United Arab Emirates University

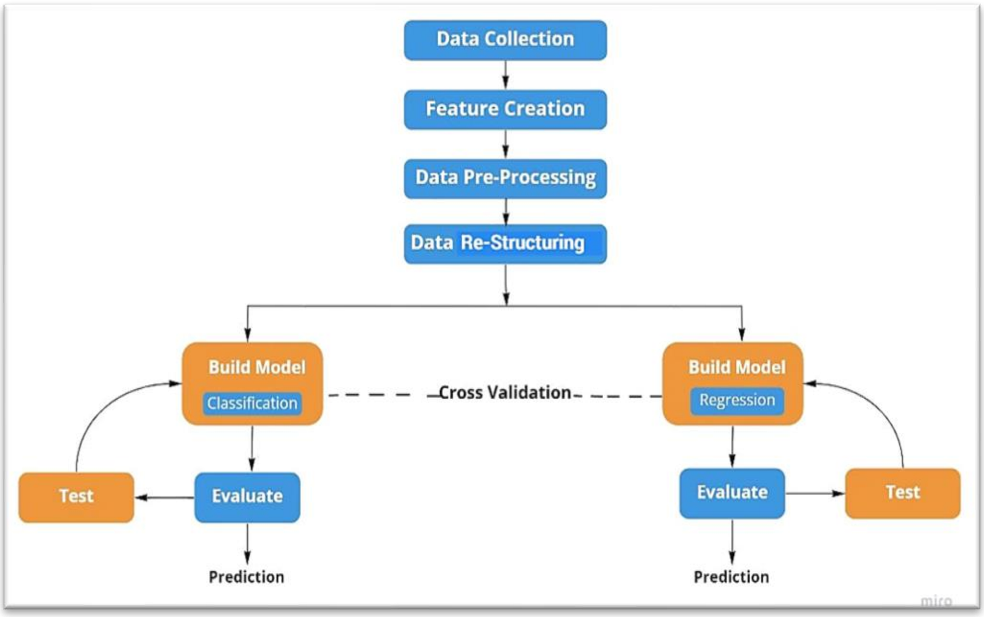
MASTER THESIS NO. 2022:52

College of Information Technology

Department of Information Systems and Security

DETERMINING KNOWLEDGE FROM STUDENT PERFORMANCE PREDICTION USING MACHINE LEARNING

Wala El Rashied Suliman Mohamed



June 2022

United Arab Emirates University

College of Information Technology

Department of Information Systems and Security

DETERMINING KNOWLEDGE FROM STUDENT PERFORMANCE PREDICTION USING MACHINE LEARNING

Wala El Rashied Suliman

This thesis is submitted in partial fulfilment of the requirements for the degree of Master of Science in Information Technology Management

Under the Supervision of Professor Nazar Zaki

June 2022

Declaration of Original Work

I, Wala El Rashied Suliman, the undersigned, a graduate student at the United Arab Emirates University (UAEU), and the author of this thesis entitled "*Determining Knowledge from Student Performance Prediction using Machine Learning*", hereby, solemnly declare that this is the original research work done by me under the supervision of Professor Nazar Zaki, in the College of Information Technology at UAEU. This work has not previously formed the basis for the award of any academic degree, diploma or a similar title at this or any other university. Any materials borrowed from other sources (whether published or unpublished) and relied upon or included in my thesis have been properly cited and acknowledged in accordance with appropriate academic conventions. I further declare that there is no potential conflict of interest with respect to the research, data collection, authorship, presentation and/or publication of this thesis.

Student's Signature: ____________________________ Date: __21/4/2022_____

Approval of the Master Thesis

This Master Thesis is approved by the following Examining Committee Members:

1) Advisor (Committee Chair): Nazar Zaki
   Title: Professor
   Department of Computer Science and Software Engineering
   College of Information Technology
   Signature ____________________ Date 18-Jun-22

2) Member: Elarbi Badidi
   Title: Associate Professor
   Department of Computer Science and Software Engineering
   College of Information Technology
   Signature ____________________ Date 20-Jun-22

3) Member (External Examiner): Farhad Oroumchian
   Title: Professor
   Department of Engineering and Information Sciences
   Institution: University of Wollongong in Dubai, UAE
   Signature Farhad Oroumchian Date 20-Jun-22

This Master Thesis is accepted by:

Dean of the College of Information Technology: Professor Taieb Znati

Signature Taieb Znati Date 23/08/2022

Dean of the College of Graduate Studies: Professor Ali Al-Marzouqi

Signature Ali Hassan Date 23/08/2022

Abstract

Recent years have seen a rapid development in the field of educational data mining (EDM), enhancing the ability to trace student knowledge. Data from intelligent tutoring systems (ITS) have been analyzed and interpreted by multiple researchers seeking to measure students' knowledge as it evolves. Human nature, as well as other factors, makes it difficult to determine whether or not students are knowledgeable. This thesis sets out to examine the level of students' knowledge by predicting their current and future academic performance based on records of their historical interactions. By restructuring data and considering a student perspective, we can gain insight into certain important attributes, their inter-relationships, and the overall effect on performance. The objectives of this study are as follows: (i) to apply machine learning (ML) techniques in order to determine students' knowledge based on their predicted academic performance using student-focused aggregated data as opposed to the usual (problem-focused) data structure; (ii) to predict the academic accuracy of the next student response with the use of ML models. Lastly, (iii) to determine the problem sequence types (delivery patterns) that lead to the best academic performance; and to analyze how these sequence types influence the accuracy of each student's response. Experimental work was carried out using the ASSISTments 2012–2013 dataset as well as ML models. It demonstrated that the proposed approach predicts student performance better than current knowledge tracing (KT) models. The results of the present study confirm the usefulness of classification and regression techniques in capturing greater variance within the data, resulting in more precise predictions.

Keywords: Knowledge tracing, Student performance, Classification, Educational data mining, Intelligent tutoring system.

Title and Abstract (in Arabic)

تحديد المعرفة من توقع أداء الطلاب باستخدام التعلم الآلي

الملخص

شهدت السنوات الأخيرة تطوراً سريعاً في مجال تنقيب البيانات التعليمية، مما أدى إلى تعزيز القدرة على تتبع المعرفة لدى الطالب. تم تحليل البيانات من أنظمة التدريس الذكية وتفسيرها من قبل العديد من الباحثين الذين يسعون إلى قياس معرفة الطلاب أثناء تطورها. تجعل الطبيعة البشرية، بالإضافة إلى عوامل أخرى من الصعب تحديد ما إذا كان الطلاب على معرفة أم لا. تهدف هذه الأطروحة إلى فحص مستوى معرفة الطلاب من خلال التنبؤ بالأداء الأكاديمي الحالي والمستقبلي بناءً على سجلات تفاعلاتهم التاريخية. من خلال إعادة هيكلة البيانات والنظر في منظور الطالب، يمكننا الحصول على نظرة ثاقبة لسمات مهمة معينة وعلاقاتها المتبادلة والتأثير العام على الأداء. أهداف هذه الدراسة هي كما يلي: (1) تطبيق تقنيات التعلم الآلي من أجل تحديد معرفة الطلاب بناءً على أدائهم الأكاديمي المتوقع باستخدام البيانات المجمعة التي تركز على الطالب بدلاً من البيانات المعتادة (التي تركز على المشكلة) بنية؛ (2) للتنبؤ بالدقة الأكاديمية لاستجابة الطالب التالية باستخدام نماذج ML. أخيرًا، (3) تحديد أنواع تسلسل المشكلة (أنماط التسليم) التي تؤدي إلى أفضل أداء أكاديمي؛ ولتحليل كيفية تأثير أنواع التسلسل هذه على دقة استجابة كل طالب. تم تنفيذ العمل التجريبي باستخدام مجموعة بيانات (ASSISTments 2012-2013) بالإضافة إلى نماذج تعلم الآلة (ML)، و أظهر أن النهج المقترح يتنبأ بأداء الطالب بشكل أفضل من نماذج تتبع المعرفة الحالية (KT). تؤكد نتائج الدراسة الحالية فائدة تقنيات التصنيف والانحدار في التقاط تباين أكبر داخل البيانات، مما يؤدي إلى تنبؤات أكثر دقة.

مفاهيم البحث الرئيسية: تتبع المعرفة، نمذجة الأداء، الانحدار، التنقيب عن البيانات التعليمية، نظام التدريس الذكي.

Acknowledgements

First and foremost, praise and thanks are due to Allah, the Almighty, for His blessings throughout my research work which have enabled me to complete this study successfully.

I would like to express my deep and sincere gratitude to my advisor, Professor Nazar Zaki, for his continuous guidance and professional and fruitful advice. His constructive feedback pushed me to sharpen my thinking and brought my work to a higher level. It was a great privilege and honor to work and study under his guidance. He taught me the methodology to carry out and present the research as clearly as possible.

I would also like to thank my committee for guidance, support, and assistance throughout my preparation of this thesis. I would like to thank Professor Elarbi Badidi for all his academic advice and examining the work. Also, my external examiner, Professor Farhad Oroumchian, for his valuable time and effort.

I am extremely grateful to my parents for their endless love, wise counsel, and sympathetic ear. My brothers and sister who have never left my side and are very special. Warm thanks to my precious daughters and husband, for their kindness and patience to complete this research paper. Also, special thanks go to my dearest friend, Samah for her limitless encouragement and massive support along the way. A sincere appreciation and thank to the kind work family I am blessed to have, starting with my managers for their profound support and consideration, my colleagues for their kind-heartful prayers and non-stop motivation.

Dedication

To my beloved parents and precious family

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence
Gaussian NB	Gaussian Naive Bayes
KNN	K-Nearest Neighbor
KT	Knowledge Tracing
LDA	Linear Discriminant Analysis
QDA	Quadratic Discriminant Analysis
RBF	Radial Basis Function
SVM	Support Vector Machine

Chapter 1: Introduction

1.1 Overview

Intelligent tutoring systems (ITS) apply artificial intelligence (AI) techniques on educational computer programs to provide personalized and automated teaching content. ITS aim to measure students' performance by capturing historical academic performance (all the assessments carried out in one academic year [1]), enabling researchers to track students' interactions with each problem they are asked to solve and determine whether they master the related skills. Despite the difficulty of measuring students' performance, there is significant value in revealing and determining students' current knowledge, which can then be deployed to predict future performance.

The method, known as knowledge tracing (KT) [2] leverages archived assessment data of each student by assessing their level of understanding reflected by the responses given and which can be the basis for future performance prediction. Tracking how each student interacts with the ITS over a series of time points yields in-depth insights on a range of relevant aspects, such as how long it takes for the student to start interacting with a question and whether the first interaction is to make an attempt to answer the question or ask for assistance (known as a 'hint'). The challenge of measuring knowledge lies in the complexity of each learner and their environment, including lesson content, cognitive load, lesson duration and context [3]. The current thesis, therefore, takes student performance as an indicator of student knowledge.

This thesis uses KT as the approach due to its robust history in the field of educational data mining (EDM). EDM is defined as the automated identification of patterns from massive educational data archives in such a way as to offer meaningful and valuable outcomes [4]. Its proven value in the field education will boost the reliability of results. Let us suppose a simple mathematics problem is presented to a student. To solve it, the student must have sufficient experience with a specific skill (e.g., addition or multiplication). Once students have read the problem, solving it correctly depends mainly on how much practice they have had with the skill in question. A skill can be learned by repeatedly applying it to solve relevant problems. Thus, KT seeks to track students' knowledge from their assignment data [5].

The main objectives of this thesis are as follows:

- To determine the students' knowledge from the prediction of their performance with the use of ML proven predictive power.
- To predict how accurately students will respond to future questions by assessing the accuracy of their historical responses using ML techniques.
- To identify the most important attributes in determining students' knowledge and highlight the skills sequence types they are taught (interleaved and blocked) and how they affect the knowledge gained.[1]

As the dataset selected for investigation in this study is large, we decided to use ML classifiers as well as regression techniques to meet the study aims. Among the principal challenges to meeting these aims was the exploration of other data sources that support our interpretations, into more confidence of our chosen dataset. Other

[1] Refer to interleaved and blocked sequence definition in Chapter 2, Section 2.2

challenges faced was the original data being a problem-centered while this thesis studies the student and requires a student-centered structure. Moreover, identifying the best order of aggregation steps before reaching the desired final structure. Additionally, maintaining objectivity and avoid bias with qualitative data analysis.

The original data described all aspects of each problem presented to a student. Major restructuring of the data was required, to detect the knowledge of each student, and show the main attributes reflecting student performance. To the best of our knowledge, we are the first to undertake this type of restructuring in this field.

1.2 Problem Statement

The problem is how to determine students' knowledge, to assess the value and quality of a course in education. Performance in academic assessments reflects the extent to which a student has understood and mastered lesson content and related skills (rates them from 'high' to 'low') and indicates the student performance level in future assessments. Moreover, it enables the assessment of whether, and to what extent, the way the problems are presented to the students – the so-called 'skills sequence' – affects performance, which may vary among the students. This thesis examines how six types of skill sequences affected student performance prediction once the dataset had been fully restructured.

