

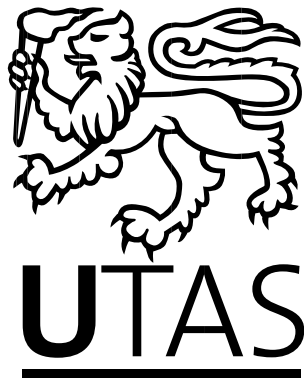
MACHINE LEARNING MODELLING: STUDENT PERFORMANCE PREDICTION USING CLICKSTREAM DATA

by

Yutong Liu

Submitted in fulfilment of the
requirements for the Master of
Information Technology and Systems

University of Tasmania
May 2022

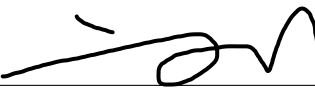


I declare that this thesis contains no material which has been accepted for a degree or diploma by the University or any other institution, except by way of background information and duly acknowledged in the thesis, and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due acknowledgement is made in the text of the thesis.

Signed: _____ 

Date: 05/26/2022

This thesis may be made available for loan and limited copying in accordance with the Copyright Act 1968

Signed: _____ 

Date: 05/26/2022

ABSTRACT

Student performance predictive analysis has played a vital role in education in recent years. Predicting student performance enables understanding of students' learning behaviours, identifying at-risk students, and development of insights into teaching and learning improvement. A new line of research has begun, which utilises various data types retrieved from Virtual Learning Environments (e.g., Learning Management Systems). This thesis investigates the potential of one of the data types, clickstream data, for predictive modelling. A total of 5341 sample students and their click behaviour data from Open University are employed. In this study, the raw clickstream data are transformed with three feature engineering strategies, integrating the time and activity dimensions of students' click actions in weekly and monthly views. This study uses multiple traditional machine learning and deep learning algorithms to build binary classification models using the three feature engineering strategies. The results demonstrate that the LSTM algorithm using a panel data structure in the weekly view produces the best predictive model, up to 90.25% accuracy. The results prove that both click times and click objectives (clicks on activity categories) are significant aspects of the input data structure. This research demonstrates that for clickstream data, deep learning with matrix-based panel data is superior to traditional machine learning with vector-based input data. Also, when using the LSTM algorithm with panel data in predictive modelling, the feature selection step can be optional.

ACKNOWLEDGEMENTS

Words cannot express my gratitude to my supervisors for their invaluable guidance and feedback. I would like to express my deepest appreciation to my primary supervisor Dr Soonja Yeom giving me a chance to join this project. I could not have undertaken this journey without your unwavering support and belief in me. Also, I am extremely grateful to my secondary supervisor Dr Frances Fan, who inspired me, and generously provided knowledge and expertise in Learning Analytics.

I would like to offer my special thanks to Dr Shuxiang Xu for his constructive comments and suggestions in Machine Learning. His immense knowledge and plentiful experience have encouraged me throughout my academic research. Also, I would like to extend my sincere thanks to Dr Quan Bai for providing me with constructive feedback. I am also grateful to a Machine Learning expert from Deakin University, Dr Atul Sajjanhar, for helping me when I encountered difficulties throughout this research project.

My appreciation also goes to my family and friends for their encouragement and support throughout my studies.

TABLE OF CONTENTS

TABLE OF CONTENTS	i
LIST OF TABLES	iv
LIST OF FIGURES.....	vi
Chapter 1 INTRODUCTION	1
1.1 BACKGROUND	2
1.1.1 <i>Learning Analytics (LA) and Educational Data Mining (EDM)</i>	2
1.1.2 <i>Student Performance Prediction and Clickstream Data</i>	3
1.2 RESEARCH MOTIVATION	4
1.3 RESEARCH QUESTIONS AND OBJECTIVES	4
1.4 RESEARCH APPROACH	5
1.5 RESEARCH CONTRIBUTIONS	6
1.6 THESIS STRUCTURE	7
Chapter 2 LITERATURE REVIEW.....	8
2.1 KEY ISSUES OF STUDENT PERFORMANCE PREDICTION	8
2.1.1 <i>Student Performance Prediction</i>	8
2.1.2 <i>Student-Related Features and Data Categories</i>	9
2.2 CLICKSTREAM DATA.....	13
2.2.1 <i>Clickstream Data Background</i>	13
2.2.2 <i>Clickstream Data and Learning Behaviours</i>	14
2.3 MACHINE LEARNING ALGORITHMS.....	15
2.3.1 <i>Overview of Algorithm Use for Clickstream Data</i>	16
2.3.2 <i>Machine Learning Algorithms</i>	18
2.3.3 <i>Deep Learning Algorithms</i>	19

2.4 SUMMARY AND RESEARCH GAPS	22
Chapter 3 METHODOLOGY.....	23
3.1 RESEARCH PHILOSOPHY	23
3.2 RESEARCH STRATEGY	23
3.3 RESEARCH DESIGN	24
3.3.1 <i>Problem Definition</i>	24
3.3.2 <i>Data Collection</i>	25
3.3.3 <i>Student Performance Prediction Approach (SPPA)</i>	25
3.4 RESEARCH METHODS.....	26
3.4.1 <i>Data Selection</i>	27
3.4.2 <i>Data Pre-processing</i>	37
3.4.3 <i>Feature Engineering Strategies</i>	39
3.4.4 <i>Feature Selection</i>	50
3.4.5 <i>Machine Learning Algorithms</i>	55
3.4.6 <i>Model Evaluation Methods</i>	57
3.4.7 <i>Experimental Design</i>	59
3.4.8 <i>Feature Importance Analysis</i>	60
Chapter 4 IMPLEMENTATION AND RESULTS.....	61
4.1 BUILDING MODELS BY TRADITIONAL MACHINE LEARNING	61
4.1.1 <i>Modelling without Feature Selection</i>	61
4.1.2 <i>Modelling with Feature Selection</i>	62
4.2 BUILDING MODELS BY DEEP LEARNING	67
4.2.1 <i>1D-CNN Models</i>	67
4.2.2 <i>LSTM Models</i>	68
4.3 MODEL EVALUATION	69
4.4 FEATURE IMPORTANCE ANALYSIS.....	71
4.4.1 <i>Important Features of Weekly-view Strategy 1</i>	72
4.4.2 <i>Important Features of Weekly-view Strategy 2</i>	73
4.4.3 <i>Important Features of Weekly-view Strategy 3</i>	74

Chapter 5	DISCUSSION AND CONCLUSION.....	75
5.1	DISCUSSION OF RQ1	75
5.1.1	<i>Temporal-Based Aggregation.....</i>	75
5.1.2	<i>Weekly and Monthly Views</i>	76
5.1.3	<i>Feature Engineering Strategies</i>	76
5.2	DISCUSSION OF RQ2	77
5.2.1	<i>Feature Selection.....</i>	78
5.2.2	<i>Algorithms and Models</i>	80
5.2.3	<i>Teaching and Learning Improvement.....</i>	81
5.3	LIMITATIONS AND FUTURE RESEARCH	83
5.3.1	<i>Limitations</i>	83
5.3.2	<i>Future Research</i>	83
5.4	CONCLUSION OF THE THESIS	84
Appendix 1:	TA-WEE - feature weighting result.....	86
Appendix 2:	TA-MON - feature weighting result.....	88
References	89

LIST OF TABLES

Table 2. 1 Taxonomy of data categories used in student performance prediction models.....	12
Table 2. 2 Research focus in clickstream data investigation papers.....	14
Table 2. 3 Algorithms used in predictive analysis using clickstream data	17
Table 3. 1 The presentation of the research methods.....	27
Table 3. 2 OULAD course information	27
Table 3. 3 Details of the seven datasets (CSV files).....	29
Table 3. 4 Details of the <i>courses</i> dataset.....	29
Table 3. 5 Details of the <i>vle</i> dataset	30
Table 3. 6 Details of the <i>studentVle</i> dataset	31
Table 3. 7 Details of the <i>studentInfo</i> dataset.....	32
Table 3. 8 Details of the <i>assessments</i> dataset	34
Table 3. 9 Comparison of the student number among seven courses	36
Table 3. 10 Details of the Strategy 1 datasets: T-WEE and T-MON	45
Table 3. 11 Details of the Strategy 2 datasets: TA-WEE and TA-MON.....	47
Table 3. 12 Details of strategy 3 datasets: P-WEE and P-MON	49
Table 3. 13 Feature engineering strategies summary.....	50
Table 3. 14 Demographic features processed with the <i>one-hot encoding</i> method	52
Table 3. 15 The result of experimental preparation modelling.....	54
Table 3. 16 Building 60 models in the experiment design	59
Table 4. 1 Weight scores of feature weighting by Information Gain in T-WEE.....	63
Table 4. 2 Weight scores of feature weighting by Information Gain in T-MON	64
Table 4. 3 Part of the weight scores of feature weighting by Information Gain in TA-WEE .	64
Table 4. 4 Part of the weight scores of feature weighting by Information Gain in TA-MON	64
Table 4. 5 Weight scores of feature weighting by Information Gain in P-WEE	64
Table 4. 6 Weight scores of feature weighting by Information Gain in P-MON	65
Table 4. 7 Outputs of building 24 M2 models	65
Table 4. 8 1D-CNN trainable parameters for each dataset	68
Table 4. 9 LSTM hyperparameters and trainable parameters for each dataset.....	69

Table 4. 10 All models' performance summaries	70
Table 4. 11 Feature importance analysis summary	72
Table 4. 12 Important features in the model GBT & TA-WEE & M2	74
Table 4. 13 Feature Importance analysis of LSTM & P-WEE	74
Table 5. 1 Feature Selection Result Comparison	79

