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Scientific Reports*, vol. 14, no. 1, 2024.

[46] M. Z. Naser and A. H. Alavi, "Error metrics and performance fitness indicators for artificial intelligence and machine learning in engineering and sciences," *Architecture, Structures and Construction*, vol. 3, no. 4, pp. 499–517, 2023.

[47] S. Raschka, "Model evaluation, model selection, and algorithm selection in machine learning," *Computing Research Repository*, vol. abs/1811.12808, 2020.

[48] M. V. Sebt, Y. Sadati-Keneti, M. Rahbari, Z. Gholipour, and H. Mehri, "Regression method in data mining: A systematic literature review," *Archives of Computational Methods in Engineering*, 2024.