1.3 Knowledge Tracing

ASSISTments is one of many well-known ITS's designed to study and explore KT [6]. In such systems, a given problem presented to students is associated with specific skills, and students must either give an answer or ask for a hint, in which case their answer is marked as incorrect. Thus, these types of KT systems enhance personalized learning and aim to model the knowledge of each student, that is, the degree to which the related knowledge component (KC) (i.e., a skill or an exercise) has been mastered [7]. KT is defined as the estimated chance (measured in probabilities) that a given learner has succeeded in mastering each KC of a given course content, based on all their interactions with it, consisting of skills and response outcomes. As the assessment content is progressively covered by students, the KT model continues to provide additional assessments based on probabilities that the student has achieved in learning every single component of the given teaching material, thus predicting the chance that a student will give a correct answer to the next problem, based on their knowledge of the related skills. Otherwise, learners must provide answers, using skills, in a specific duration of time.

The challenges inherent to measure students' knowledge are well known. In the current thesis, the primary indicator of such knowledge is derived from identifying and assessing students' past performance in assessments to accurately predict their future performance. In the next section, the three technologies mainly associated with KT are presented and discussed.

1.4 Machine Learning

As part of the recent rapid advance of science and technology, artificial intelligence (AI) has ushered in opportunities for multiple types of development. Computer-based machine technology has leveraged knowledge from many theoretical disciplines (e.g., mathematical and algorithm complexity), which adds robustness to the intelligence performance. As shown in [8] (Figure 1), ML is one branch of AI, with a focus on building applications that read, learn and increase the accuracy of data over time without being programmed to do so [9]. Figure 1 also shows how ML can be distinguished from other terms commonly used in the field.

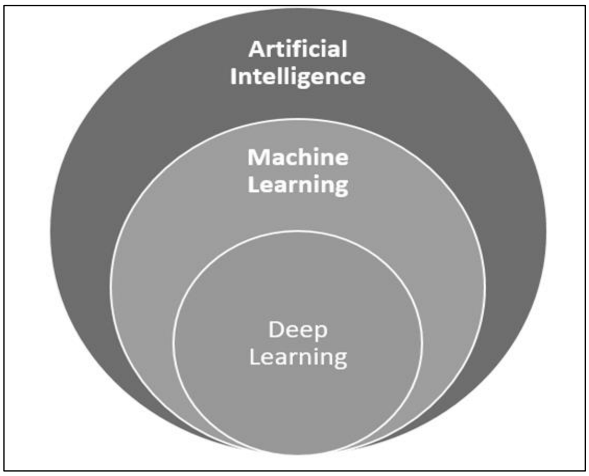


Figure 1: AI, ML, and deep learning (DL) concepts

The learning component in ML algorithms, which enables researchers to process and extract patterns from data, can be supervised, unsupervised or personalized. This thesis employs supervised learning.

1.4.1 Supervised Learning

Under supervised learning, data are input into ML model to be classified or to forecast the most accurate possible results. During the first phase of training, the machine relies on the available input and output to ensure continuous and improving learning over time, eventually yielding the best anticipated outputs [10] (Figure 2). Supervised learning is the best of the three learning methods if the aim is to fully stimulate the typical learning ability of the machine itself. Once the model has been trained, it is particularly useful to help solve systematic problems. Among the most common learning methods currently used are Naïve Bayes (NB), support vector machine (SVM) and K-nearest neighbor (KNN).

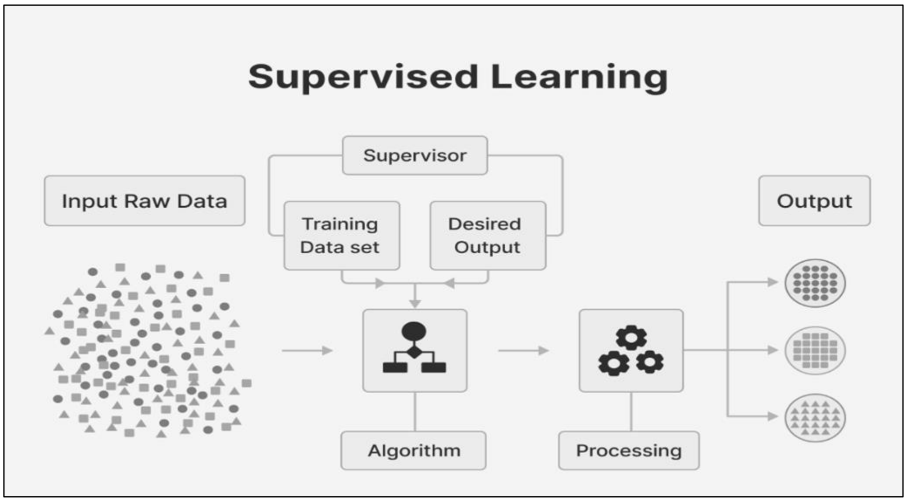


Figure 2: Supervised ML

Under unsupervised ML, as the name suggests, a machine studies uncategorized data, seeking hidden patterns and then leveraging them to group the data as required. Thus, variances or likenesses are used, but it is the machine itself which analyses the data fed to it. As well as unsupervised and supervised learning, the so-called reinforcement learning organizes and analyses part-time response data to create a closed loop of data processing.

1.5 Relevant Literature

Multiple ML techniques have been applied to KT in studies of vastly differing scopes. ML enables the automation of analytical model building, borrowing methods from neural networks, statistics, operations research, and physics which can identify previously missed insights without the need to explicitly program a model on where to search in a body of data or influencing the conclusions it makes. In-depth KT approaches can accurately predict students' skill level; however, a serious shortcoming in such approaches is that they do not take account of students' knowledge of some questions [11]. As shown in [11] (Figure 3) questions which require students to deploy the same or a similar skill can vary considerably in difficulty; in such cases, prediction of skill level may not accurately reflect the student's knowledge.

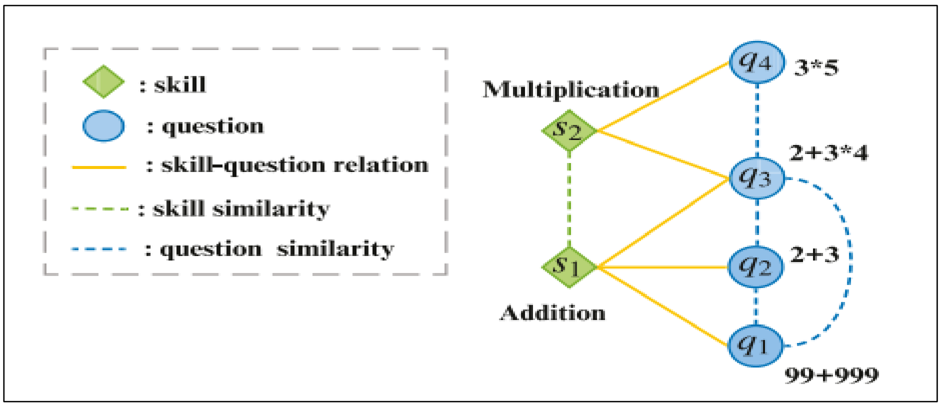


Figure 3: Bipartite graph diagram of question skills. The relationship of questions and skills is open, as is the similarity of skills and that of questions.

KT must be set at a high level, using knowledge of certain questions. However, the very limited interaction between students and questions is a serious limitation and leads to catastrophic failure when the questions are directly used as network input data [12].

A review of the literature reveals multiple studies which have set the goal of predicting student performance, as summarized in Table 1, which states the approaches and methods used. Most ML, KT and data restructuring approaches and techniques begin with basic statistics. For example, [13] built an assistance model based on 'hint' and 'attempt' parameter statistics and their relationship with the probability of getting the next response correct, while [14] used statistics to reach the minimum number of features necessary for highest prediction accuracy. Statistical studies by [1], [15], [16] and [17] explored relationships between specific attributes, including mastery speed, attempts, time interval of attempts taken, skill name and overall student performance.

[3], similarly, analyzed 50 problems attempted by a student and predicted the probability of the student mastering the mentioned number of problems.

Table 1: Summary of relevant research studies

Reference	**Year**	**Approach**	**Techniques**	**Methods**					
				Basic Statistics	Data Re-Structure	ML-Classification	ML-Regression	Knowledge Tracing	Time-Series Analysis
[13]	2011	Predicting student performance Using hints and attempts (assistance model)	PFA, KT, LR	√			√	√	
[18]	2012	Students' test score, mastery prediction	RF, LR				√		
[14]	2012	Student performance prediction using least number of features with	J48, IBK, Kmeans, NB, ONER, VFI	√		√			
[1]	2013	Student performance prediction (For delayed retention tests)	PFA, Resampling (with Mastry Speed)	√	√				
[3]	2015	Predicting student performance	Vanilla RNN, LSTM	√					
[15]	2017	Students' performance discretisation/differentiation using partial credit score	Resampling (with skills)	√	√				
[16]	2018	Predicting student performance	DKT-DSC, KT	√		√		√	
[17]	2019	Predicting confused students who have failed to master the skill	NB, GLM, LR, DL, DT, RF, XGBoost	√	√	√			
[19]	2018	Predicting student performance	BLSTM, GMP						√
[20]	2019	Predicting student performance	J48, NNge, MLP			√			
[21]	2019	Predicting student performance	DSCMN, KT					√	

Table 1: Summary of relevant research studies (continued)

Reference	Year	Approach	Techniques	Methods					
				Basic Statistics	Data Re-Structure	ML-Classification	ML-Regression	Knowledge Tracing	Time-Series Analysis
[22]	2020	Students' assessments grades and final performance	RF, MLP, Nnet, GBM, GLM, LR			√	√		
[23]	2021	Predicting academic performance	MLP, J48, PART, BAG, MB, VT			√			
[24]	2021	Students' academic performance	LR				√		
		Proposed Model		√	√	√	√	√	√

Very few studies in the literature have attempted to restructure the data to describe a specific feature before embarking on the prediction of student performance. In [1], data were divided into several bins, each of which describes a specific number of attempts made by students. In each bin, the percentage of correct answers is measured with respect to the retention test performance. [15] and [17] aggregated datasets to evolve around the skill, in [15] to describe the skill topic and in [17] to describe specific mastery skills. For further clarity, the aggregation of the data modified the structure of all other attributes to ensure a focus on a single main feature. Each value of this feature was not repeated in other rows, making each instance unique and reducing unnecessary repetitions of data.

The ML techniques mentioned above have been applied by several researchers for prediction purposes. The decision tree (DT) and NB classification models were applied, among others, by [14], [16], [17], [22] and [23]. Of these, [14] focused on the most effective features and used feature selection techniques to attain the highest predictive accuracy results; [16] measures student's learning ability then assigns them into a distinct group of students with similar abilities. [17] used various classifiers, random forest (RF) and generalized linear models as well as the DT; [22], [23] and [20] used the single multi-layer perceptron (MLP) classifier; and [23] used both single and ensemble classifiers. One further type of ML technique has been used to predict student performance, known as regression, although, to the best of our knowledge, only by four research groups ([13], [18] [22], and [24]). Remarking that few research who have applied LR are targeting the scope of university performance prediction (such as GPA and bachelor's degree courses predictions) which different than this thesis scope of student's performance in school mathematical (Al Gerba) assessments. [13] and [22] used linear regression (LR) with student performance as the dependent variable and independent variables being prediction results from assistance and other models. LR and RF regressors were used in [18] to explore and compare error differences with other regression methods in predicting performance, while in the study by [24] using LR, input and target variables were enrolment data taken from the student information system of the American University of Nigeria (AUN).

Some research has used KT, however, with the exceptions of [13] and [21]. [13] combined a KT model with both Performance Factor Analysis PFA and assistance models, whereas [21] used the Dynamic Student Classification on Memory Networks (DSCMN) model, which improved on existing KT approaches by recording temporal learning ability for each time interval in the learning process. While for [16], who applied KT techniques to consider and measure question and skill similarities.

Possible visualization techniques include line, bar, and histogram, all of which have a role in, among other things, identifying patterns and finding corrupt data and outliers. As regards time, demonstrating correlations between certain variable changes in a time-series analysis described in graphical plots can usefully determine how data trend over time as well as whether data points are random or fall into patterns. [19], for example, visualize the number of students' interactions with a mentor over a specific number of weeks to see how early instances compared with the overall progress of such interactions.

Table 1 summarizes related studies which have sought to predict students' performance but without using their knowledge. The current thesis bridges the literature gap by determining students' knowledge from their actual performance.

This thesis uses basic statistics for data exploration as well as to help in the creation of additional variables; moreover, statistics were an essential tool in restructuring the data. In addition, classification and regression were adopted as the main ML techniques to achieve the research objectives. The approaches discussed above were used to detect knowledge, while a time-series analysis was used to adduce further evidence of certain results, despite the fact this method has rarely been used in related research, as can be seen from the summary presented in Table 1.

Chapter 2: Methodology

The study presented in this paper was structured in eight stages: data acquisition, data pre-processing, features creation, data restructuring, feature evaluation, training, testing of classification and regression and, lastly, evaluation. Figure 4 in Section 2.2 displays the overall research design.

2.1 Dataset

This research used the ASSISTments Dataset 2012–2013 [25] published by Professor Neil Heffernan of the Worcester Polytechnic Institute (WPI). An ITS for the teaching and evaluating of math students, it compiles all student problem-level data assembled by ASSISTments from September 2012 to September 2013 to present common student behaviors seen across multiple problems. Each row within the dataset designates a particular problem experienced by a specific student, and the dataset encompasses 35 attributes, 46,674 students and 6,123,270 interactions with the ASSISTments system, during which 198 skills were practiced. Data which were not germane to the current research were removed, as were all learners who had had fewer than ten interactions with the system. Moreover, new columns were created based on other attributes. Once the pre-processing stage was complete, the dataset had 39,128 students and twenty-four attributes. In our models, both the ‘correct’ and ‘performance’ columns were used for prediction purposes.