LIST OF FIGURES

Figure 1. 1 Process of Research Design.....	5
Figure 3. 1 Student Performance Prediction Approach (SPPA)	25
Figure 3. 2 The structure of a module presentation	28
Figure 3. 3 Structure of OULAD (Kuzilek, Hlosta & Zdráhal 2017).....	28
Figure 3. 4 A screenshot of the <i>courses</i> dataset.....	30
Figure 3. 5 A screenshot of a part of the <i>vle</i> dataset.....	31
Figure 3. 6 A screenshot of a part of the <i>studentVle</i> dataset.....	32
Figure 3. 7 A screenshot of a part of the <i>studentInfo</i> dataset (part A).....	33
Figure 3. 8 A screenshot of a part of the <i>studentInfo</i> dataset (part B)	33
Figure 3. 9 A screenshot of a part of the <i>assessments</i> dataset	34
Figure 3. 10 Relationship of <i>assessments</i> , <i>courses</i> , <i>vle</i> , <i>studentVle</i> and <i>studentInfo</i> datasets .	35
Figure 3. 11 Assessment data summary of the course BBB	36
Figure 3. 12 Label changes of the course BBB <i>studentInfo</i> dataset	37
Figure 3. 13 Distribution of demographics	38
Figure 3. 14 A screenshot of a part of the temporary dataset <i>clickRecords</i>	40
Figure 3. 15 The course BBB's 12 activity categories	40
Figure 3. 16 A screenshot of a part of the <i>studentWeeklyClick</i> dataset	40
Figure 3. 17 Pattern differences generated by the <i>studentMonthlyClick</i> dataset	41
Figure 3. 18 A screenshot of a part of the <i>studentMonthlyClick</i> dataset	41
Figure 3. 19 Pattern differences generated by the <i>studentMonthlyClick</i> dataset	42
Figure 3. 20 A screenshot of a part of the dataset <i>activityClickRecords</i>	42
Figure 3. 21 Visualisation of students' click patterns on activity categories (part A).....	43
Figure 3. 22 Visualisation of students' click patterns on activity categories (part B)	44
Figure 3. 23 Dataset structure of Strategy 1 - $X^{(T)}$	45
Figure 3. 24 Daily-based dataset structure.....	46
Figure 3. 25 Dataset structure of Strategy 2 - $X^{(TA)}$	47
Figure 3. 26 Dataset structure of Strategy 3 - balanced panel data $X^{(T) \times (A)}$	48
Figure 3. 27 Using grid search to find the best accuracy through 21 iterations (an example)	53

Figure 3. 28 The architecture of 1D-CNN model	56
Figure 3. 29 1D Convolutional work process (kernel size = 3, stride = 2).....	56
Figure 3. 30 The stacked LSTM architecture	57
Figure 4. 1 M2 model implementation process in RapidMiner	63
Figure 4. 2 Results of the 11 iterations - threshold weight scores in LR models	66
Figure 4. 3 Results of the 11 iterations - threshold weight scores in k-NN models	66
Figure 4. 4 Results of the 11 iterations - threshold weight scores in RF models.....	66
Figure 4. 5 Results of the 11 iterations - threshold weight scores in GBT models	67
Figure 4. 6 Important features analysis in GBT & T-WEE & M2	72
Figure 4. 7 Important features analysis in GBT & TA-WEE & M2.....	73
Figure 4. 8 Features in the model GBT & TA-WEE & M2 (threshold = 0.3).....	73

Chapter 1

INTRODUCTION

Student performance prediction is one of the sub-topics of Learning Analytics (LA) and Educational Data Mining (EDM). Student performance predictive analysis aims to improve teaching and learning, and it can be used in a range of applications such as educational early warning systems (Akçapınar, Altun & Aşkar 2019) and Intelligent Tutoring Systems (Antunes 2008). Conducting such tasks by means of machine learning has been drawing the attention of educators, researchers, and data analysis practitioners in recent years. Some student performance prediction tasks involve predicting students' academic results (e.g., pass/fail) in courses. The data types used in such tasks rely on the course design and the data generated from online learning platforms or systems. According to current research, some data types, such as demographics, academic background, and learning behaviour data, are commonly used (Chen & Cui 2020; Imran et al. 2019; Yang et al. 2020b). Also, a common phenomenon is using mixed-type data to build student performance predictive models. However, these models built for specific courses using specific course-related data types are difficult to reuse. Some studies focus on one type of behaviour data (e.g., video-viewing data) to do such tasks (Brinton & Chiang 2015).

As one of the behaviour data types, clickstream data in education indicates the path a student takes through one or more learning sites. It is an educational data source that is easy to access, regardless of course conditions or learning management systems. To date, clickstream data have attracted insufficient attention from educators and researchers for conducting student performance prediction tasks. One of the reasons is that clickstream data seem to have less explicit connections to students' learning behaviours. A few studies investigate students' learning strategies (e.g., time management) using clickstream data, and demonstrate a certain connection between click actions and students' learning behaviours (Rodriguez et al. 2021). The results are useful in examining the potential of clickstream data in student performance prediction tasks.

1.1 Background

1.1.1 Learning Analytics (LA) and Educational Data Mining (EDM)

Learning Analytics (LA) is an increasingly explored area of education (Romero & Ventura 2020). Although LA has a variety of definitions in studies, one has been widely used in papers (Gasevic et al. 2019; Lee, Cheung & Kwok 2020; Mangaroska & Giannakos 2019). That definition is: “the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs” (Long 2011). As this definition implies, LA is strongly connected to computer-supported learning environments and educational data that are collected from Learning Management Systems (LMSs) (Nistor & Hernández-García 2018). The data used in LA are considerably diverse, from log and survey data to eye tracking, automated online dialogue and “Internet-of-Things” data (Nistor & Hernández-García 2018). Generally, LA aims to generate insights from educational data to improve learning and teaching (Oliva-Cordova, Garcia-Cabot & Amado-Salvatierra 2021; Viberg et al. 2018). These insights help educational institutions to enhance education-related policies, management strategies, and the learning system or environment (Aljohani, Fayoumi & Hassan 2019).

Due to the intimate linkages between LA and educational data, the field of Educational Data Mining (EDM) is developing. EDM has a similar goal to LA: learning and teaching improvement. With the rapid expansion in the volume of educational data, the trend of developing sophisticated techniques grows (Akram et al. 2019; Dutt, Ismail & Herawan 2017). EDM “is an emerging multidisciplinary research area, in which methods and techniques for exploring data originating from various educational information systems have been developed” (Calders & Pechenizkiy 2012, p. 3). Educational information systems such as LMSs have increasing powers to track and trace students’ learning behaviours, a considerable amount of student-related data are constantly collected for EDM applications (Pishtari et al. 2020). Researchers point out that the variables extracted from the LMS data are considerable indicators of student academic success (Macfadyen & Dawson 2010). One of the examples of EDM application is the Intelligent Tutoring System (ITS), which is a data-driven learning system providing tailored and immediate instruction and feedback to students (Romero & Ventura 2013). Predictive modelling is one of the basic EDM tasks (Calders & Pechenizkiy 2012; Namoun & Alshantqi 2020).

1.1.2 Student Performance Prediction and Clickstream Data

Today, LA and EDM drive a new method of prediction that enhances traditional techniques in student performance predictive analysis. Student performance prediction is not a new topic (Aldowah, Al-Samarraie & Fauzy 2019). It has been widely researched in educational institutions to provide insights to inform decision making. Traditionally, student performance prediction uses statistical analysis of students' past academic performance to solve student achievement or retention problems (Aldowah, Al-Samarraie & Fauzy 2019). In recent years, online teaching and learning systems appeared as new phenomena, "contributing in the generation of educational data repositories encompassing learners' interactions, activities, and engagement patterns" (Aljohani, Fayoumi & Hassan 2019, p. 2). These educational data repositories can be further utilised for identifying students who are more likely to be successful or at risk by building predictive models (Aljohani, Fayoumi & Hassan 2019). With this trend, student performance prediction has been rapidly developing in many education areas, such as secondary and higher education, and Massive Open Online Courses (MOOCs).

Following this trend, researchers devote themselves to investigating student performance prediction modelling using different data types. LMSs produce different data types based on their functionalities as well as the course design. For prediction modelling, some researchers use data regarding students' personal and social background, previous academic achievements (e.g., transcripts, admission data), or student self-reporting (e.g., interview or survey data) (Mengash 2020; Nahar et al. 2021; Zollanvari et al. 2017). These data indicate individuals' information or learning environments, which could impact students' academic performance. Other data types, such as event-stream data, are also popular because they directly relate to students' learning behaviours in activities or tasks, such as discussions/fora, quizzes, learning material access, video viewing, and assignments/homework. These types of data explicitly indicate students' learning behaviours.

Another type, clickstream data, is also easily-collected from LMSs. Some studies argue that it is hard to directly observe the exact learning behaviours of students in each online session through the records of the number of clicks (Chen & Cui 2020). However, students' click actions can be used to identify students' learning behaviours from interactions with the learning environment. Therefore, although clickstream data do not directly reflect students' learning, such data are still valuable in investigating students' learning and performance.

1.2 Research Motivation

It is argued that clickstream data can be valuable for student performance prediction analysis. Although the investigation of clickstream data in education is still in its infancy, clickstream data are widely used in customer purchase decision analysis and web page design evaluation in Marketing Analytics (Filvà et al. 2019). Therefore, clickstream data have their value in reflecting peoples' behavioural patterns. In self-regulated online learning contexts, clickstream data can reflect learning behaviours associated with learning tasks, activities (e.g., click count on a quiz), study pace (e.g., clicking frequency), or learning strategies (e.g., click actions before the course starts). In addition, some studies claim that students' click patterns are highly associated with their learning outcomes in MOOCs (Al-Shabandar et al. 2017). From this point of view, it is significant to explore the use of clickstream data for student performance prediction.

Even though a few papers have investigated the application of clickstream data in predictive models (Aljohani, Fayoumi & Hassan 2019), the feature engineering strategies for dealing with such data for improving teaching and learning remain limited. "Feature Engineering" in data science is used to leverage data to create new variables to simplify data transformation while also enhancing predictive model performance.

This research intends to fill the gap. Clickstream data are one of the common types of data generated from LMSs. It is believed that this type of data has the potential to be utilised effectively for student performance prediction. The development of feature engineering strategies helps data mining practitioners process clickstream data and generate a form of student-related features (e.g., students' learning behavioural features) that can be learned from predictive models. From this point of view, exploring strategies for using clickstream data is significant in student performance prediction applications. Feature engineering strategies in this research are the ways of developing some actionable methods to generate the new variables for models that are most likely to lead to a prediction success. Apart from that, predicting student performance using clickstream data and machine learning algorithms and models is explored.

1.3 Research Questions and Objectives

This paper aims to explore clickstream data in student performance prediction. This aim involves developing strategies for processing clickstream data and using them to build a student

performance prediction model. This research aim is achieved through answering two research questions as follows:

RQ1: What feature engineering strategies can be used in student performance prediction using clickstream data?

RQ2: How can machine learning be used to predict student performance to improve teaching and learning?