2.2 Research Design

Figure 4, below, presents the methodological tree used by this study, which can be understood as a trunk and two branches. The first steps consisted of cleaning and pre-processing the data, after which it was possible to embark on the training and testing of the model. After the pre-processing stage, the data were restructured to give a student-based representation: after processing, each row represents a unique student interactions, and repetition and duplication in other rows were eliminated.

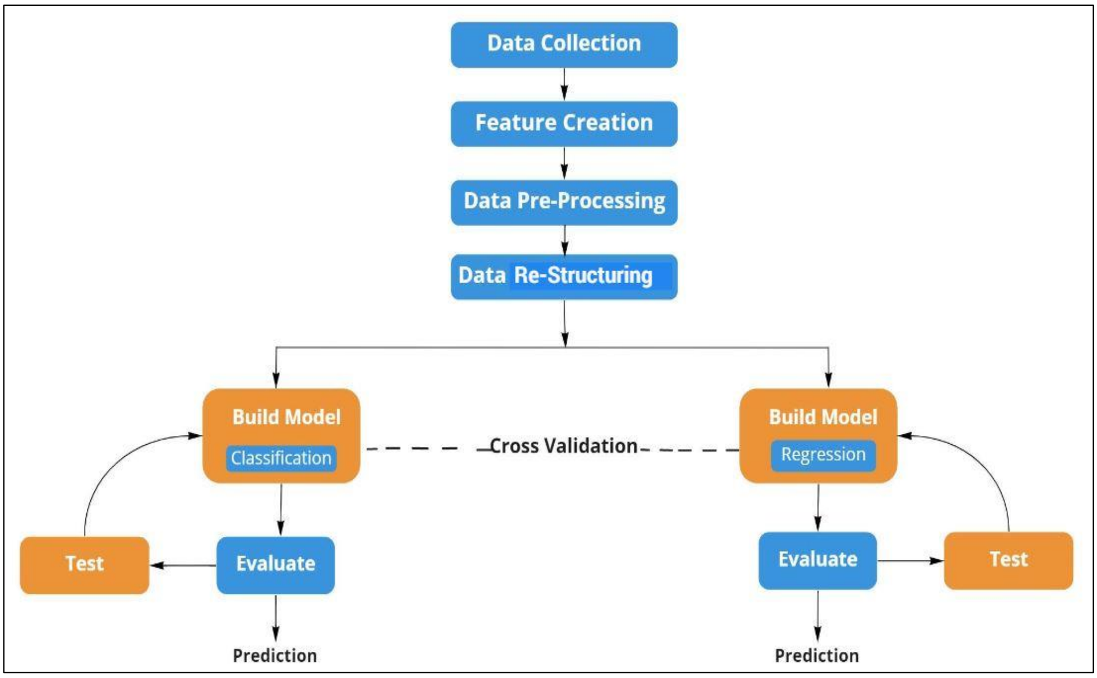


Figure 4: Overall Research Design.

Thereafter, it was critical to determine the essential variables potentially affecting a student's knowledge performance prediction. For this purpose, the classification models shown in the left-hand branch in Figure 4 were used. Features were then evaluated to see which had most effect on the performance in accordance with the problem type (skills sequence), named interleaved or blocked. In a blocked sequence, the same skill is practiced three times consecutively, while in an interleaved sequence, different skills are deployed randomly and linearly [26] (Figure 5).

Blocked	$A_1, A_2, A_3, A_4, B_1, B_2, B_3, B_4, C_1, C_2, C_3, C_4$
Interleaved	$A_1, A_2, B_1, B_2, C_1, C_2, A_3, B_3, C_3, B_4, C_4, A_4$

Figure 5: Blocked vs. Interleaved Study Patterns.

The right-hand branch of the research design shown in Figure 4 presents how ML algorithms were used to predict the correctness of students' next response, using regressions such as LR, DT and RF. Comparison of actual and predicted output were visualized later. The prediction process also integrated extra features inferred from the existing data attributes, specifically hints ratio, first response duration and level of problems difficulty.

2.3 Feature Creation

Historical data were considered to derive three further features which added value to student performance prediction. The features are described below, along with a brief explanation of how they were added, eliminated, or combined, as appropriate.

2.3.1 'Difficulty'

Differentiating between the levels of difficulty of the various problems which students were asked to solve, enabled us to determine students' level as well as how they performed on each specific problem. Level of difficulty was denoted by subtracting 1- 'correct' attribute for each skill, instead for each problem [27]. In other words, the difficulty attribute created for our model defines the difficulty level of the skills associated with all problems. In the following formula, d denotes difficulty level across all related skills, and c denotes average of all the answers for the same skill, obtained from the attribute 'correct':

$$d_{skill} = 1 - c_{skill}$$

2.3.2 'Time_Diff'

We calculated how long each student spent solving a problem (expressed in milliseconds) by using 'Start_Time' and 'End_Time', both of which describe an exact date and time stamp [25]. The time taken to solve a problem indicates the student's understanding of it: the shorter the time, the higher the understanding, and vice versa. Here, t_e is the end timestamp (specific hour, minute and second); t_s is the starting time at which the student began to interact with the problem; and t_d is the time elapsing between t_s and t_e:

$$t_d = t_e - t_s$$

2.3.3 'Performance'

The output variable for the students' performance prediction categorizes learners as low or high performers according to the median values of three conditions: attempt count, hint count and correctness of answers [15]. A fourth condition (bottom_hint) [2] was tested; however, as it was found to have no effect on the results, hence it was eliminated. 'Performance' comprises two values, namely high performance, and low performance, which are essential in predicting performance through classification methods.

[2] For *bottom_hint* and all features description refer to Table 2 in Section 2.4

Additional features were created by calculating the total count of multiple variables, including problems, skills, main problem, and scaffolding problem, each of which was added to the dataset as an additional variable.

2.4 Data Pre-processing and Restructuring

The original data were arranged in rows, each of which presented a specific problem; thus, the data gave an overview of features. The current section describes how the data were prepared, pre-processed, and restructured to be fit to meet the research aims of this study. Twenty-four columns in the original dataset were found to be relevant to the current study, while certain additional attributes were formed from the original data; for example, 'time_diff' was created by subtracting 'start_time' from 'end_time', whereas others, including 'action' feature, were removed on the basis they were too complex.

The first stage in pre-processing was to exclude any students recorded as having had fewer than 10 interactions with the tutoring system [1]. The usual cleaning steps (e.g., filling or excluding null values) were not necessary, as the current study was carried out by ensuring the data was aggregated in a student-centered manner. By primarily considering the average of numeric features and the count of categorical ones, to minimize computational complexity during mining. Specifically, 'skill' values refer to the naming of each skill; hence, it was not necessary to act towards null values of the 'skill' or 'bottom_hint' attributes [15]. Similarly, the following attributes all become unnecessary: 'school_id', 'answer_id', 'assignment_id', 'assistment_id', 'sequence_id', 'teacher_id', 'template_id' and 'student_class_id'. Whereas the 'overlap' column is often computed incorrectly [25], therefore it was excluded. Table

2 below presents the seventeen attributes which were used in making up the final dataset, additional features created shall be explained in Chapter 2.

Table 2: The seventeen original attributes included in the study

Count	Attribute	Description
1	'User_ID'	Refers to the ID of the learner solving the problem.
2	'Skill'	Refers to the name of the problem-related skill, where various skills are linked to many problems.
3	'Problem_ID'	ID of each unique problem, where each major problem will have a different problem ID.
4	'Original'	Refers to whether the current problem is a main or sub-problem (also called scaffold problems: partial problems of the key problem given).
5	'Attempt_Count'	Refers to the number of the times the learner attempted to answer a specific problem.
6	'Hint_Count'	Refers to the count of the times the learner requested a hint while solving a specific problem.
7	'Bottom_Hint;	Shows whether the student requested the final hint which reveals the answer to a specific problem.
8	'First_Action'	Shows whether an attempt or request for a hint was the student's first act on a specific problem.
9	'Start_Time'	Refers to both date and timestamp of the learner starting to solve a specific problem.
10	'End_Time'	Refers to both date and timestamp of the learner completing the solution to a specific problem.
11	'Ms_First_Response;	Refers to the duration spent after the 'First_Action' and 'Start_Time' on a specific problem.
12	'Type'	Describes the problems in the order in which the students receive them; it also refers to the skills sequence provided, where each problem is related to a specific skill; 'Type; is usually one of the following three: • Linear - the student solves all problems in a pre-determined manner. • Random - the student solves all the problems, but each student is introduced to the problems in a different, randomised way. Mastery - indicates the same skill is practised three times consecutively and requires all related problems to be solved correctly to be able to master the skill and move to the next problem, which is known as blocked sequence. The interleaving sequence is practising different skills randomly and linearly, as in the two types random and linear.
13	'Average_confidence (Frustrating)'	Shows the level of frustration of the learners while solving a specific problem.
14	'Average_confidence (Confused)'	Shows the level of confusion of the learners while solving a specific problem.
15	'Average_confidence (Concentrating)'	Shows the level of concentration of the learners while solving a specific problem.
16	'Average_confidence (Bored)'	Shows the level of boredom of the learners while solving a specific problem.
17	'Correct'	Shows whether the learner's answer to a specific problem is correct or incorrect.

Of the six skill sequence types contained in the data presented in Table 3, the current study only investigates two: LinearSection and MastrySection, wherethey serve the third objective of this study which is examining the relationship between interleaved and blocked sequences and students' performance overall. From Table 3 its noticeable that the two sequence types have more data than the other four.

Table 3: Volume of each problem type (skill sequence) within the data

Problem Type (Skill Sequence)	**Number of Related Problems**
Linear Section	149297
Mastery Section	53095
Random Child Order Section	8511
Choose Condition Section	475
Placements Section	47
Numeric Limit Section	15

The aim of the restructuring step was to attain high-level, student-focused information from the dataset, thus unlocking a deeper understanding of the knowledge level of each student. Two levels of restructuring were carried out. First, the data were restructured according to the unique values of each student. Averages were found of important features, such as average of hints and attempts, along with a total count of features such as problems (both main and sub-problems) which enabled deeper insights to be gained into each user [15]. Second, skill levels gave an in-depth representation of skill sequence types and volume, as shown in Table 3. By merging the two levels, further valuable attributes were created, as addressed in the following section. When data restructuring was complete, the essential variable of 'performance' had been added to the study, which played the main role in predicting student performance. Further details are given in the tables in the experiments section.

2.5 Mining Process

This study used ML models to train the data to predict student performance. Both classification and regression were used, with the default parameters settings. Classification is frequently used in prediction as it is a robust means of building a model able to classify data items into predefined class labels. This strength is in alignment with the research goal of predicting future performance from an existing dataset. Regression techniques, on the other hand, selected to predict students' future responses, use the independent variables of aptitude assessment scores. A total of nine classifiers and four regressors was selected. Further details of each are given in this section. It should be noted that the default hyperparameters of the Python library were used in every model.

2.5.1 Logistic Regression

A logistic regression (LR) classifier is a statistical means of analyzing and investigating datasets to produce a binary result. Limiting the outcome to two possibilities requires the variables in the dataset to be dichotomous. Hence, the LR targets attain the model that best fits the relationship between dependent and independent variables X_1 up to X_k [28]:

$$P(D = 1 \mid X_1, X_2, \ldots, X_k) = 1/1 + e^{-(\alpha + \sum_1^k \beta_i X_i)}$$

Where: α and β are the LR model's parameters, selected from a labelled set of instances in the training dataset. The superior fitting function of the LR method is derived from its use of the maximum possibility technique to attain the optimum chance of properly classifying and dividing the categorized data [29].

2.5.2 K-Nearest Neighbor Classifier

The KNN [30] algorithm has the advantage of being easy to use, being a simple, supervised ML process able to resolve both regression and classification problems. The formula below expresses the KNN algorithm using the Euclidian distance square, where n denotes the nearest number of neighbors [31]:

$$D(x, y) = \sqrt{\sum_{k-1}^{n} (x_k - y_k)^2}$$

Subsequently, it is necessary to identify the records with the smallest Euclidian distance, that is, the smallest distance to-n. After being arranged by the minimum distance to-n, the output can be found using the type of KNN with the largest majority. The next step is to determine which records have the smallest Euclidian distance, that is, the smallest distance to-n. Once arranged by minimum distance to-n, the output can be found through majority KNN type.

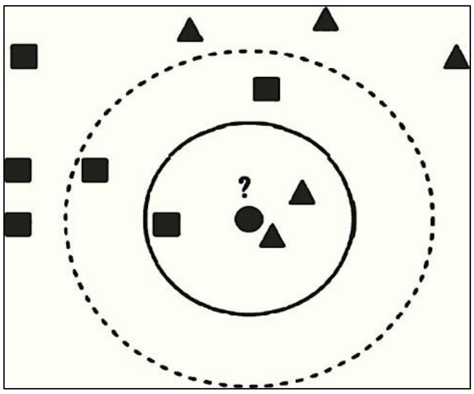


Figure 6: KNN model.

As shown in [32] Figure 6, the KNN model assumes that similar objects exist nearby, that is, that the same things are closer to each other. The K that best suits the data is selected by running the KNN algorithm several times, using different K values, then choosing the K which returns the smallest number of errors encountered while continuing to accurately predict incoming data.

2.5.3 Multi-Layer Perceptron MLP Classifier

The MLP classifier [33] is connected to the neural network. Unlike other segmentation algorithms (e.g., vendor support, NB), the MLP classifier uses the neural network shown in Figure 7, below, to carry out the classification function.

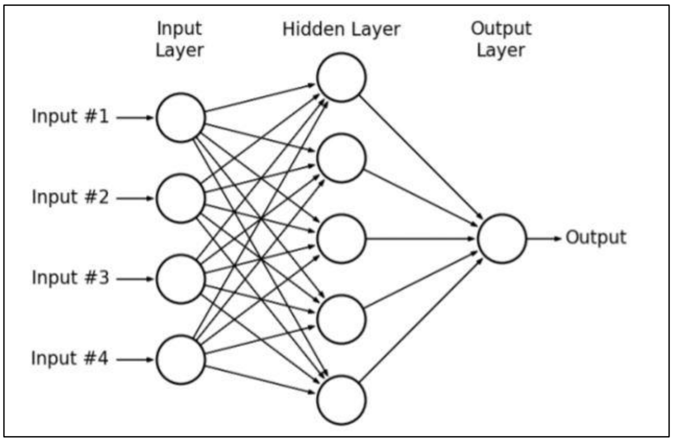


Figure 7**:** Multi-Layer Perceptron Model.

As shown in [34] Figure 7, there are at least three layers of nodes in the MLP model, which are input, hidden and output, and each node represents a neuron that uses a non-linear function. MLP uses the supervised learning method known as back propagation for training purposes.

2.5.4 Linear Discriminant Analysis

The method of dimensionality reduction known as normal discriminant analysis or linear discriminant analysis (LDA) is principally used in supervised ML classification problems. LDA uses multivariate statistics to discover high-dimensional linear relationships and map them in low-dimensional space [35]. It is used to model differences between groups so they can be separated into two or more classes and to project high-dimensional features into a lower dimension. The LDA model assumes that the two classes have equivalent covariance matrices.

LDA, which is frequently applied to controlled classification problems and to enable size reduction, regulates the linear proportions of the higher extremity into a lower dimension setting. The relationship between two classes before and after applying LDA is shown in [36] (Figure 8).

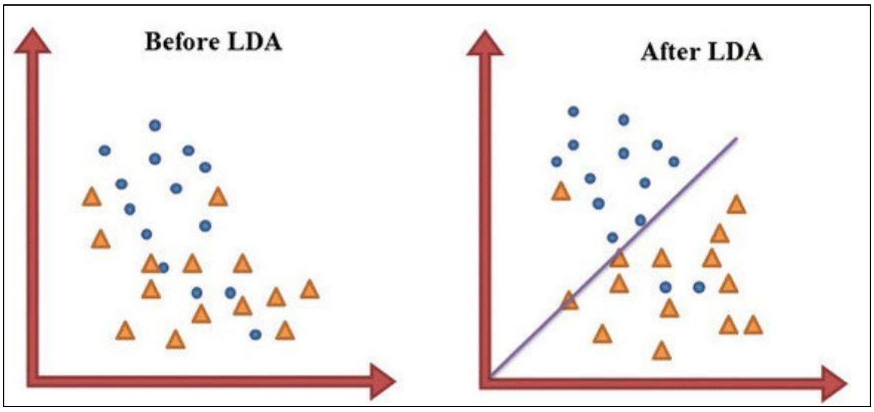


Figure 8: Linear Discriminant Analysis between two classes.

The current study used LDA to create a new axis for two purposes:

- To maximize distance among the means of two classes.
- To minimize the disparity within each class.

2.5.5 Quadratic Discriminant Analysis

Quadratic discriminant analysis (QDA) is considered a standard method for classification purposes as it offers both flexibility and simplicity [37]. It is very similar to LDA, departing from the assumption that all classes have an equivalent covariance and mean.

In QDA, it is not supposed that the covariances will certainly be equal [38]. QDA is particularly appropriate if there is prior expertise, However, QDA has the shortcoming that it cannot be used as a method of reducing dimensionality.

2.5.6 Decision Tree

DT analysis [39] is one of the most widely used predictive models, and different DT packages can be applied to a range of areas. Broadly, DTs are constructed by using an algorithmic approach that organizes approaches to decompose datasets according to different situations.

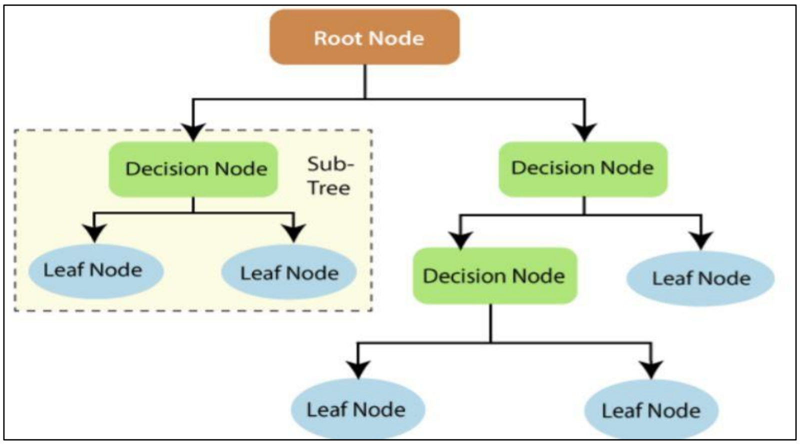


Figure 9: Decision Tree.

Referring to [40] Figure 9 shows the supervised learning and non-parametric techniques used in DT for regression and classification. The goal is to construct an approach which can predict the target variable value by learning simple decision rules that are conditional on the data structure.

2.5.7 Support Vector Machine and Radial Basis Function

SVMs are a means of supervised ML frequently applied to classification problems. The radial basis function (RBF) is the default kernel used in the sklearn's SVM classification [41] method. It can be defined using the formula below:

$$\mathrm{K}(\mathrm{x}, \mathrm{x}') = \mathrm{e}^{-\mathrm{r}} \, ||\mathrm{x} - \mathrm{x}'||$$

Where the gamma value can be manually set and must always be >0. The default value for gamma in sklearn's SVM classification algorithm is:

$$r = \frac{1}{nfeatures * \partial^2}$$

2.5.8 Random Forest

RF [42] is a supervised ML algorithm. The 'forest' built symbolizes the DT, as shown in [43] (Figure 10). Bagging, which is normally used to train the RF, generally combines learning models to raise the final result. The RF has multiple advantages, leading to its being one of the most widely used models: it is easy-to-use, flexible, simple, versatile, generally produces the best result, even without hyper-parameter tuning, and can be used in both regression functions and classification.

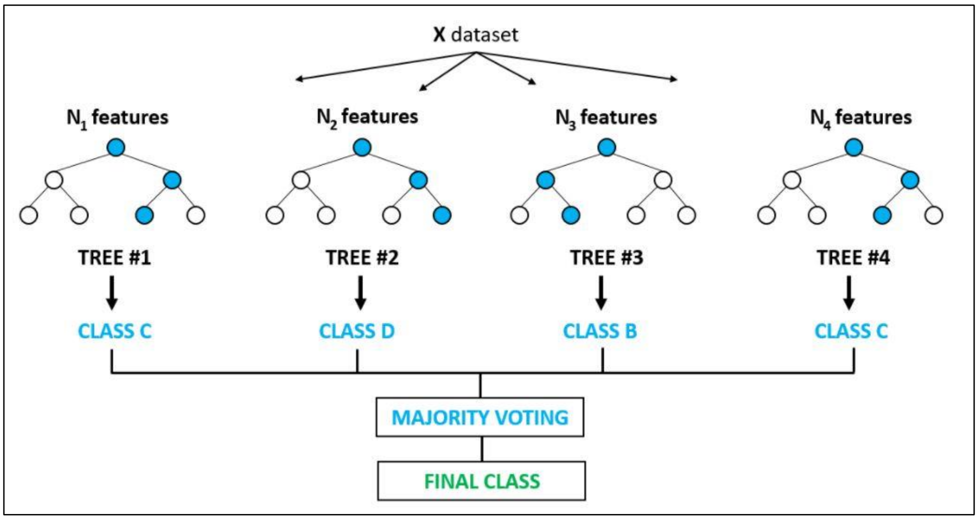


Figure 10: Random Forest Algorithm.

2.5.9 Gaussian NB

Gaussian NB [44] differs from NB in that it tails the general Gaussian distribution and works with continuous data. NB consists of a group of algorithms based on supervised ML approaches based on the Bayes concept. Its advantage is that it offers an easy means of differentiating between classes while maintaining high performance. NB algorithms are frequently used when the input size is high and are generally suitable to solve the complex classification problems of today.

The NB classifier, built on the Bayes Theorem, rests on the strong assumption that the value of one element is independent of the value of any other. In a supervised ML environment, NB classifiers are well trained, and the algorithms only require a small amount of training data to quantify the parameters needed for classification.

2.5.10 Linear Regression

LR [45] is used to find the relationship between the independent and dependent variables by fitting the linear equation to the dataset. Under label encoding [46], the labels can be converted to numeric form to make them easily readable by the machine.

Linear equations represent the linear relationship between the independent and dependent variables. As both dependent variable y and independent x are numeric, the linear equation [47] below:

$$y = a_0 + a_1 x + e$$

which expresses a_0, the constant term, is the interception of the regression line with the vertical axis, where a_1, the slope of the regression line, is named as regression coefficient; e is the accidental error used to measure the result of random factors on the dependent variable:

2.5.11 Extreme Gradient Boost

Extreme Gradient Boost (XGB) is an open-source library that offers an efficient, effective execution of the gradient boosting algorithm. It leverages speed and scalability and constructs supervised regression models which assess the distance between model results and real values, using a function approximation by optimizing exact loss functions and applying multiple regularization methods [18]. The objective function (loss function and regularization) at iteration t that must be minimized is:

$$f(x + \Delta x) \; where \; x = \hat{y}_i^{(t-1)}$$

2.6 Evaluation Measures

An evaluation model must be used if the success of the ML algorithms applied is to be measured. While classification is the process of predicting a discrete class label, regression predicts the value of one variable based on that of another – which, in this study, is the student's correct or incorrect response.

The current study uses four main evaluation metrics – accuracy, precision, recall and F-Score – these being the four metrics of the classification report for the predictive analysis of the classification algorithms performance. Where, the precision and recall precision and recall are considered as diagnostic tools that help in the interpretation of binary (two-class) classification predictive models and make it possible to assess the performance of a classifier on the minority class. Therefore, no additional evaluation metrics were not sought to be needed. They are defined as follows [17]:

- $Accuracy = \frac{TP+TN}{TP+FN+FP+TN}$
- $Precision = \frac{TP}{TP+FP}$
- $Recall = \frac{TP}{TP+FN}$
- $F\ Score = \frac{2PR}{(P+R)}$

The four metrics listed above offer a means to evaluate a classification model by giving information about actual and predicted data and can be understood as follows. Accuracy is the ratio of total number of correct predictions; precision is the ratio of correctly predicted positive cases; recall is the ratio of correctly recognized positive cases; and the F-score or F-measure, in which P = Precision and R = Recall, indicates the balance between P and R.

Regression models use loss as a critical measurement of how accurately a given ML model predicts the expected outcome. The first of the three-evaluation metrics used, mean square error (MSE), estimates the squared difference between real and the estimated values and cannot be negative. MSE is officially defined as follows [48]:

- $MSE = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y_i}\right)^2$

Where: N is the number of samples set for testing. MSE holds high sensitivity for the outliers. When several examples are given with the values of the same input feature, the higher speculation is the target value under MSE. Two other evaluators for regression are mean absolute error (MAE) and root mean squared error (RMSE) [49]:

- $MAE = \frac{1}{N}\sum_{i=1}^{N} |y_i|$
- $RMSE = \sqrt{MSE}$

As the above equations show, MAE merely provides the average scale of errors in the prediction but does not take error direction into consideration, while RMSE provides the square root of the average of squared differences between prediction and observation. Both approaches give better results with lower values.

2.7 Experimental Work

The current research conducted all experiments using Python version 3.6 and Google Collaboratory notebooks. Three major experiments were carried out consecutively with the aim of measuring student performance and determining student knowledge. The final attributes were selected for the training and testing models discussed in Section 2.3. The first experiment aligned with the first research aim, in specific, to predict students' performance to indicate their knowledge and forecast whether their responses in the second experiment would be correct or incorrect. The third experiment aimed to identify and examine the main attributes affecting students' performance and focus on the role of interleaved and blocked skill sequences in it.

2.7.1 Experiment 1 Classification

The classification models discussed in Section 2.5 were used in Experiment 1, comprising the following pre-steps: features creation, data restructuring and setting conditions to measure the performance column values.

Firstly, it was necessary to reorganize the data to make it more suitable to address the study aim of measuring students' performance as an indicator of KT. Two existing features were taken, 'Start_Time' and 'End_Time', both of which describe an exact date and time stamp, and the new feature 'Time_Diff' was created by subtracting one from the other. The time a student takes to solve a specific problem indicates their understanding of it. A further critical feature added, level of difficulty of each skill across all students, was measured in two key steps: aggregating all skills and the overall correctness of all users' related answers, then subtracting the overall correctness from the value of '1'. The outcome provides the pure difficulty level of a given skill across all users, enabling the performance of an individual student to be evaluated against a question related to a specific skill difficulty.

The rich resulting data enabled us to move to the next step, with identifying a single attribute capable of measuring the actual performance of each student from other existing features. We used the median of the three main features (hints, attempts and correctness), noting that the cut-off of the values of each of these three features depends on the median conditions mentioned along with relation set (the higher or lower) in each condition. For example, for a student to be categorized as 'high performance', their overall correctness must be higher than the median correctness value and their total number of hints must be less than the median hint value. When these three conditions had been set, two class labels were created under the feature of 'performance'.