1.4 Research Approach

This research assumes that the student population can be generalised from selected samples. The research approach has four steps (Figure 1. 1). The first two steps are cycled. The *step 1* is to analyse clickstream data comprehensively to deeply understand its characteristics. It includes reviewing existing literature about clickstream data investigation. The *step 2* is generating feature engineering strategies, based on the outcomes of *step 1*. While generating strategies, it is essential to return to the current papers (*step 1*) to produce more ideas and then return to the dataset for progressive adjustment (*step 2*).

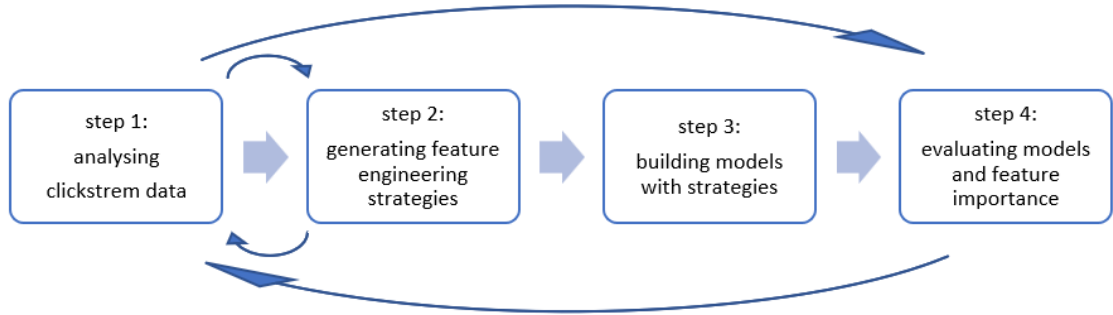


Figure 1. 1 Process of Research Design

The *step 3* is building multiple models with the generated strategies. Due to different strategies resulting in different forms of features and samples of the input data, building multiple models using a range of algorithms to compare their performance is necessary. The reason is that different algorithms have their advantages and limitations when they learn features from the input data. Therefore, a modelling experiment is conducted in this study to examine the prediction success of feature engineering strategies with both traditional machine learning and deep learning algorithms. Also, feature selection is involved in this step to achieve the optimal predictive model.

The *step 4* is to evaluate the predictive models and features. Model performance indicates the prediction effectiveness of combining a form of input data and an algorithm. When the model performance is not ideal (e.g., lower than the benchmark), it is necessary to return to the cycle of *step 1* and *step 2* to modify strategies or even generate new strategies until the model performance becomes reasonable. Therefore, this step is involved in another cycled process. Importantly, the dominant features in the models are related to some influential factors in student performance. The evaluation of these features' importance can provide insights for teaching and learning improvement.

1.5 Research Contributions

This research uses different feature engineering strategies and various machine learning algorithms to build models to predict student performance. The models with these strategies provide valuable insights for teaching and learning improvement. From this point of view, this study has significance in both technical and educational aspects.

From the technical side, first, this research adopts a Student Performance Prediction Approach (SPPA). This research modifies a commonly-used data mining workflow, forming an approach for student performance prediction modelling. Second, this research comprehensively analyses the characteristics of clickstream data and demonstrates how to generate clickstream features from the strategy level. The feature generation involves the aggregation of the click-behaviour aspect, the utilisation of time and activity dimensions, and the temporal data transformation. These provide data mining practitioners with ideas from a broader sense of feature engineering on educational data, especially for clickstream data. Third, this study fits clickstream data into different machine learning models. The results show some best dataset-algorithm combinations' outstanding performance. In this regard, the study provides empirical evidence regarding incorporation of temporal features into traditional machine learning and deep learning models.

From the educational side, this study involves empirical evidence of the effectiveness of student performance prediction based on students' click behaviour data analysis. This contributes insights into the use of clickstream from LMS data to solve educational problems, such as identifying at-risk students. Also, some click-related features indicate influential factors of student academic outcomes in virtual learning environments. These factors reflect the importance of different course periods, access to different learning materials or activities in the learning environments. They

provide educators with insights into how to implement course design and teaching interventions for at-risk students.

1.6 Thesis Structure

This thesis has five chapters in total. Apart from this introductory chapter (Chapter 1), in Chapter 2, this paper presents a literature review regarding some significant aspects related to the research topic, including the overview of Learning Analytics (LA), Educational Data Mining (EDM) and student performance prediction, clickstream data and data mining algorithms. Based on this literature review, in Chapter 3, the methodology is determined and stated from research philosophy to research methods. Then the implementation and results of this research experiment are presented in Chapter 4. Finally, Chapter 5 contains the discussion, conclusion and limitations of this research and recommendations for future work.

Chapter 2

LITERATURE REVIEW

The focus of this study is to predict student performance using clickstream data. First, reviewing relevant research helps further identify the research gap and determine the research significance. Also, reviewing existing literature investigating clickstream data in prediction tasks leads to deep insights into how to generate feature engineering strategies and build predictive models in this research. The literature is categorised into three themes: (1) key issues of student performance prediction, (2) clickstream data, and (3) machine learning algorithms.

2.1 Key Issues of Student Performance Prediction

2.1.2 Student Performance Prediction

According to some studies, in LA and EDM, the biggest proportion of research is computer-supported predictive analysis (63.25%) (Aldowah, Al-Samarraie & Fauzy 2019). Student performance prediction is one of the areas in predictive analysis in education (Hung et al. 2020; Oliva-Cordova, Garcia-Cabot & Amado-Salvatierra 2021). The term ‘student performance’ is used differently across different studies. Generally, there are two types. The first one is student performance at the program level (e.g., a degree program), such as student retention (Burgos et al. 2018; Kemper, Vorhoff & Wigger 2020; Xu, Moon & Schaar 2017). Student performance prediction in this type aims to identify the probability of student dropouts or graduates from a degree program. Such tasks are complex because there are numerous factors that influence student performance. Except for student learning behaviour data from the LMS, such tasks usually also use students’ culture, society, family, socioeconomic and psychological data, previous educational background, and institution interactions (Araque, Roldán & Salguero 2009). Identifying at-risk students supports the institution in increasing its retention rate by offering teaching interventions to students (Hussain et al. 2019).

This research focuses on predictive analysis of the second type of student performance - at the course level (Marbouti, Diefes-Dux & Strobel 2015). *Student performance* is defined as students’

learning outcomes such as assignments or assessments in their courses after a study period (Lemay & Doleck 2020). The predictive analysis can focus on predicting students' final scores or grades or students' pass or fail at the end of the course (Akçapınar, Altun & Aşkar 2019; Yang et al. 2020a). At-risk students refer to those that are more likely to fail the course (Akçapınar, Altun & Aşkar 2019). This type of student performance prediction focuses on student learning behaviours on various learning tasks.

Prediction analysis at the course level can enhance teaching and learning. Instructors can adopt predictive analysis to understand each student's behaviour characteristics in a particular course (e.g., the total time in learning content viewing) (Oliva-Cordova, Garcia-Cabot & Amado-Salvatierra 2021). Then, instructors or educators can design instruction accordingly (Akram et al. 2019). Also, some students who encounter learning difficulties would have more chances to acquire timely intervention from their instructors (Prada et al. 2020). That also gives at-risk students an opportunity to adjust their learning strategies. The integration of prediction analysis has been shown to enhance educational strategies. For example, the forecasting analysis result of a study investigating the adoption level of blended learning systems using online behaviour data supports strategic developments of LMS (Park, Yu & Jo 2016). Existing studies, such as Waheed et al. (2020) and Behr et al. (2020), identify feature importance in prediction modelling to determine the impact of student-related features in academic performance. Identifying important features provides essential insights into teaching and learning enhancement (Helal et al. 2019).

2.1.3 Student-Related Features and Data Categories

Student-Related Features

In student performance prediction modelling, the input features of the models are related to the factors that influence students' performance, such as students' demographic, social, academic background, and learning behaviours (Khan & Ghosh 2020). Many studies use the features of students' enrolment information (e.g., part-time/full-time study, study credits), study achievement (e.g., formative assessment marks), learning behaviours (e.g., the total time of accessing the learning materials, attempting times of quiz) to feed into prediction models (Ifenthaler & Yau 2020). From a practical point of view, features from multiple dimensions are more likely to achieve better predictive results (Hung et al. 2020).

However, some features used in prediction models, such as previous academic records, students and their families' demographics (e.g., gender, age, occupation, residential information, the

highest education), have raised an ethical concern for educational practices (Prada et al. 2020). The ethical concern is that the use of these features in student performance predictive analysis potentially creates a 'student profile' based on their background information (Tomasevic, Gvozdenovic & Vranes 2020). It is not a suitable educational practice when teachers or educators set positive or negative expectations towards their students based on their backgrounds such as nationalities, locations and family income. Some researchers point out that for improving learning and teaching practices, LA and EDM should be employed ethically (Ferguson & Clow 2017).

It is believed that using behavioural features in predictive models is appropriate. That means that students' performance is predicted based on their learning behaviours rather than their demographics or past academic records, thus avoiding potential stigmatisation through stereotyping in educational practices (Seidel & Kutieleh 2017).

Student-Related Data Categories

Existing papers show consistency in small dataset size for course-level student performance prediction tasks. It is argued that educational data mining does not require enormous data sets to be effective (Natek & Zwilling 2014). The research of Nahar et al. (2021) involves merely 80 students with 26 features in two datasets to effectively predict random subjects' grades. Another study, Imran et al. (2019) successfully predicted academic performance by adopting 1,044 students from two schools with 33 features, including students' demographic and social features, grade features and school-related features. The limited size of educational data could be related to the maximum enrolment capacity in some learning environments, including higher education. It is worth noting that a few papers investigated the benefits and importance of big data in EDM. For example, an investigation of Yousafzai, Hayat and Afzal (2020) involved using data from 80,000 students to develop an automated system for predicting students' grades and marks. Nevertheless, the data volume used was still significantly lower than prediction applications in other industries with millions or billions of samples (e.g., Marketing Analytics).

The number of courses involved in student performance tasks is varied. Some studies conduct prediction analysis with multiple courses involved. For example, Tomasevic, Gvozdenovic and Vranes (2020)'s research on student exam performance prediction uses 32,000 students across seven courses. The use of multiple courses increases the size of the datasets to some extent. Some people believe that a bigger scale of data benefits the development of automated predictions of student performance (Yousafzai, Hayat & Afzal 2020). However, other studies argue that course-specific data mining should be used in predicting academic success. These studies demonstrate that the findings from the course-specific predictive models offer educators meaningful insights

in terms of the ways to enhance instructional practice (Gašević et al. 2016). This is because the instructional conditions significantly affect prediction of students' academic performance (Gašević et al. 2016).

Data acquisition is limited to the capability of tracking in LMS. Due to this limitation, a phenomenon is raised - researchers build a predictive model using the types of data that they can acquire. Generally, there are two categories - non-behavioural and behavioural data (Table 2. 1). Non-behavioural data refers to data unrelated to students' learning behaviours. Behavioural data are related to students' learning behaviours and commonly produced from LMSs, which potentially allow considerable gains in prediction results in a range of forms of predictive models (Lopez-Zambrano, Lara Torralbo & Romero 2021; Seidel & Kutieleh 2017). Behavioural data can be further categorised as event-stream data and clickstream data. Event-stream is a form of log data regarding timestamped events related to learning tasks or activities (Gašević et al. 2016). Event-stream data reflect students' behaviours in a known learning task such as discussion/forum, quiz, content access, assignments, and homework.

Existing papers commonly adopt mixed behavioural data from LMS although there are some limitations. For example, a study from Helal et al. (2018) explores prediction analysis using the combined data from Moodle - student involvement in forums, quizzes, book resources and assignments. Another study of Hung et al. (2020) adopts the input variables to improve predictive power, including learning content access and discussion board data. Some people believe that mixed behavioural data provide the multi-view of students' learning behaviour in different tasks (Lopez-Zambrano, Lara Torralbo & Romero 2021; Prada et al. 2020), which is more likely to improve the effectiveness of prediction. However, using mixed behavioural data for prediction tasks leads to low generalisability of the models created. For predicting a new course, the availability of data types for modelling could be different from existing courses. Therefore, although event-stream data have seen significant progress in EDM in recent years, reusing these event-stream-data-based models with good prediction performance is challenging.

Another data type, clickstream, traces and records students' paths when visiting learning sites or LMSs. This research proposes that the education sector lags behind the commercial sector when it comes to integrating clickstream data to conduct predictive analysis. More investigations of clickstream data are reviewed in the next section.

Table 2. 1 Taxonomy of data categories used in student performance prediction models

Category		Examples	Sources
Non-behavioural data	Personal data	demographic data	Prada et al. (2020) Natek and Zwilling (2014) (2018)
	Academic/study record data	transcript/GPA data	Zollanvari et al. (2017)
		admission data	Mengash (2020) Natek and Zwilling (2014)
		participation/attendance data	Marbouti, Diefes-Dux and Madhavan (2016) Kim (2014) Yu et al. (2018) Tsiakmaki et al. (2021)
	Other data	self-report data	Zollanvari et al. (2017) El Fouki, Aknin and El Kadiri (2019) Nahar et al. (2021)
Behavioural data	Event-stream (Log data)	discussion/forum data	Helal et al. (2018) Hung et al. (2020)
		quiz data	Helal et al. (2018) Marbouti, Diefes-Dux and Madhavan (2016)
		learning material data	Helal et al. (2018)
		video viewing data	Doleck et al. (2019) Lemay and Doleck (2020)
		assignment/homework data	Lemay and Doleck (2020) Helal et al. (2018) Marbouti, Diefes-Dux and Madhavan (2016)
		in-class test/response data	Choi et al. (2018)
		assessment data	Livieris et al. (2018)
	Clickstream	clickstream data	Brinton and Chiang (2015) Vo and Nguyen (2020) Tomasevic, Gvozdenovic and Vranes (2020)

2.2 Clickstream Data

2.2.1 Clickstream Data Background

This research transfers definitions of clickstream data from business to education contexts. It is believed that the analysis of a flow of clicks on a website generates insights for decision-making in business contexts (Montgomery et al. 2004). The investigation of clickstream data has been widely used for customer analysis. In business contexts, clickstream “denotes the path a visitor takes through one or more websites” (Bucklin et al. 2002, p. 246). In business analysis, clickstream data are analysed to inform webpage design and evaluation of marketing programs’ effectiveness (Filvà et al. 2019). For example, clickstream data are commonly used to investigate a website design, such as the ease of navigation within the site, the pages that cause the most confusion, and the pages that are essential to reach the desired page (Gao et al. 2022). Also, the pathways of site visitation are used to analyse customers’ behaviours, e.g., classifying customers based on their web visiting patterns or predicting customers’ purchasing likelihood (Baumann et al. 2018; Moe 2003).

Similarly, clickstream in education indicates the path a student takes through one or more learning sites. Clickstream data reflect learning behaviours of students. First, the analysis of clickstream data can be used to understand students’ learning behaviours in online courses (Li, Baker & Warschauer 2020). Also, predictive analysis is able to generate insights into how the learning behaviours impact their academic performance (Yang et al. 2020b). Second, analysing the behaviours also provides insight into LMS design (Jaggars & Xu 2016). For example, visitation of some pages or sites may be consistent with students’ navigation habits in learning (Broadbent & Poon 2015). Therefore, those pages can be used to display the most important materials or notifications when considering course presentation design.

Clickstream data are reliable and offer valid and nuanced information about students’ actual learning processes (Li, Baker & Warschauer 2020). For example, clickstream contains subtle information with time-stamped “footprints” on individual learning behavioural pathways (Jiang, Chi & Gao 2017; Li, Baker & Warschauer 2020). In prediction tasks, these footprints indicate learning efforts, and are more reliable measures compared to conventional methods of self-reporting (Gasevic et al. 2017; Li, Baker & Warschauer 2020).

Clickstream data can be sparse. Clickstream data indicate non-continuous events in behaviour patterns resulting in sparse data. Each click action could be the start point or endpoint of each fragment in learning so that the mid-process could be missed. For example, a click on URLs in a dataset indicates that a student has requested the URL directory paths (e.g., <https://onlinelearning/homepage>), but the request itself is not semantically meaningful in educational contexts (Li, Baker & Warschauer 2020). Therefore, it is not easy to know if a click action on a place (homepage in this case) is related to a learning behaviour. This could be one of the reasons why clickstream data seem to lack explicit meaningful information in learning.

2.2.2 Clickstream Data and Learning Behaviours

Clickstream data are being used for all kinds of educational purposes, with more learning behavioural analysis applications constantly being derived. Table 2. 2 summarises some literature that focused on solving various educational problems by analysing clickstream data.

Table 2. 2 Research focus in clickstream data investigation papers

	Research focus (educational problem-oriented)	Source
1	to analyse student performance by their procrastination patterns	Park et al. (2018)
2	to investigate students' online behaviour changes regarding previewing and reviewing behaviours	Park et al. (2017)
3	to analyse students' engagement with videos in learning by clicking on pausing or changing playback speed on videos	Seo et al. (2021)
4	to investigate how dropout students learn in MOOCs	Zhang, Gao and Zhang (2021)
5	to investigate how students managed their independent learning time.	Rodriguez et al. (2021)
6	to examine students' effort regulation and time management behaviours	Li, Baker and Warschauer (2020)
7	To analyse students answering interactive online questions	Chan and Yeung (2021)
8	To detect student behaviour changes through an e-Book system	Shimada et al. (2018)
9	To analyse student behaviours in an academic library	Jiang, Chi and Gao (2017)
10	To understand students' behaviours in programming activities	Filvà et al. (2019)
11	To investigate video-watching events data to improve prediction quality in MOOC	Brinton and Chiang (2015)

Clickstream data reflect how students interact with online learning environments. Students' learning behaviours can be identified from two dimensions from clickstream data. The first one is related to the sites students click on. These sites in a learning environment are usually categories

of the learning activities or tasks (e.g., quiz or forum sites) – the dimension of activity. The second one regards when students make click actions – the dimension of time. Learning behaviours can be identified by integrating these two dimensions (e.g., the count of clicks or time spent on a web page or site in the LMS, or what pathway students used to arrive there and where they move next).

The utilisation of the two dimensions is varied across existing papers. Some studies utilise the dimension of activity, e.g., learning activity or task categories of clickstream data. For example, the study of Chen and Cui (2020) generates 21 features from clickstream data. Nearly half of the features are based on activity categories using the number of clicks within a target period of the course. The authors adopt aggregated click data, producing a horizontal comparison between multiple learning-activity or learning-task categories. Another study focuses on one category – assessments, using the sum of clicks per assessment as features indicating students’ learning behaviours (Tomasevic, Gvozdenovic & Vranes 2020). Other studies utilise the time-based data sequence of clickstream. For example, the study of Park et al. (2017) investigates student behaviour changes over time from clickstream data by grouping students into decreased, increased, or no change in behaviours for previewing and reviewing learning materials.

To identify learning behaviours from clickstream data, aggregation is one of the most popular ways to create prediction variables. The vectors of counts and frequency are commonly-used measures (Li, Baker & Warschauer 2020). Also, some papers investigate clickstream data using a weekly view of the dataset (Aljohani, Fayoumi & Hassan 2019). From these points of view, clickstream data afford click count and frequency in specific time intervals. Some studies involve the measurement of other behaviour, such as studying in advance (how early students start clicking on a target learning activity) and studying on time (Li, Baker & Warschauer 2020; Park et al. 2017).

In summary, clickstream data can reflect customers’ behaviours in business contexts, such as purchasing-related behaviours. Similarly, clickstream data can reflect learning behaviours of students in education contexts. Clickstream data are reliable but can be sparse. It can be analysed from time and activity dimensions. Also, click count aggregation is one of the most popular clicking measures.

2.3 Machine Learning Algorithms

In this section, some popular data mining algorithms are reviewed from existing papers to build clickstream data models for classification problems. This section aims to obtain ideas of practical algorithms to effectively cope with clickstream data in modelling. It is observed that only a few

papers involve such data in student performance prediction. However, it is possible to transfer techniques used in marketing clickstream research into educational contexts (Werner, McDowell & Denner 2013). Therefore, this research also reviews clickstream data analysis of online users or customer click behaviours in prediction tasks.

2.3.1 Overview of Algorithm Use for Clickstream Data

Some traditional machine learning and deep learning algorithms are commonly used in current clickstream data investigations (see Table 2. 3). One of the most popular baseline algorithms is Logistic Regression (LR). Also, for machine learning, Random Forest (RF), Gradient Boosting-based algorithms, k-Nearest Neighbours (k-NN) and Support Vector Machine (SVM) appeared in several papers. Long Short-term Memory (LSTM), as one of the recurrent neural networks (RNNs), is the most popular in deep learning techniques. Some authors utilise Artificial Neural Networks (ANN)-based and Convolutional Neural Networks (CNN)-based techniques. Fewer papers involve Naïve Bayes, Decision Trees and other proposed or tailored techniques. These algorithms are reviewed in next two sections 2.3.2 and 2.3.3.