For the classification experiment, ML algorithms were trained to limit the variance between the actual and forecast values. Common values of parameter k such as 3, 5, and 10 are usually set for the k-fold cross-validation, where k defines the folds number in which to split a given dataset. In this experiment we used a 10-fold cross-validation for its common use in literature. K value of 10 means that 10% of the data shall be used for testing. The default parameters have been maintained the in scikit-learn in Python of the mentioned nine classifiers [50]. For LR, regulation is applied by default, where the C value is 1.0, that is, the inverse of regularization strength in the form of a positive float. For other parameter penalties, tol, max_iter and multi_class default values are shown in Table 4.

Table 4: Default parameters of LR

Parameter	**penalty**	**tol**	**max_iter**	**multi_class**
Value	L2	0.0001	100	auto

The norm of the penalty value was set to l2, and the respective tolerance for stopping criteria is 0.0001. The maximum iteration is 100 with an auto multi_class parameter.

The parameter default values for LDA are shown in Table 5. ‘svd’, the singular value decomposition, is the value of the solver parameter; hence, it does not calculate the covariance matrix, making it preferable for datasets with a high number of variables. A default value of none was set for the ‘shrinkage’, class prior probabilities ‘priors’ and number of components ‘n_components’, indicating that no shrinkage is performed. The class proportion ‘priors’ was derived from the training set, the number of components is min (n_classes - 1, n_features), and the tolerance value is set at 0.0001.

Table 5: Default parameters of LDA

Parameter	**solver**	**shrinkage**	**priors**	**n_components**	**tol**
Value	svd	None	None	None	0.0001

In quadratic discriminant analysis (QDA), an extension of LDA, each class uses its own estimate of variance (or covariance when there are multiple input variables). As with LDA, the default value of the class prior probabilities is set to none.

The default parameter DT values are shown in Table 6. The ‘criterion’ default value supports gini impurity in measuring the quality of the split. The best split in each node is chosen. At least two samples are required to divide an internal node, and at least one sample is required as a default value for a leaf node to exist. The maximum RT number is None, which indicates that nodes are expanded until either all the leaves are pure or they contain fewer than the ‘min_samples_split’ samples, while the ‘max_features’ parameter default value is None.

Table 6: Default parameters of DT Classifier

Parameter	**criterion**	**splitter**	**max_depth**	**min_samples_split**	**min_samples_leaf**	**max_features**
Value	gini	best	None	2	1	None

The key parameters of the SVM with the RBF kernel are C, degree, gamma and tol. As shown in Table 7, the regularization parameter C value is 1, the degree of the polynomial kernel function is 3, for the RBF kernel coefficient gamma it is ‘scale’, and the tolerance value is 0.001 for stopping criterion.

Table 7: Default parameters of SVM RBF

Parameter	**C**	**Degree**	**Kernel**	**gamma**	**tol**
Value	1	3	rbf	scale	0.001

The RF classifier has a default value of 100 trees in a forest, called n_estimators, while, as shown in Table 8, gini represents the quality of the split measurement function. Neither max_depth nor max_samples have a default value, given their expansion is potentially unlimited. RF has the same default values as the DT parameters, namely min_samples_split and min_samples_leaf.

Table 8: Default parameters of RF

Parameter	**n_estimators**	**criterion**	**max_depth**	**min_samples_split**	**min_samples_leaf**	**max_samples**
Value	100	gini	None	2	1	None

The main parameter of the KNN is the number of neighbors to inspect for queries; this number is set to a default of 5. The other parameters shown in Table 9 are weights, algorithm, ‘leaf_size’ and ‘p’ in the metric parameter. The ‘uniform’ default value enables all points in each neighborhood to be weighted equally, and the ‘auto’ attempts to select the most appropriate algorithm to fit the technique, based on the values passed. The default value of 30 defines the leaf size which impacts the speed of the building process and enquiry as well as the tree storage memory. There is a direct relation between the metric and p, where p is the power in the Minkowski distance metric and has the value of 2.

Table 9: Default parameters of KNN

Parameter	**n_neighbors**	**weights**	**algorithm**	**leaf_size**	**p**	**metric**
Value	5	uniform	auto	30	2	minkowski

There are two parameters in Gaussian NB: ‘priors’, the preceding probabilities of the classes, which has a value of None; and var_smoothing, which represents a part of the largest variances across all variables, which is added to variances to ensure robust computation, which has a value of 1e-09 (0.000000001).

The MLP classifier enhances the log-loss function. Its parameters are shown in Table 10, with their default values in scikit-learn. Of these, the value of ‘hidden_layer_sizes’ is 100, which is the number of neurons in the hidden layer. The default activation function for the ‘hidden layer’ is relu, which refers to the rectified linear unit function, while the ‘solver’ default value is adam. The remaining parameters, ‘alpha’, ‘learning rate’ and ‘maximum iteration’, have default values of 0.0001, 0.001 and 200, respectively.

Table 10: Default parameters of MLP

Parameter	hidden_layer_sizes	activation	solver	alpha	learning_rate_init	max_iter
Value	100	relu	adam	0.0001	0.001	200

Experiment 1 have summarized the nine classification models used to classify the performance variable values as ‘high’ or ‘low’ performance.

2.7.2 Experiment 2 Regression

Regression algorithms have been proven to be highly effective within ML. This study uses the following four regressors: LR, RF, DT and XGB. As with the classification models, the default parameters in scikit-learn in Python were used [50], and no tuning was applied.

The LR model fits a linear model with coefficients to reduce the remaining sum of squares among the observed targets in the dataset. The ‘fit_intercept’ parameter value is True, which forces the computation of the intercept for this model, while the ‘number of jobs’ used for computation is set by default to None.

The default values of the RF regressor parameters are similar to those of RF classifier; the exception is the ‘criterion’, which is set to ‘squared_error’ value, thus supporting the mean squared error. There are 100 trees.

XGB was designed to improve the execution of the Gradient Boosting framework. The main parameter is the ‘objective function’; for this experiment, the default function is ‘reg_squarederror’ which is normally used for LR. The ‘max_depth’ default value is 6 per tree, the ‘learning_rate’ is 0.3, and the default value of the ‘n_estimators’ is 100, representing the number of trees, which is similar to the count of boosting rounds.

As described above, the default parameters in scikit-learn were maintained in the four applied regressors. Moreover, a 10-fold cross validation was used to ensure well-defined results in the evaluation phase, under which the entire dataset is divided into ten bins of equal size, nine for training and one for testing.

2.7.3 Experiment 3 Identifying Major Attributes Affecting Performance

The most important step within pre-processing was to restructure the data to make them suitable for this study. Originally, the attributes in the dataset described problems; however, the current study aims to detect students’ knowledge and differentiate between the different skills sequences given to students and how each affects students’ performance. Thus, the first task within the re-structuring step was to aggregate the students, represented by the attribute ‘user_id’, after which the number of skills each student encounters was also aggregated. Third step of restructuring stage was to count each problem solved by a student (whether it’s main or sub-problem), while the fourth was to count each of the six skills sequences given to students. The skills sequence type is important to enable comparison of student’s performance across interleaving and blocked sequences. As mentioned above, blocked sequence is one in which a single skill is applied three times consecutively through different questions; all related problems must be correctly solved for the skill to be mastered and the student to move to the next problem. In the interleaving sequence, students apply different skills randomly and linearly [26]. These steps were carried out between pre-processing and features creation to prevent any bias entering the dataset that could mislead the evaluation results. The resulting dataset was aligned with the focus of this study, that is, to predict students' performance to indicate their knowledge.

Table 11: Partial data sample of three random students after the dataset re-structing (variables such as "attempt_count", "hint_count", "average_correct" are not shown due to limitation of the table size).

User_ID	Number of Skill	Number of Problem	MasterySection	LinearSection	Main	Scaffolding	Difficulty	Time_Diff	Performance
21421	3	10	6	4	10	0	0.27	23498.7	High
52535	17	63	26	37	63	0	0.28	18401.41	High
73604	5	33	14	19	31	2	0.3	28560.4	High

The sample data presented in Table 11 demonstrate the final structure of the dataset. Two levels have been developed to illustrate the created features ‘Difficulty’ and ‘time_diff’, in addition to the class label ‘performance’ and the other attributes previously developed, such as ‘scaffolding’ (count of main and sub-problems). The ‘linear’ and ‘mastery’ features refer to the interleaving and blocked skills sequences explained above.

For Experiment 3, we generated and examined the correlation between features in order to reveal the relationships between features and performance attributes. A correlation indicated as between 1 and 0, whether positively or negatively, is identified as among the major variables causing a student’s performance to improve or deteriorate. Moreover, a detailed time series analysis was carried out to identify the effect of the two main skills sequence types on the students’ performance, whereby interactions with the ASSISTment system by four random students over the period of a year were examined by their ability to solve both LinearSection and MastrySection problem types. The analysis sought to substantiate both correlation results and related literature.

Chapter 3: Results

The experiments carried out for this study used multiple measures to evaluate the classification and regression models applied to the ASSISTments 2012–2013 dataset. The degree to which each individual model fit the data was assessed by applying four metrics, that is accuracy, precision, recall and F1-score, to examine the predictive performance of the classifier used on the unseen test set. Three evaluation metrics, MAE, MSE and RMSE, were used on the regression models. Lower values for MAE and RMSE and higher values for accuracy reflected a better fit. The current chapter begins by discussing how the model evaluated the first and second objectives, after which the main relationships between the features are explained, along with their relation to overall performance. Thereafter, the third study objective, the effect of interleaved and blocked sequence types on overall performance, is also explained in relation to the same features.

3.1 Model Evaluation

Both the ML techniques used were applied with a 10-fold cross-validation, using the default values of hyperparameters. Classification quality was chosen as the evaluation metric, including four main measurements, while the regression was evaluated through the three common metrics MAE, MSE and RMSE.

3.1.1 Classification Results

Common evaluation measurements were used to evaluate the performance of the nine classifiers (e.g., precision, recall, accuracy, F-score). Precision relates to how correctly students' performance level, whether high or low, is predicted from total students' total performance samples. Recall gives the ratio of correctly predicted records in the dataset. Accuracy is the ratio of correctly categorized performances within the total number of student performances in the dataset, while the F-score shows the consistent mean of both recall and precision.

As shown in Figure 11, 26.5% of students fell within the high-performance category, with the rest categorized as low-performance learners, using the median technique explained in Chapter 2. Nine classification models were used in the prediction of the performance column, as also discussed in Chapter 2.

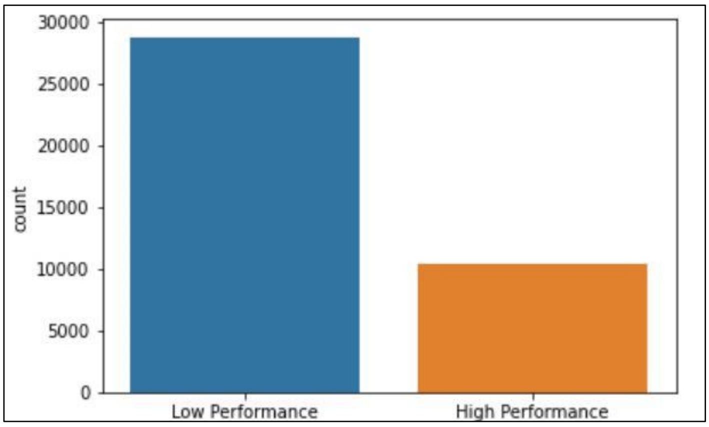


Figure 11: Performance attribute showing high and low performers.

It should be noted that the performance attribute was set as the output of the classification. As shown in Table 12, the best accuracy results were achieved by using both DT and RF with all twenty-four features of the final dataset, in which case accuracy was 99.97% and 100% for each of precision, recall and F-score. The MLP and LR classifiers scored accuracy results of 99.35% and 94.40%, respectively. The performance results are compared with other related literature studies results in Chapter 4. As, to the best of our knowledge, although some studies addressed certain aspects of the problem investigated in the current paper, none targeted exactly the same problem or followed precisely the same methodology.

Table 12: Classification evaluation results on the ASSISTments12

Classification Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	94.40%	95%	95%	95%
LDA	89.94%	89%	90%	89%
QDA	68.25%	85%	67 %	68%
Decision Tree	99.97%	100%	100%	100%
SVM RBF	95.58%	96%	96%	96%
Random Forest	99.97%	100%	100%	100%
K-Nearest Neighbor classifier	92.63%	93%	93%	93%
Gaussian NB	87.01%	90%	87%	87%
Multi-Layer Perceptron MLP Classifier	99.35%	100%	100%	100%

The aim of the classification target was to predict whether students would be low or high performers. Thus, the performance of the nine classifiers used was compared based on accuracy, given this is a standard classification measurement.

3.1.2 Regression Results

A predictive analysis was carried out, using four regression models, with the aim of predicting whether learners' future responses would be correct or incorrect. Hence, the variable containing these existing values is the "Correct", and the model to predict a quantitative response. As shown in Figure 12, most responses were correct and are thus represented by the value of 1. Hence, the regressors applied with 10-fold cross validation produces competent results, as discussed in more detail in the next section.