Table 2. 3 Algorithms used in predictive analysis using clickstream data

Context	Research topic	Algorithms	Sources
Customer prediction (Marketing Analytics)	Purchase prediction using within-session graph metrics from clickstream data	Logistic Regression , Random Forest , Gradient Boosting Machine	Baumann et al. (2018)
	Product-choice probabilities estimation using a novel model	A tailored expectation-maximisation (EM) algorithm	Nishimura et al. (2018)
	Purchase prediction in the context of e-tourism with modelling	co-EM Logistic Regression (co-EM-LR), Random Forest , Gradient Boosting Decision Tree ; Baseline: Logistic Regression	Zhu et al. (2019)
	Online shopping prediction and marketing interventions	LSTM ; Baseline: Logistic Regression	Koehn, Lessmann and Schaal (2020)
	Customer behaviour prediction with LSTM and no feature engineering	LSTM , Random Forest ; Baseline: Logistic Regression	Sarkar and De Bruyn (2021)
Student prediction (Learning Analytics)	Investigation of students' time management and effort regulation using clickstream data	Multiple Regression Analyses	Li, Baker and Warschauer (2020)
	A deep learning in predictive analytics with clickstream log data.	LSTM , Logistic Regression , Naïve Baye, Support Vector Machine , Decision Tree, k-Nearest Neighbours , Random Forest , Gradient Boosting Machine	Chen and Cui (2020)
	Predicting students' exam results: supervised data mining techniques comparison	Regularised logistic regression , Naïve Bayes, Decision trees, ANN , Support Vector Machine , k-Nearest Neighbours	Tomasevic, Gvozdenovic and Vranes (2020)
	Student performance prediction through a deep ANN and significant features influencing student performance	Deep Artificial Neural Network (deep ANN) Baseline: logistic regression and Support Vector Machine	Waheed et al. (2020)
	At-risk student identification with sequential weekly format of clickstream data using LSTM	LSTM Baseline: logistic regression, Support Vector Machine and Artificial Neural Networks	Aljohani, Fayoumi and Hassan (2019)
	MOOC Performance prediction using clickstream data	Proposed VID-A and VID-N algorithm, The Biases algorithms, Matrix Factorisation, Baseline: k-Nearest Neighbours	Brinton and Chiang (2015)

2.3.2 Machine Learning Algorithms

Logistic regression (LR)

Logistic regression (LR) is a transformation of linear regression that can be seen as a simple version of the regression model for dealing with the problem of binary classification (Zou et al. 2019). LR is based on the estimation mechanism of probability with the 0 or 1 output of the model, using a logistic function defined as follows (Zou et al. 2019):

$$y = \frac{1}{1 + e^{-\theta^T \cdot x + b}}$$

LR is viewed as a reliable prediction technique widely adopted in both marketing and education contexts (Marbouti, Diefes-Dux & Madhavan 2016; Zhu et al. 2019). LR is adopted as the baseline technique for predicting student performance in some studies investigating clickstream data (Aljohani, Fayoumi & Hassan 2019; Waheed et al. 2020).

Support Vector Machine (SVM)

Support Vector Machine (SVM) has a unique classification principle associated with finding an optimal hyperplane to classify classes. SVM can deal with both linear and non-linear separation problems. An important parameter called the kernel trick allows SVM to effectively accomplish non-linear classification (Tomasevic, Gvozdenovic & Vranes 2020). By setting up the kernel trick, SVM would explore the optimal separating hyperplane from the converted new higher-dimensional space. According to Table 2. 3, four papers use SVM, and two of them use it to build the baseline model (Aljohani, Fayoumi & Hassan 2019; Waheed et al. 2020). Another two papers show that SVM presents medium performance in prediction with clickstream data (Chen & Cui 2020; Tomasevic, Gvozdenovic & Vranes 2020).

K-Nearest Neighbors (k-NN)

K-Nearest Neighbors (k-NN) is one of the most frequently-used traditional machine learning methods for solving classification problems. k-NN is a simple, similarity-based algorithm, meaning that it classifies by comparing the similarities between test and training samples (Zhou et al. 2017). Also, k-NN is one of the simplest algorithms because only the k value needs to be managed (Marbouti, Diefes-Dux & Madhavan 2016). In prediction tasks, the value of k is often a small positive integer, indicating the number of nearest neighbours involved with the most votes.

Finding the value k is significant to model performance (Vo & Nguyen 2019). k -NN can also deal with non-linear problems. In the study of Brinton and Chiang (2015), k -NN performs excellently as a baseline model for student clickstream data.

Random Forest (RF)

Random Forest (RF) is an ensemble classifier aiming to improve the performance of the classic tree-based algorithms, developed by Breiman (2001). RF uses the Bagging method to build independent decision trees (creating an uncorrelated forest of trees by Bootstrapping) and combine them in parallel (Bader-El-Den, Teitei & Perry 2019). The prediction by the committee of RF is more accurate than that of any single tree (Bader-El-Den, Teitei & Perry 2019). The advantages of RF include constructing a more accurate model with estimated errors and not being sensitive to noise and exceptions. However, the weakness of ensembles, including RF, is less understandability. The implementation of RF in some cases demonstrates optimal performances of models (Miguéis et al. 2018).

Gradient Boosting Trees (GBT)

Gradient Boosting Trees (GBT), or Gradient Boosting Decision Trees (GBDT), uses the Boosting method to sequentially combine weak learners (typically shallow decision trees) to allow each new tree to correct the previous errors (Gupta, Gusain & Popli 2016). This method intends to reduce *Bias*, which is one of the components of *Accuracy*. A key factor in whether its potential can be realised is related to the number of trees in GBT. Some recent studies have found them to perform better than RF in clickstream data classification tasks (Chen & Cui 2020; Zhu et al. 2019).

2.3.3 Deep Learning Algorithms

Deep learning approaches are based on the Neural Network (NN) concept, which characterises extracting highly complex abstractions and representing hierarchical learning in the data mining process (El Fouki, Akinin & El Kadiri 2019). As a black-box method, deep learning models have low comprehensibility (Romero & Ventura 2013). However, deep learning algorithms are one method that is able to automatically extract complex features (Najafabadi et al. 2015). The existing papers show that some deep learning algorithms are effective in analysing clickstream data, including LSTM, CNN and ANN.

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM, as a Recurrent Neural Network (RNN), is able to learn sequential data (e.g., time-series data) via ‘memorising’ earlier inputs. LSTMs have four interactive layers which communicate in a chain-like process. “Each LSTM cell contains a hidden vector, \mathbf{h} , and a memory vector, \mathbf{m} ” (Karim, Majumdar & Darabi 2019, p. 67719). The memory vector \mathbf{m} adjusts the status, then updates and outputs it, with computation as follows (Graves 2012):

$$\begin{aligned}\mathbf{g}^u &= \sigma(\mathbf{W}^u \mathbf{h}_{t-1} + \mathbf{I}^u \mathbf{x}_t) \\ \mathbf{g}^f &= \sigma(\mathbf{W}^f \mathbf{h}_{t-1} + \mathbf{I}^f \mathbf{x}_t) \\ \mathbf{g}^o &= \sigma(\mathbf{W}^o \mathbf{h}_{t-1} + \mathbf{I}^o \mathbf{x}_t) \\ \mathbf{g}^c &= \tanh(\mathbf{W}^c \mathbf{h}_{t-1} + \mathbf{I}^c \mathbf{x}_t) \\ \mathbf{m}_t &= \mathbf{g}^f \odot \mathbf{m}_{t-1} + \mathbf{g}^u \odot \mathbf{g}^c \\ \mathbf{h}_t &= \tanh(\mathbf{g}^o \odot \mathbf{m}_t)\end{aligned}$$

“where \mathbf{g}^u , \mathbf{g}^f , \mathbf{g}^o , \mathbf{g}^c are the activation vectors of the input, forget, output and cell state gates respectively, $\mathbf{W}^u, \mathbf{W}^f, \mathbf{W}^o, \mathbf{W}^c$ are the recurrent weight matrices, $\mathbf{I}^u, \mathbf{I}^f, \mathbf{I}^o, \mathbf{I}^c$ portrays the projection matrices, σ is the logistic sigmoid function, \odot is an elementwise multiplication, and \mathbf{h}_t is the hidden state vector of the t th time step” (Karim, Majumdar & Darabi 2019, p. 67719).

In complex settings, such as multidimensional data with intersequence and inter-temporal interactions, LSTM is more powerful compared to traditional models (Sarkar & De Bruyn 2021). There are some investigations regarding LSTM and feature engineering. In customer analysis, study of Sarkar and De Bruyn (2021) believes that LSTM does not rely on feature engineering. The authors adopt LSTM architectures, allowing the prediction tasks to skip time-consuming feature engineering processes to forecast a panel of customer behaviours. Another study from Koehn, Lessmann and Schaal (2020) successfully predicts user conversions from clickstream data using LSTM. This study creates several features by transforming the raw clickstream data (e.g., counts and values of the items viewed or in baskets, and the number of page views). These show that LSTM performs very well in using feature engineering-based clickstream data.

In student predictive analysis, some studies highlight the ability of LSTM networks to cope with student temporal behaviours for prediction (Aljohani, Fayoumi & Hassan 2019; Chen & Cui 2020). First, a study of Chen and Cui (2020) adopts click-frequency aggregation data with LSTM to investigate the early prediction of course performance, resulting in moderate prediction accuracy. Another study of Aljohani, Fayoumi and Hassan (2019) uses a clickstream dataset with LSTM to solve a binary classification problem by manipulating 38 weeks as features (from week 1 to week 38). Click counts from the i^{th} week are appended from the first week until the i^{th} week. As a result,

the deployed LSTM model performed with 93.46% precision and 75.79% recall (Aljohani, Fayoumi & Hassan 2019). These studies highlight the potential of LSTM using student temporal behavioural data to predict course performance.