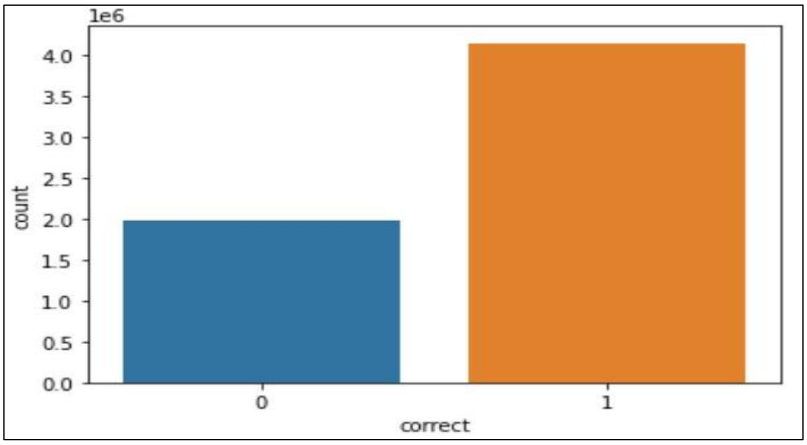


Figure 12: Correct attribute showing “1” value as correct, and “0” as incorrect.

The best mean absolute loss result achieved (see Table 13) is 0.0520, by the RF regressor, while the MSE and RMSE measurements have the exact values of 0.0053 and 0.0729, respectively.

Table 13: Regression evaluation results on the ASSISTments12.

Regression Model	**MAE**	**MSE**	**RMSE**
Linear Regression	0.0845	0.0125	0.112
Random Forest Regressor	0.0520	0.0053	0.0729
Decision Tree Regressor	0.0745	0.0109	0.1047
XGB Regressor	0.0609	0.0069	0.0833

Figure 13 below shows the variance between the actual and predicted values for the entire dataset. Hence, the visualization of the MSE results is the same as that of the prediction results, as the data points fit very closely.

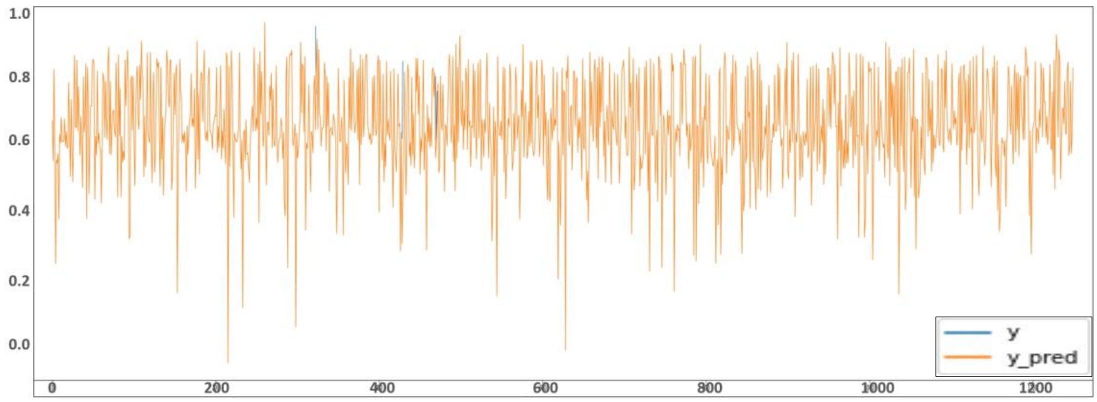


Figure 13: Comparative illustration of partial actual and predicted regression.

3.1.3 Feature Evaluation

Evaluation of all attributes and the class label column ‘performance’ yields certain insights. As shown in Table 14, the variables ‘correct’, ‘attempt_count’ and ‘hint_count’ are medium correlation coefficients, which is reasonable considering that they were also the features used in the conditions applied to create the performance class label. For the ‘correct’ value of 0.6 correlates positively with the class label, where the more answers results are correct, the higher the performance will become. ‘Bottom_hint’, ‘hint_count’ and ‘attempts_count’ all reveal a negative moderate correlation with performance, with values of -0.398, -0.399 and -0.439, respectively, demonstrating that the higher the number of attempts or requests for a hints made by a student, the lower the performance reflected, as the ASSISTment system marks all hints as incorrect answers [25]. Similarly, there is a low to moderate relationship between ‘first_action’ and student performance, as a student’s first action may be an attempt, a request for a hint or an encounter with a scaffolding problem, which are marked ‘0’, ‘1’ and ‘2’, respectively. To clarify further, the first action is given a higher value because the student requests assistance through hints or solves a main problem incorrectly and is then given a scaffolding problem, which lowers the performance.

Table 14: Features correlation with "Performance"

Feature	Correlation
Correct	0.602
Average_confidence(CONCENTRATING)	0.185
Average_confidence(FRUSTRATED)	0.089
LinearSection	0.068
Average_confidence(CONFUSED)	0.066
ChooseConditionSection	0.009
Main_Prob	-0.006
Problem_Count	-0.011
PlacementsSection	-0.013
RandomIterateSection	-0.015
Skill_Count	-0.034
Scaffold_Prob	-0.045
Difficulty	-0.073
RandomChildOrderSection	-0.075
Time_Diff	-0.077
Ms_First_Response	-0.093
MasterySection	-0.112
Average_confidence(BORED)	-0.158
First_Action	-0.286
Bottom_Hint	-0.398
Hint_Count	-0.399
Attempt_Count	-0.439

The MastrySection, or blocked, sequence type correlates negatively with the performance attribute, having the low value of -0.112. The LinearSection, or interleaved, sequence type, in contrast, has a positive relationship on the class label as it helps to improve students' performance and knowledge gain. Two further features related to students' performance are Average_Confidence (CONCENTRATING) and Average_Confidence (BORED). The first has a positive low relation with performance, while the second has the opposite, as will be proven later when a data sample with exact values is used to provide further clarity. Hence, concentration level positively affects students' performance, while boredom affects it negatively. Some features have close-to-zero effect, such as the number of problems (whether main or sub-problem) and number of skills handled by a specific student. The maximum correlation value is correct attribute, which is 0.602.

3.2 Time-Series Analysis

As part of this study, four random students and their entire range of interactions with the system between the chosen dates (2012 to 2013) were selected to further verify the results of the experimental work described above. Figures 14–17 show the interactions of students A, B, C and D. There are two figures per student. The X axis represents the skill sequences (how the problems were presented to the student) accuracy values of between 0 and 1 for answers. To enable the figures to be more easily interpreted, they have been named according to the skills sequence types in the dataset (i.e., LinearSection indicates interleaved and MastrySection indicates blocked). The figures also show the date (year, month) of each interaction, although the exact timing was not considered in this analysis. The figure representing student A shows that the performance ranges between 0.2 and 1.0 for LinearSection-type problems (see left-hand box in Figure 14). The lowest answer value was 0.2, and there were no '0' answers. In contrast, answer values for the MastrySection sequence type ranged between 0.0 and 1.0 (see right-hand box in Figure 14), indicating a lower knowledge gain. Figure 15 similarly shows that student B performed within the range of 0.4–1.0. Two LinearSection-type answers were marked as wrong; hence, student B performed better on this type than on MasterySection sequence type.

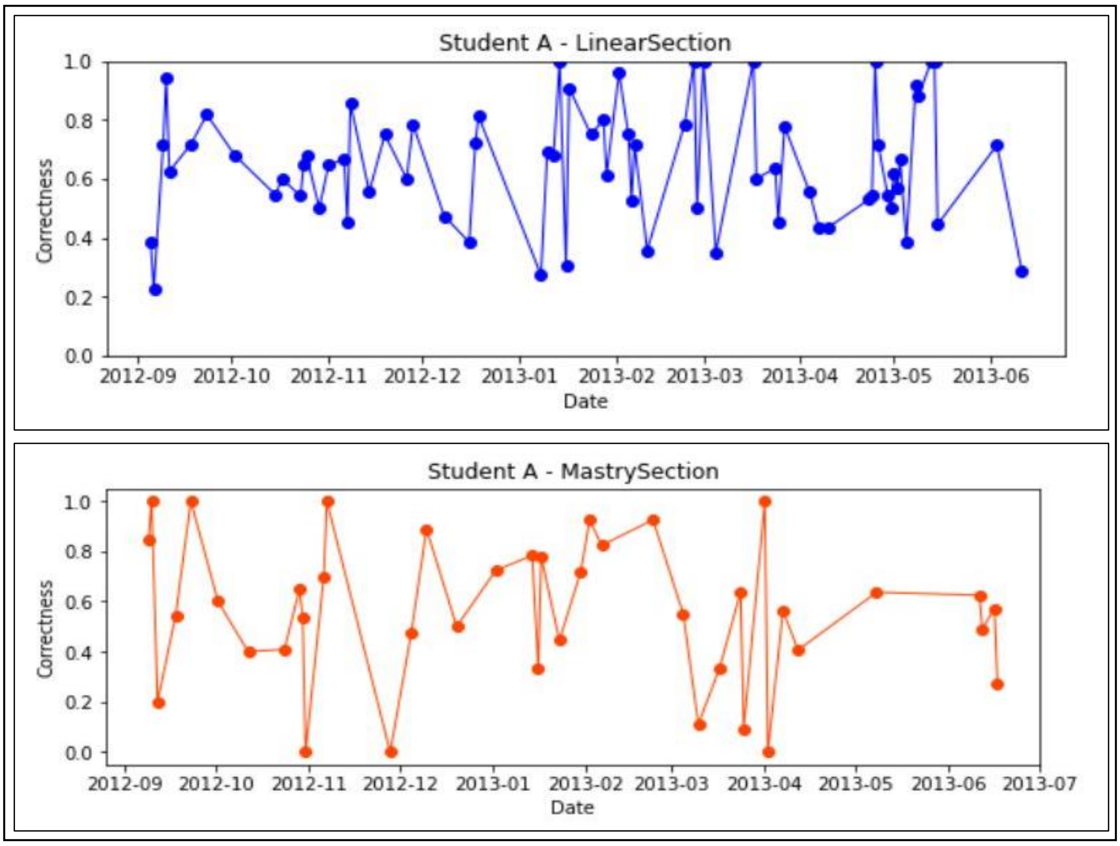


Figure 14: Student A performance comparison between LinearSection and MastrySection Sequence.

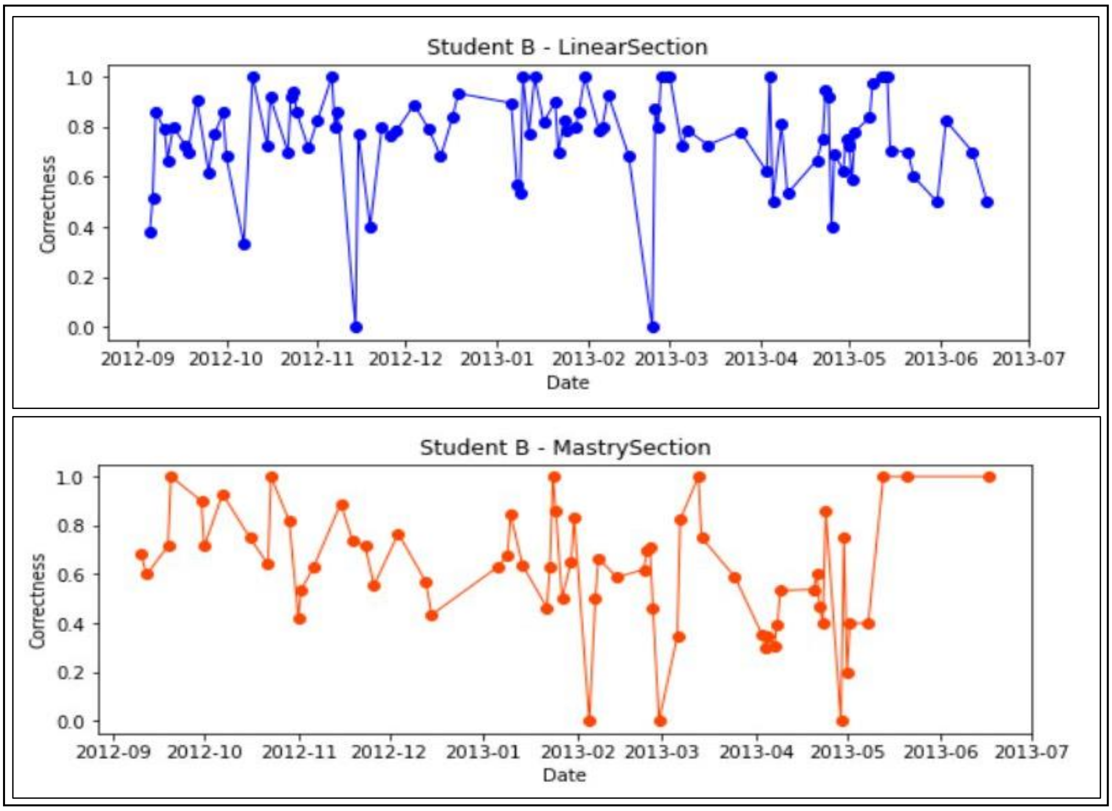


Figure 15: Student B performance comparison between LinearSection and MastrySection Sequence.