Convolutional Neural Networks (CNNs)

Although Convolutional Neural Networks (CNNs) has no clear concept of dealing with time steps, a few papers use CNN-based models to make predictions with temporal data. CNNs is good at dealing with image-related tasks. An innovative study conducted by Vo and Nguyen (2020) uses an enhanced two-dimensional convolutional neural network (2D-CNN) on temporal educational data by transforming the temporal educational data into a similar data structure of colour images. As a result, with the dataset with 43 features (referring to 43 subjects in the degree-level program), the classification task achieves an accuracy from 85% to 95% in classification. Although this study is not related to clickstream data, it provides evidence of the ability of CNN-based models on real temporal educational datasets. Another study from Qiu et al. (2019) proposes a model DP-CNN to predict student dropout, employing time series clickstream data of student learning behaviours. The authors highlight that the model DP-CNN has the capability to extract features from clickstream data for prediction.

Artificial Neural Networks (ANNs)

According to Table 2. 3, three papers involve Artificial Neural Networks (ANN) in student performance prediction with clickstream data. First, in the study of Tomasevic, Gvozdenovic and Vranes (2020), the overall best model performance (precision) is obtained with a ANN approach by using the non-sequential features of clickstream data and past performance data. Second, research from Waheed et al. (2020) focuses on three categories of prediction (early withdrawals, students with distinction, at-risk students) with the proposed deep ANN. As a result, by using non-sequential features, the deep ANN achieves an 84%-93% overall accuracy in a binary classification, outperforming SVM and LR. Third, in the study of Aljohani, Fayoumi and Hassan (2019), ANNs is one of the baseline models that are employed to identify at-risk students with a sequential weekly format of clickstream data to evaluate LSTM. The result from ANN is slightly higher than another two baseline models - SVM and LR, but significantly lower than the result from LSTM when using sequential clickstream data. According to these three cases, it is believed that the principle of ANN is more likely to perform better when given non-sequential data. In other words, coping with temporal data is not the advantage of this algorithm.

2.4 Summary and Research Gaps

This research firstly reviews the current studies regarding the use of clickstream data in prediction in marketing and education contexts. Reviewing papers in the marketing industry aims to generate ideas based on customer purchase prediction modelling with clickstream data and transfer them into the prediction task of this research. In summary, based on the literature review, this research revealed the following:

- In performance prediction, most studies use event-stream data with an explicit sense of learning tasks (e.g., discussion data, quiz data, assignment data), and fewer studies explore clickstream data (as discussed in section 2.1.3).
- To date, a few works on predictive modelling with clickstream data demonstrate temporal features and non-temporal features generated from clickstream data (as discussed in section 2.1.3). Although some papers involve a feature engineering process, none of them demonstrate how clickstream data feature engineering methods are developed.
- Some studies consider building the generalised models, which involve multiple courses. In contrast, others draw attention to building course-specific models for predicting student performance to provide insights into teaching intervention in the course-specific context (as discussed in section 2.2.1).
- For prediction modelling, current studies using clickstream data in prediction tasks in business and education contexts show similar algorithms used. The most popular algorithms include machine learning and deep learning (as discussed in section 2.3).

It is found that the current available studies have not explored students' click behaviours from the angle of feature engineering strategies in academic performance prediction. Also, few papers investigated how to use clickstream data for student performance prediction tasks. Thus, this study intends to fill this research gap. This study aims to develop a student performance prediction model using various clickstream data feature generation strategies and multiple machine learning algorithms.

Chapter 3

METHODOLOGY

This chapter elaborates on the research methodology. Firstly, the philosophy that underpins the research strategy taken in this research is outlined (section 3.1). Then, the resulting choice of a quantitative approach is demonstrated (section 3.2). The following section discusses the research design, including problem definition, the method of data collection, and data mining approach for student performance prediction using clickstream data (section 3.3). The chapter then outlines all the detailed methods of this research (section 3.4).

3.1 Research Philosophy

This research aims to investigate clickstream data in predicting student performance. For this research, an objective standpoint is taken. As an epistemology, objectivism assumes that “reality is external to the knower” (Jonassen 1991, p. 9). The assumption of objectivism is a realist ontology, and the objective reality is not dependent on human understanding but instead exists independent of the subject (Jonassen 1991). In other words, no matter whether people are aware of it, a reality of a matter exists. Finding the truth of objective reality is related to analysing repeated observations in highly controlled contexts.

3.2 Research Strategy

Based on the objective perspective, this research adopts a quantitative research methodology dedicated to quantifying data analysis with a deductive approach. The quantitative method of this research is based on an experiment. By experimenting, this research systematically examines whether there is a relationship between variables. Specifically, this research involves manipulating an independent variable (students’ click behaviour) to measure its effect on a dependent variable (the students’ final course results) by predictive modelling.

3.3 Research Design

The research design integrates three coherent components of the student performance prediction problem - problem definition, data collection and the Student Performance Prediction Approach (SPPA). First, a clear definition of the problem facilitates understanding and defines the direction of the investigation (section 3.3.1). Based on the problem definition analysis, a set of data and a data mining approach are necessary to conduct this research. Therefore, the data collection method (section 3.3.2) and the approach for student performance prediction of this research are defined next (section 3.3.3).

3.3.1 Problem Definition

This research investigates a binary classification problem. The clickstream student performance prediction problem is stated as follows: The data include a collection of click behaviour observations, X , of a set of students, S . Specifically, $S = \{s_1, s_2, \dots, s_i, \dots, s_n\}$, where s_i represents the i^{th} student in S , and there are n students in total. $X = \{x_1, x_2, \dots, x_i, x_n\}$ where x indicates each individual student's click behaviour. x_i is the click behaviour of the i^{th} student. The prediction outcomes of the performance of S are labelled. A set of labels is $Y = \{y_1, y_2\}$. The label is $y_1 = 1$ if students pass their courses, and $y_2 = 0$ if students fail their courses. Y_i is the prediction outcome of the i^{th} student. The goal of classification is to find a function f :

$$f: X \rightarrow Y$$

In the equation, the function f is able to correctly predict unknown labels of a new student's click behaviours \tilde{x} , i.e.

$$f: \tilde{x} \rightarrow \tilde{Y}$$

The focus of this research is to generate a form of X , then find the function f by modelling.

According to this definition, a dataset containing a set of S and their click behaviour observations X is needed. Additionally, this research needs a predictive modelling approach to solve the problem – find the function f . Therefore, how to collect the data and what predictive modelling approach to use are proposed in the following sections.

3.3.2 Data Collection

This research utilises secondary data (e.g., open-source clickstream data) to conduct analysis and examine in the experiment. The data collected must include a set of S , their click behaviour observations X , and their academic performances Y , such as course results.

3.3.3 Student Performance Prediction Approach (SPPA)

This research develops a Student Performance Prediction Approach (SPPA) to investigate the use of clickstream data in student performance prediction tasks (Figure 3. 1). The development of this approach is based on a commonly used predictive modelling process in data science. This approach is technical-and-educational-driven analytics for student performance prediction. Thereby, some steps in the modelling have educational considerations, including educational goals, instructional and course conditions, or student learning environments. The following points elaborate on each step.

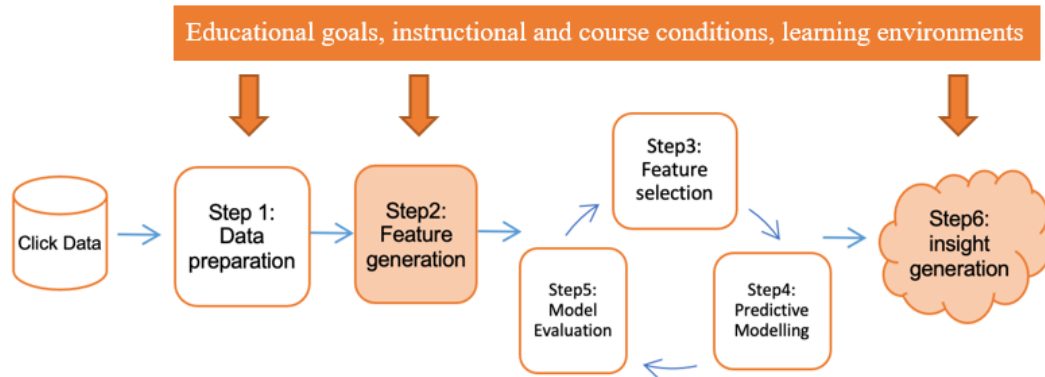


Figure 3. 1 Student Performance Prediction Approach (SPPA)

- *Step 1* is data preparation. After acquiring a set of raw clickstream data, it is necessary to integrate and pre-process the data to produce a target-aligned and clean dataset using data cleansing techniques. The educational consideration of this step involves understanding the course conditions and determining target students from the raw data.
- *Step 2* is feature generation. This step involves generating a form of click behaviour observations X by feature engineering. It includes using some appropriate methods for technically transforming the clickstream data. As for the educational perspective, when

generating sequential features, a typical period of the instructional cycle (e.g., week-based or month-based) can be considered. One instructional cycle may include establishing learning objectives, providing learning opportunities, assessing student learning, and then using the results from the assessment. Also, the learning environment affects feature engineering. For example, the course web pages are available before the course officially commences; whether this leads to a feature like early access can be considered.

- *Steps 3 to Step 5* focus on utilising feature selection methods and data mining algorithms to build predictive models, then evaluating the models. This process is purely technically driven. This process is repeated until the result of the model evaluation achieves the expectation. First, feature selection aims to improve the model performance by selecting a subset of features when training the model (Liu & Yu 2005). Predictive modelling involves using machine learning algorithms to build models, while model evaluation is related to analysing the models' performances.
- *Step 6* is insight generation. This step involves analysis of model performances and then generating insights. Insights of the practice of predictive modelling are generated based on analysing of the algorithms and datasets. Also, insight generation can be related to the educational goal of student performance prediction. For example, one of the goals is translating the predictive analysis findings into suggestions in terms of course design and teaching intervention to improve students' learning (Pardo, Han & Ellis 2017). The insights can be formed by analysing the dominant features of the models.

3.4 Research Methods

The presentation of the research methods aligns with the steps of SPPA (Figure 3. 1). First, aligning with *Step 1* of SPPA, this research conducts data preparation by selecting open-source data and pre-processing the data appropriately. Regarding *Step 2* of SPPA, feature engineering strategies are developed, generating a series of features and shapes of the datasets. For *Step 3* to *Step 5* of SPPA, feature selection, data mining algorithms, and model evaluation measures are determined. The experimental design of the process to involve all these aspects is also determined. *Step 6* of SPPA is associated with feature importance analysis.

Table 3. 1 The presentation of the research methods

# Step	Steps in SPPA	Research methods
<i>Step 1</i>	Data preparation	3.4.1 data selection 3.4.2 data pre-processing
<i>Step 2</i>	Feature generation	3.4.3 feature engineering strategies
<i>Step 3 to 5</i>	Feature selection Predictive modelling Model evaluation	3.4.4 feature selection 3.4.5 data mining algorithms 3.4.6 model evaluation methods 3.4.7 experimental design
<i>Step 6</i>	Insight generation	3.4.8 feature importance analysis

3.4.1 Data Selection

This research utilises open-source clickstream data OULAD (Open University Learning Analytics Dataset) from a distance-learning university - Open University (Kuzilek, Hlosta & Zdráhal 2017). The data contain information regarding 32,593 students in 7 courses (or called modules in Open University). There are 22 module presentations taught from 2013 to 2014, with 3 courses of Social Sciences and 4 courses of Science, Technology, Engineering, and Mathematics (STEM) (Table 3. 2). The data contain 10,655,280 click interaction events.

According to the description of OULAD, the data obey the Data Protection Policy and Policy on Ethical use of Student Data for Learning Analytics (Kuzilek, Hlosta & Zdráhal 2017). The data were collected anonymously, and all the students consented to their data being used for academic research.

Table 3. 2 OULAD course information (Kuzilek, Hlosta & Zdráhal 2017)

Module	Domain	Presentations	Students
AAA	Social Sciences	2	748
BBB	Social Sciences	4	7,909
CCC	STEM	2	4,434
DDD	STEM	4	6,272
EEE	STEM	3	2,934
FFF	STEM	4	7,762
GGG	Social Sciences	3	2,534

The Open University courses are represented in a Virtual Learning Environment (VLE) with typical online course structures (Kuzilek, Hlosta & Zdráhal 2017). Each course (called Module in

Open University) has multiple module presentations. A module presentation is a whole study period for a module. In Figure 3. 2, *VLE opens* means that the learning content was made available

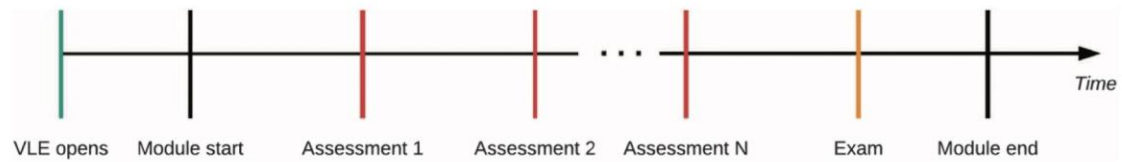


Figure 3. 2 The structure of a module presentation (Kuzilek, Hlosta & Zdráhal 2017)

online before the module’s official commencement – *Module start*. Each module presentation consists of a few formative assessments and a final exam. Formative assessments have two categories – Tutor Marked Assessment (TMA) and Computer Marked Assessment (CMA) (Kuzilek, Hlosta & Zdráhal 2017).

After understanding the learning environment, the raw dataset files are examined closely. The data reflect four aspects of students (see Figure 3. 3): registrations, assessments, VLE interactions, and demographics (Kuzilek, Hlosta & Zdráhal 2017). Demographics and VLE interactions align with the goal of this research. Demographics contains the classification label (*final result*). VLE interaction indicate click-based data. Formative assessments are parts of the course structure, spread throughout the course. Therefore, assessment data are used to understand the course condition. Registrations and its corresponding files are outside of the research scope, so they are excluded from this research.

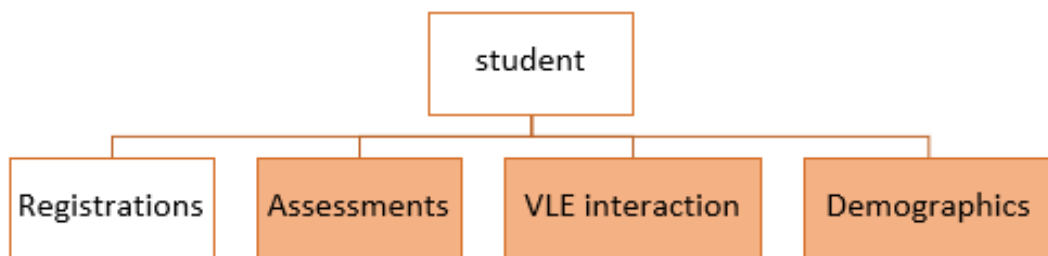


Figure 3. 3 Structure of OULAD (Kuzilek, Hlosta & Zdráhal 2017)

There are five csv files (datasets) related to demographics, VLE interactions and Assessments. They are *studentInfo*, *studentVle*, *vle*, *courses* and *assessments*. The datasets *studentInfo* and *studentVle* involve demographic information, the *final_result* column (label), and click behaviours that can be used for predictive modelling (Table 3. 3). The datasets *vle*, *courses* and *assessments*

involve the learning environment and course information, which are used to understand the context of the data (Table 3. 3), such as activity categories, course structure, and course length.

Table 3. 3 Details of the seven datasets (CSV files) (Kuzilek, Hlosta & Zdráhal 2017)

	Dataset name	Description	Data use for this research
1	<i>studentInfo</i>	students' demographics and their final results	for predictive modelling
2	<i>studentVle</i>	students' clickstream data	for predictive modelling
3	<i>vle</i>	VLE information	for understanding course conditions
4	<i>courses</i>	course information	for understanding course conditions
5	<i>assessments</i>	assessment information	for understanding course conditions
6	<i>studentAssessment</i>	students' assessment data	excluded for this research
7	<i>studentRegistration</i>	students' registration data	excluded for this research

Next, all the details of the five datasets are shown in Table 3. 4 to Table 3. 8. Also, five datasets screenshots are shown in Figure 3. 4 to Figure 3. 9. The relationships of the five datasets are identified and illustrated in Figure 3. 10.

Table 3. 4 Details of the *courses* dataset (Kuzilek, Hlosta & Zdráhal 2017)

#	Columns	Description	Data type
1	<i>code_module</i>	code name of the module, which serves as the identifier	nominal
2	<i>code_presentation</i>	code name of the presentation	nominal
3	<i>module_presentation_length</i>	the length of the module-presentation in days from module start date to module end date.	numerical

n.b. This table contains the list of all available modules and their presentations with 22 rows

code_module ▾	code_presentation ▾	module_presentation_length ▾
AAA	2013J	268
AAA	2014J	269
BBB	2013J	268
BBB	2014J	262
BBB	2013B	240
BBB	2014B	234
CCC	2014J	269
CCC	2014B	241
DDD	2013J	261
DDD	2014J	262
DDD	2013B	240
DDD	2014B	241
EEE	2013J	268
EEE	2014J	269
EEE	2014B	241
FFF	2013J	268
FFF	2014J	269
FFF	2013B	240
FFF	2014B	241
GGG	2013J	261
GGG	2014J	269
GGG	2014B	241

courses.csv (22 rows)

Figure 3. 4 A screenshot of the *courses* dataset

Table 3. 5 Details of the *vle* dataset (Kuzilek, Hlostá & Zdráhal 2017)

#	Columns	Description	Data type
1	<i>code_module</i>	the identification code for the module	nominal
2	<i>code_presentation</i>	the identification code of the presentation	nominal
3	<i>activity_type</i>	the role associated with the module material	nominal
4	<i>id_site</i>	the identification number of the material	numerical
5	<i>week_from</i>	the week from which the material is planned to be	numerical
6	<i>week_to</i>	the week until which the material is planned to be used	numerical

n.b. This table contains the materials available in the VLE, and the dataset consists of 6,364 rows

id_site	code_module	code_presentation	activity_type	week_from	week_to
703963	BBB	2013J	subpage	25	25
704233	BBB	2013J	url	16	16
703966	BBB	2013J	subpage	28	28
704236	BBB	2013J	url	22	22
704239	BBB	2013J	url	27	27
703943	BBB	2013J	subpage	1	1
703946	BBB	2013J	subpage	5	5
703949	BBB	2013J	subpage	9	9
703952	BBB	2013J	subpage	13	13
703955	BBB	2013J	subpage	16	16
703958	BBB	2013J	subpage	20	20
703961	BBB	2013J	subpage	23	23
704231	BBB	2013J	url	15	15
703964	BBB	2013J	subpage	26	26
704234	BBB	2013J	url	16	16
704237	BBB	2013J	url	23	23
704240	BBB	2013J	url	28	28
703944	BBB	2013J	subpage	2	2
703947	BBB	2013J	subpage	6	6

vle (6,364 rows)

Figure 3. 5 A screenshot of a part of the *vle* dataset

Table 3. 6 Details of the *studentVle* dataset (Kuzilek, Hlosta & Zdráhal 2017)

#	Columns	Description	Data type
1	<i>code_module</i>	the module identification code	nominal
2	<i>code_presentation</i>	the presentation identification code	nominal
3	<i>id_site</i>	id_site—the VLE material identification number	numerical
4	<i>id_student</i>	the unique student identification number	numerical
5	<i>date</i>	the day of student's interaction with the material	numerical
6	<i>sum_click</i>	the number of times the student interacted with the material	numerical

n.b. This table contains student's click-based interactions with the VLE with 10,655,280 rows

code_module ▾	code_presentation ▾	id_student ▾	id_site ▾	date ▾	sum_click ▾
BBB	2013J	2078479	703737	2	1
BBB	2013J	2056947	703737	2	1
BBB	2013J	2164944	703737	2	1
BBB	2013J	1411627	703737	2	1
BBB	2013J	1421720	703737	2	1
BBB	2013J	1421720	703737	2	1
BBB	2013J	1421720	703737	2	1
BBB	2013J	1536774	703737	2	1
BBB	2013J	1536774	703737	2	1
BBB	2013J	1624707	703737	2	1
BBB	2013J	1624707	703737	2	1
BBB	2013J	2239140	703737	2	1
BBB	2013J	2273256	703737	2	1
BBB	2013J	2318763	703737	2	1
BBB	2013J	2320306	703737	2	1
BBB	2013J	2299338	703737	2	1
BBB	2013J	2280314	703737	2	1
BBB	2013J	2473780	703737	2	1
studentVle (10,655,280 rows)					

Figure 3. 6 A screenshot of a part of the *studentVle* dataset

Table 3. 7 Details of the *studentInfo* dataset (Kuzilek, Hlosta & Zdráhal 2017)