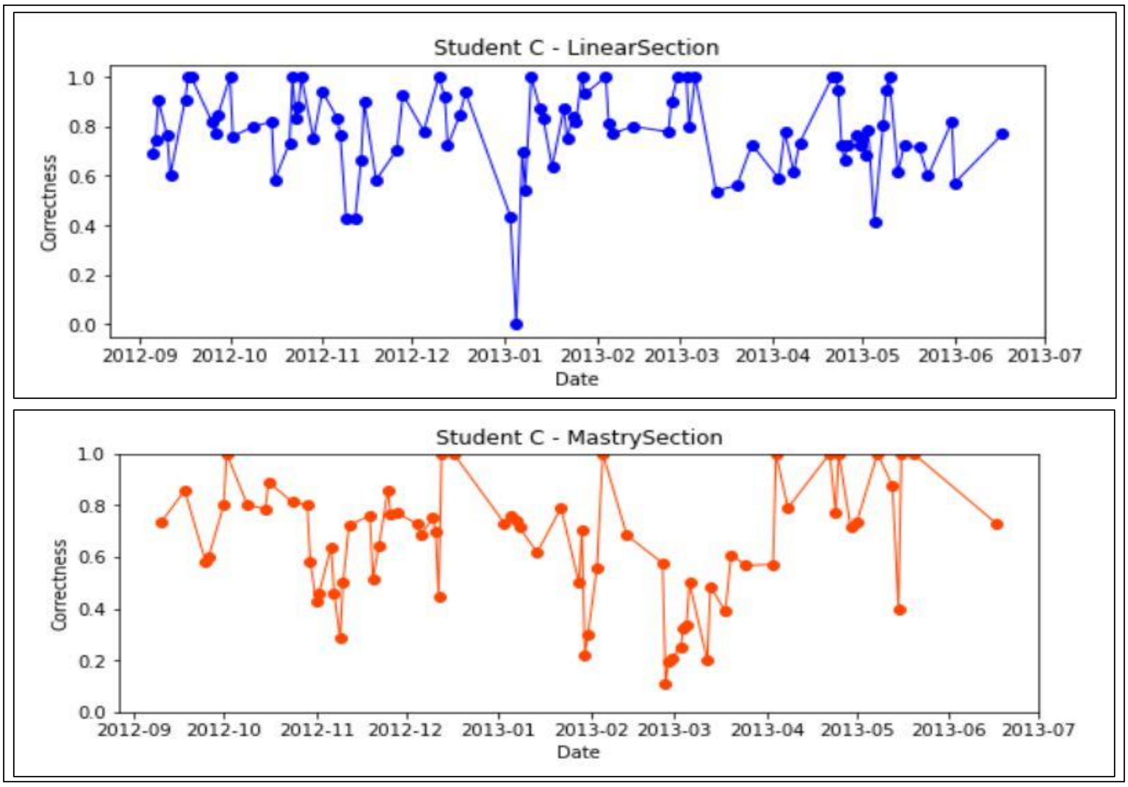


Figure 16: Student C performance comparison between LinearSection and MastrySection Sequence.

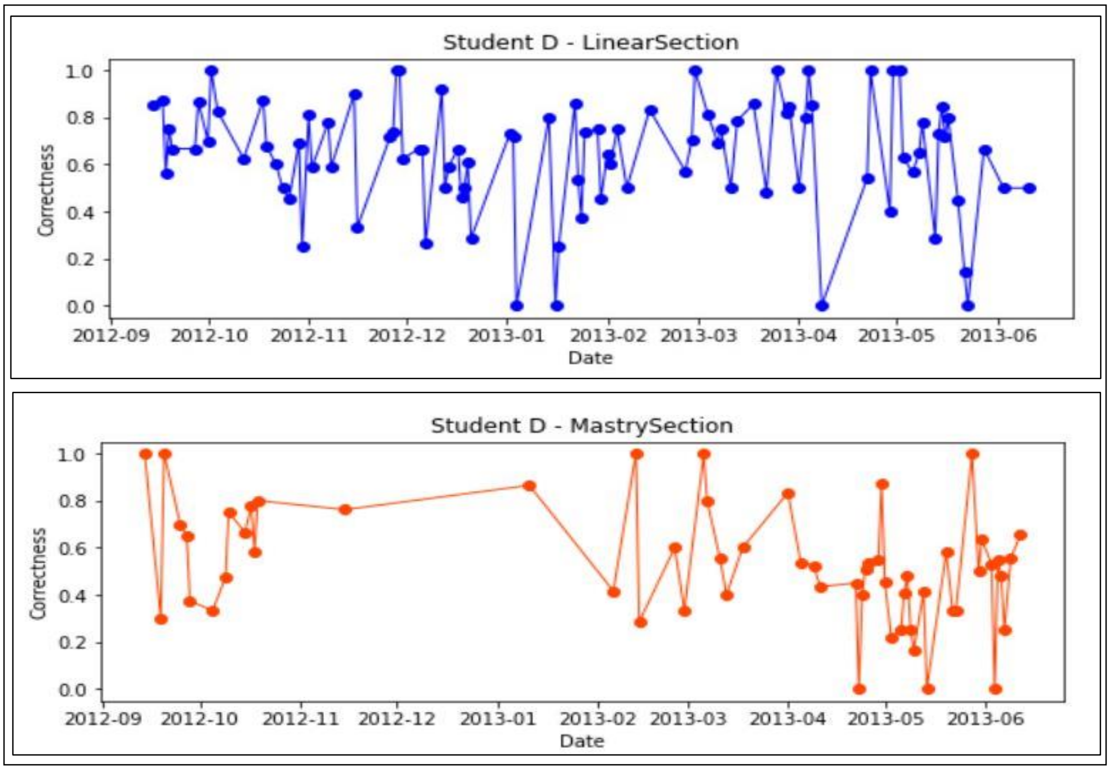


Figure 17: Student D performance comparison between LinearSection and MastrySection Sequence.

The interaction record of student C in Figure 16 clearly reveals the difference in knowledge between the left and right sides of the figure. LinearSection-type problems given to student C yielded a correctness range of 0.4–1.0 with one incorrect answer, whereas the range of answers for the MastrySection-type questions lies between 0.1–1. Turning to student D, Figure 17 shows a gap of practice for the two months from mid-November 2011 to January 2012, which resulted in fewer MastrySection-type than LinearSection-type problems being solved. Nonetheless, the LinearSection-type problems on the left-hand side still show higher correctness. Overall, the results of the analysis of four random learners bear out the results presented earlier in this chapter.

Chapter 4: Discussion

This chapter evaluates and discusses the results presented above, as well as the methodology used in this study and the reliability of the experiments carried out. The study aimed to address the problem of determining students' knowledge, which is well known to be challenging because it encompasses multiple aspects. By measuring a student's performance, however, we indicate whether they are knowledgeable or not.

Several studies in the literature have sought to measure and quantify knowledge from existing student records; however, such studies have employed other EDM mechanisms [51]. As well as predicting future responses, the current study seeks to investigate the effect of all features on students' performance, highlighting the impact on knowledge of different types of skill sequences (blocked and interleaved) in the problems given to students to solve. Our literature research revealed that some studies have targeted similar topics; however, our study and our methodology, is unique, to the best of our knowledge. Hence, in this section, we compare the results of the present study with the findings of other work to show where our results stand in overall of the related classification and regressions models targeting the students' performance prediction. Moreover, this section indicates whether other datasets and re-structuring mechanisms might give a better result in terms of accuracy.

By reshaping the historical data in ASSISTment into an appropriate form for examination from a student-based, rather than a problem-based, perspective, we achieved a dataset which could be used in the experiments undertaken to meet each of the study objectives. The evaluation outcome demonstrates that the results hold up well against two different types of supervised learning prediction, represented by a total of thirteen different ML models: nine classification models for students' performance prediction, representing the first objective, and four regression models for future response prediction, representing the second objective. Table 15 shows an informal comparison of our proposed method results with the related research of same scope (student performance prediction) and who used the two main supervised learning techniques, classification and regression. The highest accuracy achieved in previous methods was 98.8% in [23] forming the average accuracy of MLP, J48, PART, BAG, MB and VT. The details of other methods accuracy are shown in Table 15. Both DT and RF from the proposed method have given an improved accuracy score in the classification models 99.97%, proving the high performance of this model, which indicates that the two classes are uncomplicated to separate enabling simple models such as DT but yet powerful to detect the results in similar way to RF. It is notable that, on the basis of the results of other metrics, such as precision, recall and F1-Score, both DT and RF had the value of 1, indicating both types of models correctly classified every observation. QDA, on the other hand, had an accuracy of 0.684, which is lower than the performance of the other classifiers.

Table 15: Comparison of proposed model with other Classification techniques results

Reference	**Technique**	**Dataset**	**Accuracy**
[14]	J48, IBK, Kmeans, NB, ONER, VFI	ASSISTments 2009-2010	80.87%
[17]	NB, GLM, LR, DL, DT, RF, XGBoost	ASSISTments 2009-2010	85.3%
[22]	RF, MLP, Nnet, GBM, GLM, LR	OULAD	86.8%
[20]	J48, NNge, MLP	UCI 2005-2006	95.78 %
[23]	MLP, J48, PART, BAG, MB, VT	Private Dataset	98.8%
Proposed Model	Classification: DT	ASSISTments 2012-2013	99.97%

Turning to the regression models used, previous methods are listed as an informal comparison in Table 16. Lowest regression results have been achieved by other methods with an RMSE of 0.3730 in [16], while others performance results are shown on same table. Our method has achieved the best performance of 0.0729 by RF indicating slight differences between the observed and predicted values. Nevertheless, our method has proved to outperform all other related research who have applied regression models. Both DT and XGB regressors gave low error prediction value, proving that regression models perform well. LR had a higher error rate than the regressors. Although this cannot be considered incorrect or correct, it does show that the other three prediction models performed better than the LR regressor. The overall regression analysis undertaken to predict the accuracy of students' next response demonstrated advanced MSE results across the models used. Hence, the proposed approach can obtain high regression outcomes, without the need for advanced models such as deep ML.

Table 16: Comparison of proposed model with other Regression techniques results

Reference	Technique	Dataset	RMSE
[22]	RF, MLP, Nnet, GBM, GLM	OULAD	8.131
[18]	RF, LR Expectation Maximization (EM)	ASSISTments 2004-2005 ASSISTments 2005-2006	0.4273
[16]	IRT, DKT-DSC	Cognitive Tutor	0.3730
Proposed Model	Regression: RF	ASSISTments 2012-2013	0.0729

It is of great importance to bear in mind that both classification and regression results emerged from a restructured dataset. The original dataset had developed around a focus on problems; thus, all original features described problems. Once restructured, however, the data were ready to serve the objectives of the present study, having undergone high-level aggregation to eliminate a vast amount of unnecessary information, such as the repetition of problems or skills-related values in each row when practiced by an additional student. Thus, it maintained its flexibility without losing any critical features.

Several observations were made possible in this study due to the restructuring of the dataset. Exploring problem- or skills-based data is a common practice in EDM, such as in [16] and [11], who applied KT techniques to consider and measure question and skill similarities. [15] manipulated the data to describe each skill topic and critical related features, then set a partial score function to discretize the performance in each skill topic of participating students as low or high. The approach taken in the current thesis; each row describes a student. This approach is rarely used, despite the fact it enables a unique understanding of the data and relationships, as well as revealing underlying features which impact student knowledge and performance to give a high-level representation.

When categorizing low and high achievers, the three best known variables were chosen as definers. [15] chose the same three factors by calculating the median of attempts, hint counts and correct answers, demonstrating a fine cut-off between learners based on reliable outcomes of their interactions with the system. The resulting distinction between the two groups was the principal performance indicator in the final dataset.

The findings of [26] show the effect of both interleaving and blocked skill sequences on math students and indicate that interleaved sequences are more powerful. The current study offers further proof that the interleaved sequence (LinearSection in the dataset) has a better effect on students' cognition and the correctness of their answers than the blocked sequence. This finding rests on the data relationship between students in the two groups after the aggregation phase. Table 15 shows each sequence type and its related average of all correct answers given by students within the dataset, after students with fewer than ten interactions with the system have been eliminated. It should be recalled at this point that 'Problem_Type' refers to the sequence of how problems are presented to students. RandomIterateSection and other three types were ignored by the current study as it comprised far fewer problems than LinearSection and MasterySection.

Table 17: Problem type attribute and their corresponding average correct responses

Type	**Difficulty**	**Attempt Count**	**Hint Count**	**Correct Average**
Linear Section	≈ 0.33	≈ 1.27	≈ 0.16	≈ 0.694
Mastery Section	≈ 0.31	≈ 1.45	≈ 0.57	≈ 0.692
Random Iterate Section	≈ 0.30	≈ 1.42	≈ 0.28	≈ 0.628

As can be seen from Table 15, the highest correctness measures were achieved by students to whom blocked-sequence problems were presented, even when these had a higher level of difficulty. A smaller number of hints and attempts can be seen among all students left in the final dataset after pre-processing was completed. On the MastrySection side, a slight difference can be seen in correctness, and a greater number of hints were requested, and attempts were made, underlining the effect of the two sequence types. Moreover, the analysis contributed some early insights into differences between the skills sequences when addressed by the random four students discussed in the results section. The overall performance of these random users proved that an interleaved skill sequence enabled better performance than the mastery-type sequence.

The benefits of this study can be summarized as follows. The approach taken by the current study offers a state-of-the-art technique in which ML algorithms are appropriately applied to students' interaction history to enable the most important features to be identified, additional critical variables to be added, and performance and future responses to be predicted. Restructuring the data to create a suitable high-level dataset facilitated the discovery of which variables have most impact on students' knowledge.

Several contributions are made by this study. First, the newly created and added features, along with the final dataset structure, were essential in revealing the insights and attaining the advanced results presented. The dataset structure will remain available for future researchers to investigate other aspects of it.

Chapter 5: Conclusion

The problem investigated by this study was how to detect knowledge by predicting students' academic performance, given that measuring knowledge is a complex matter which must consider factors including lesson materials, students' cognitive load and time spent attempting to solve problems. The current study departed from the usual focus of problems to concentrate on each student's record of interactions with the questions provided, leading us to restructure the original data to render it student- rather than problem- or skill-based, which are the most common approaches. Each row of the restructured dataset describes all instances of a single student, specifying how many attempts were made to solve the problem, how much time was spent solving all the questions given and all other attributes. Moreover, we examined the impact on students' performance of the different types of skill sequences given to students. Two ML classification and regression respective models were applied to predict student performance. The classification models gave impressive results in terms of predicting performance levels, while the regression models showed low differences in predicting future responses in terms of MAE, MSE and RMSE. After the predictions were made on the ASSISTments dataset 2012–2013, it was proved by the evaluation results that the proposed method predicts performance better than the latest KT models. Moreover, relevant features were evaluated to understand the extent to which they impact students' performance, thus revealing the specific attributes which impact performance overall. In addition, analysis was undertaken of skills sequence types, which revealed that the interleaving sequence outweighs the blocked sequence on performance, as found by previous studies. The results of this study are remarkable, while the classification and regression techniques used ensured flexibility and reliability in predicting performance, leading to more accurate estimates.

The restructured data used in the current study may be used in several further areas of study. First, future researchers should consider additional features in predicting students' performance; the "actions" variable, for example, includes all a student's interactions throughout a year, while response time refers to the time which elapses before the student responds to a question. Investigating these features could add further insights to KT. Second, researchers could attempt to measure student' 'drop out' or 'forgetting' by focusing on the variable of time taken by students to solve questions. Also, to attempt to predict the performance with the use of the time-series factor could be as a third future direction to be studies. Fourth, further research into the effects of interleaved and blocked skill sequences on students' retention levels would be valuable to confirm hypothetical concepts found in the literature, such as Ebbinghaus theory. Fifth, two separate performance prediction on the individual student interactions of blocked and interleaved sequences could be performed using the proposed model, to have further insights and results on this area. Sixth, since existing literature have focused the low and high achievers, a future direction could investigate the average performers of students to add them as a third class, as this could aid in the early prevention of marking average students as low performers. Last future direction is to explore consists of learners' reactions and behaviors, such as boredom, confusion, focus and distraction, towards the skills and questions given, with a view to detecting and interpreting unusual reactions.

In conclusion, this study results recommends the use of interleaved sequence in setting the associated skills with mathematical problems for an improved academic performance. Along with considering all affecting aspects such as, first type of response made, the ratio of the taken attempts and hints, sequential patterns and more, and their values in describing the student actions and behavior in assessments to enhance the individual learner's performance and the knowledge gained. This work has restructured a dataset to render it entirely student-centered, enabling student knowledge to be better detected than through datasets centered on skills or problems. Both classification models and regression ML models showed great potential and the results were achieved without needing to call on methods such as DL algorithms.

References

[1] X. Xiong, S. Li, J. Beck, "Will You Get It Right Next Week: Predict Delayed Performance in Enhanced ITS Mastery Cycle," in *Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference*, 2013.

[2] Y. Choi, et al., "EdNet: A Large-Scale Hierarchical Dataset in Education," in *International Conference on Artificial Intelligence in Education*, 2020.

[3] C. Piech, et al., "Deep knowledge tracing," pp. 505-513, 2015.

[4] M. Yağcı, "Educational data mining: prediction of students' academic performance using machine learning algorithms," *Smart Learning Environments*, vol. 9, no. 1, pp. 1-19, 2022.

[5] A. Corbett, J. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Modeling and User-Adapted Interaction*, pp. 253-278, 1994.

[6] M. Desmarais, R. Baker, "A review of recent advances in learner and skill modeling in intelligent learning environments," *User Modeling and User-Adapted Interaction*, pp. 9-38, 2012.

[7] C. Yeung, D. Yeung, "Addressing two problems in deep knowledge tracing via prediction-consistent regularization," in *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, 2018.

[8] P. Ongsulee, "Artificial intelligence, machine learning and deep learning," in *15th International Conference on ICT and Knowledge Engineering (ICT&KE)*, 2017.

[9] S. Luelmo, N. El Mawas, J. Heutte, "Machine Learning Techniques for Knowledge Tracing: A Systematic," in *Proceedings of the 13th International Conference on Computer Supported Education (CSEDU)*, United Kingdom, Science and Technology Publications, Lda, 2021, pp. 60-70.

[10] R. Loon, "Machine learning explained: Understanding supervised, unsupervised, and reinforcement learning," https://www.crayondata.com/, 2018.

[11] Y. Liu, Y. Yang, X. Chen, J. Shen, H. Zhang, Y. Yu, "Improving Knowledge Tracing via Pre-training Question Embeddings," in *Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*, 2020.

[12] T. Wang, F. Ma, J. Gao, "Deep Hierarchical Knowledge Tracing," in *Proceedings of the 12th International Conference on Educational Data Mining*, Montreal, 2019.

[13] Y. Wang, N. Heffernan, "The “Assistance” Model: Leveraging How Many Hints and Attempts a Student Needs," in *Twenty-fourth international FLAIRS conference*, 2011.

[14] H. Harb, M. Moustafa, "Selecting Optimal Subset of Features for Student Performance Model," *IJCSI International Journal of Computer Science Issues,* vol. 9, no. 5, pp. 253-262, September 2012.

[15] K. Ostrow, Y. Wang, N. Heffernan, "A Comparative Analysis of Scoring Methodologies across Learning Platforms in the Context of Group Differentiation," pp. 96-100, 2017.

[16] S. Minn; Y. Yu, M. Desmarais, F. Zhu, J. Vie, "Deep Knowledge Tracing and Dynamic Student Classification for Knowledge Tracing," in *IEEE International Conference on Data Mining (ICDM)*, 2018.

[17] S. Abidi, M. Hussain, Y. Xu, and W. Zhang., "Prediction of Confusion Attempting Algebra Homework in an Intelligent Tutoring System through Machine Learning Techniques for Educational Sustainable Development," *Sustainability*, vol. 11(1), p. 105, 2018.

[18] Z. Pardos, Q. Wang, S. Trivedi, "The Real World Significance of Performance Prediction," in *International Educational Data Mining Society, Paper presented at the International Conference on Educational Data Mining (EDM)*, 2012.

[19] B. Kim, E. Vizitei, V. Ganapathi, "GritNet: Student Performance Prediction with Deep Learning," arXiv preprint arXiv, 2018.

[20] M. Imran, S. Latif, D. Mehmood, M. Shah, "Student Academic Performance Prediction using Supervised Learning Techniques," *International Journal of Emerging Technologies in Learning*, vol. 14, no. 14, pp. 92-104, 2019.

[21] S. Minn, M. Desmarais, F. Zhu, J. Xiao, J. Wang, "Dynamic Student Classiffication on Memory Networks for Knowledge Tracing," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2019.

[22] R. Alshabandar, A. Hussain, R. Keight and W. Khan,, "Students Performance Prediction in Online Courses Using Machine Learning Algorithms," in *International Joint Conference on Neural Networks (IJCNN)*, 2020.

[23] A. Siddique, A. Jan, F. Majeed, A. Qahmash, N. Quadri, M. Abdul-Wahab, "Predicting Academic Performance Using an Efficient Model Based on Fusion of Classifiers," *Computational Methods for Medical and Cyber Security,* vol. 11, no. 24, p. 11845, 2021.

[24] M. Yakubu, A. Mohammed, "Applying machine learning approach to predict students’ performance in higher educational institutions," *Kybernetes*, vol. 51, no. 2, pp. 916-934, 2021.

[25] N. Heffernan, "ASSISTments Datasets," 2014. [Online]. Available: https://sites.google.com/site/assistmentsdata/datasets. [Accessed 01 Februaury 2021].

[26] K. Ostrow, N. Heffernan, C. Heffernan, Z. Peterson, "Blocking Vs. Interleaving: Examining Single-Session Effects Within Middle School Math Homework," in *International Conference on Artificial Intelligence in Education,* 2015.

[27] Z. Pardos, N. Heffernan, "KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model," in *International Conference on User Modeling, Adaptation, and Personalization,* 2011.

[28] H. Altabrawee, O. Ali, S. Ajmi, "Predicting Students' Performance Using Machine Learning Techniques," *Journal of university of Babylon for pure and applied sciences,* vol. 27, no. 1, pp. 194-205, 2019.

[29] S. Celine, M. Dominic, M. Devi, "Logistic regression for employability prediction," *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* vol. 9, no. 3, 2020.

[30] K. Taunk, S. De, S. Verma, A. Swetapadma, "A Brief Review of Nearest Neighbor Algorithm for Learning and Classification," in *International Conference on Intelligent Computing and Control Systems (ICCS),* Madurai, India, 2019.

[31] Y. Safri, R. Arifudin, M. Muslim, "K-Nearest Neighbor and Naive Bayes Classifier," *Scientific Journal of Informatics Algorithm in Determining The Classification of Healthy Card Indonesia Giving to The Poor,* vol. 5, no. 1, p. 18, 2018.

[32] T. Srivastava, "Introduction to k-Nearest Neighbors: A powerful Machine Learning Algorithm (with implementation in Python & R)," 26 March 2018.

[33] A. Taravat, S. Proud, S. Peronaci, F. Del-Frate, N. Oppelt, "Multilayer Perceptron Neural Networks Model for Meteosat Second Generation SEVIRI Daytime Cloud Masking," *Remote Sensing,* no. 2, pp. 1529-1539, 2015.

[34] H. Hassan, A. Negm, M. Zahran, O. Saavedra, "Assessment of artificial neural network for bathymetry estimation using High Resolution Satellite imagery in Shallow Lakes: case study El Burullus Lake," in *Eighteenth International Water Technology Conference, IWTC18 Sharm ElSheikh*, 2015.

[35] R. Fisher, "The Use Of Multiple measurements In Taxonomic Problems," *Annals of Eugenics,* vol. 7, no. 2, pp. 179-188, 1936.

[36] M. Mohammadi, et al., "Data mining EEG signals in depression for their diagnostic value," *BMC Medical Informatics and Decision Making,* vol. 15(1), pp. 1-14, 2015.

[37] B. Jiang, X. Wang, C. Leng, "A Direct Approach for Sparse Quadratic Discriminant," *Journal of Machine Learning Research*, vol. 19, no. 31, pp. 1-37, 2018.

[38] B. Ghojogh, M. Crowley, "Linear and Quadratic Discriminant Analysis: Tutorial," arXiv preprint arXiv:1906.02590, 2019.

[39] H. Patel, P. Prajapati, "Study and Analysis of Decision Tree Based Classification Algorithms," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 10, pp. 74-78, 2018.

[40] C. Janikow, "Fuzzy decision trees: issues and methods," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 28, no. 1, pp. 1-14, 1998.

[41] K. Thurnhofer-Hemsi, E. López-Rubio, M. Molina-Cabello, K. Najarian, "Radial basis function kernel optimization for Support Vector Machine classifiers," arXiv:2007.08233, 2020.

[42] G. Biau, "Analysis of a Random Forests Model," *The Journal of Machine Learning Research*, vol. 13, pp. 1063-1095, 2012.

[43] R. Abilash, "Applying Random Forest (Classification) Machine Learning Algorithm from Scratch with Real Datasets," https://medium.com/@ar.ingenious, 2018.

[44] Z. Bi, Y. Han, C. Huanf, M. Wang, "Gaussian Naive Bayesian Data Classification Model Based on Clustering Algorithm," in *Proceedings of the 2019 International Conference on Modeling, Analysis, Simulation Technologies and Applications (MASTA 2019)*, Wuhan, China, Atlantis Press, 2019, pp. 396-400.

[45] W. Kraemer, H. Sonnberger, The Linear Regression Model Under Test, Heidelberg: Physica Heidelberg, 2012.

[46] K. Potdar, T. Pardawala, C. Pai, "A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers," *International Journal of Computer Applications*, vol. 175, no. 4, pp. 7-9, 2017.

[47] S. Rong, Z. Bao-Wen, "The research of regression model in machine learning field," in *MATEC Web of Conference*, Chuan, China, 2018.

[48] H. Kling, K. Yilmaz, G. Martinez, "Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling," *Journal of Hydrology*, vol. 377, no. 1-2, pp. 80-91, 2009.

[49] A. Botchkarev, "Evaluating Performance of Regression Machine Learning Models Using Multiple Error Metrics in Azure Machine Learning Studio," arXiv:1804.01825, 2018.

[50] J. Boisberranger, J. Bossche, L. Estève, "Scikit-Learn," [Online]. Available: https://scikit-learn.org/stable/index.html. [Accessed 01 April 2022].

[51] A. Khan, S. Ghosh, S. Ghosh, "Measuring Domain Knowledge for Early Prediction of Student Performance: A Semantic Approach," in *IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, Takamatsu, Japan, December 2020.

UAE UNIVERSITY MASTER THESIS NO. 2022:52

This thesis aims to examine the level of students' knowledge by predicting their current and future academic performance based on records of their historical interactions with the use of machine learning (ML) models. By restructuring data and considering a student perspective, gaining insight into certain important attributes, their inter-relationships, and the overall effect on performance. The result of the proposed approach proves the usefulness of classification and regression techniques and predicts student performance better than current knowledge tracing (KT) models.

Wala Suliman received her MSc in Information Technology Management from the Department of Information Systems & Security, College of Information Technology at UAE University, UAE. She received her B.Sc in Network Engineering from the College of Engineering, Al Ain University of Science & Technology, UAE.