#	Columns	Description	Data type
1	<i>code_module</i>	module identification code on which the student is registered	nominal
2	<i>code_presentation</i>	presentation identification code during which the student is registered on the module.	nominal
3	<i>id_student</i>	the unique student identification number	numerical
4	<i>gender</i>	student's gender	
5	<i>region</i>	the geographic region, where the student lived while taking the module-presentation	nominal
6	<i>highest_education</i>	the highest student education level on entry to the module presentation	nominal
7	<i>imd_band</i>	the IMD band of the place where the student lived during the module-presentation.	nominal
8	<i>age_band</i>	a band of student's age	nominal
9	<i>disability</i>	indicates whether the student has declared a disability	nominal
10	<i>num_of_prev_attempts</i>	the number of times the student has attempted this module	numerical
11	<i>studied_credits</i>	the total number of credits for the modules the student is currently studying	numerical
12	<i>final_result</i>	student's final result in the module-presentation	nominal
n.b. This table contains student demographics and their results in each module with 32,593 rows			

code_module	code_presentation	id_student	gender	region	highest_education
BBB	2014J	41547	F	Yorkshire Region	A Level or Equivalent
BBB	2014J	57285	F	London Region	A Level or Equivalent
BBB	2014J	106095	F	North Western Region	A Level or Equivalent
BBB	2014J	226839	F	South Region	A Level or Equivalent
BBB	2014J	230383	F	Yorkshire Region	A Level or Equivalent
BBB	2014J	294134	F	East Anglian Region	A Level or Equivalent
BBB	2014J	314675	F	North Western Region	A Level or Equivalent
BBB	2014J	368521	F	North Western Region	A Level or Equivalent
BBB	2014J	395822	F	South West Region	A Level or Equivalent
BBB	2014J	403213	F	South Region	A Level or Equivalent
BBB	2014J	444768	F	London Region	A Level or Equivalent
BBB	2014J	473271	F	South East Region	A Level or Equivalent
BBB	2014J	476769	F	South East Region	A Level or Equivalent
BBB	2014J	484252	F	Yorkshire Region	A Level or Equivalent
BBB	2014J	492979	F	East Anglian Region	A Level or Equivalent
BBB	2014J	499819	F	Yorkshire Region	A Level or Equivalent
BBB	2014J	507014	F	East Anglian Region	A Level or Equivalent
BBB	2014J	509578	F	West Midlands Region	A Level or Equivalent
BBB	2014J	543439	F	South Region	A Level or Equivalent

studentInfo (32,593 rows)

Figure 3. 7 A screenshot of a part of the *studentInfo* dataset (part A)

imd_band	age_band	num_of_prev_attempts	studied_credits	disability	final_result
40-50%	0-35	0	60	N	Pass
0-10%	0-35	0	60	N	Pass
70-80%	0-35	0	60	N	Pass
90-100%	0-35	0	60	N	Pass
60-70%	0-35	0	60	N	Pass
40-50%	0-35	0	60	N	Pass
70-80%	0-35	0	60	N	Pass
40-50%	0-35	0	60	N	Pass
40-50%	0-35	0	60	N	Pass
20-30%	0-35	0	60	N	Pass
50-60%	0-35	0	60	N	Pass
60-70%	0-35	0	60	N	Pass
70-80%	0-35	0	60	N	Pass
0-10%	0-35	0	60	N	Pass
60-70%	0-35	0	60	N	Pass
10-20	0-35	0	60	N	Pass
80-90%	0-35	0	60	N	Pass
20-30%	0-35	0	60	N	Pass
80-90%	0-35	0	60	N	Pass

Figure 3. 8 A screenshot of a part of the *studentInfo* dataset (part B)

Table 3. 8 DetailsSS of the *assessments* dataset (Kuzilek, Hlosta & Zdráhal 2017)

#	Columns	Description	Data type
1	<i>code_module</i>	module identification code, to which the assessment belongs	nominal
2	<i>code_presentation</i>	presentation identification code, to which the assessment belongs	nominal
3	<i>id_assessment</i>	assessment identification number	numerical
4	<i>assessment_type</i>	a type of assessment. Three types of assessments exist—Tutor Marked Assessment (TMA), Computer Marked Assessment (CMA) and Final Exam (Exam).	nominal
5	<i>date</i>	information about the cut-off day of the assessment.	numerical
6	<i>weight</i>	the weight of the assessment. Typically, Exams are treated separately and have the weight equal to 100%; the sum of all other assessments is also 100%.	numerical

n.b. This table contains assessments in module-presentations. The table consists of 206 rows

code_module	code_presentation	id_assessment	assessment_type	date	weight
AAA	2014J	1759	TMA	54	20
AAA	2014J	1760	TMA	117	20
AAA	2014J	1761	TMA	166	20
AAA	2014J	1762	TMA	215	30
AAA	2014J	1763	Exam		100
BBB	2013B	14991	CMA	54	1
BBB	2013B	14992	CMA	89	1
BBB	2013B	14993	CMA	124	1
BBB	2013B	14994	CMA	159	1
BBB	2013B	14995	CMA	187	1
BBB	2013B	14984	TMA	19	5
BBB	2013B	14985	TMA	47	18
BBB	2013B	14986	TMA	89	18

assessments.csv (206 rows)

Figure 3. 9 A screenshot of a part of the *assessments* dataset

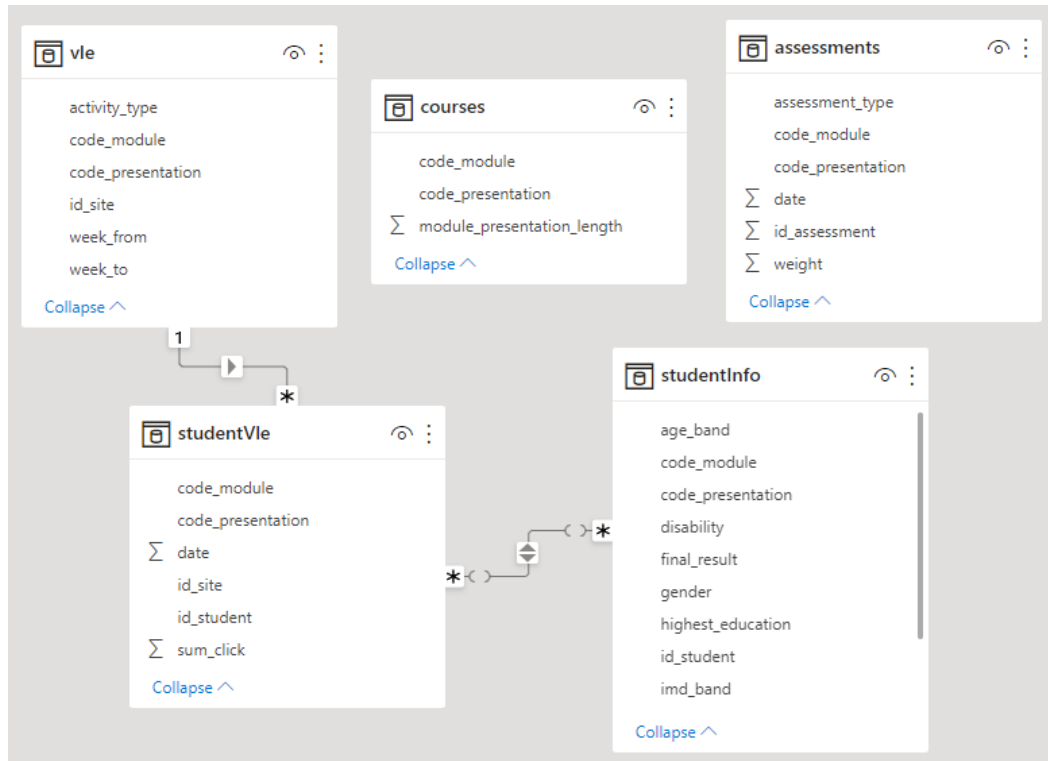


Figure 3. 10 Relationship of *assessments*, *courses*, *vle*, *studentVle* and *studentInfo* datasets
(Kuzilek, Hlosta & Zdráhal 2017)

After assessing and cleaning the data, it was determined that *single course data*, rather than data from all seven courses of OULAD, should be used for the research. To select the course with the biggest dataset, a selection method was conducted with the following three steps:

- there are four classes in the label column of *studentInfo* - *Withdrawn*, *Fail*, *Pass*, and *Distinction*. Students with the *withdrawn* label are excluded as they are not related to the purpose of this research.
- the number of non-withdrawn students is compared among seven courses (Table 3. 9).
- the course BBB is selected for this research as it has the greatest number of non-withdrawn students (5521 students).

Based on the above, the course BBB data are selected, containing 5521 students with three labels *Fail*, *Pass*, and *Distinction*. The course BBB's activity categories include *forumng*, *oucontent*, *subpage*, *homepage*, *quiz*, *resource*, *url*, *oucollaborate*, *questionnaire*, *ouelluminate*, *glossary* and *sharedsubpage*. Also, the course BBB's assessment data are summarised in Figure 3. 11.

Table 3. 9 Comparison of the student number among seven courses

Courses	Module Presentation	Length of presentation (days)	Number of students	of Withdrawn students	Non-withdrawn students
AAA	2013J	268	383	60	323
	2014J	269	365	66	299
	Total		748	126	622
BBB	2013J	268	2237	644	1593
	2014J	262	2292	749	1543
	2013B	240	1767	505	1262
	2014B	234	1613	490	1123
	Total		7909	2388	5521
CCC	2014J	269	2498	1077	1421
	2014B	241	1936	898	1038
	Total		4434	1975	2459
DDD	2013J	261	1938	681	1257
	2014J	262	1803	647	1156
	2013B	240	1303	432	871
	2014B	241	1228	490	738
	Total		6272	2250	4022
EEE	2013J	268	1052	243	809
	2014J	269	1188	306	882
	2014B	241	694	173	521
	Total		2934	722	2212
FFF	2013J	268	2283	675	1608
	2014J	269	2365	855	1510
	2013B	240	1614	411	1203
	2014B	241	1500	462	1038
	Total		7762	2403	5359
GGG	G2013J	261	952	66	886
	G2014J	269	749	126	623
	G2014B	241	833	100	733
	Total		2534	292	2242

Assessment	weight	cut off week #		
		2013B	2013J	2014B
CMA	1	8	8	7
CMA	1	13	14	12
CMA	1	18	19	17
CMA	1	23	24	22
CMA	1	27	30	28
TMA	5	3	3	2
TMA	18	7	7	6
TMA	18	13	14	12
TMA	18	18	19	17
TMA	18	23	24	22
TMA	18	27	30	28

Assessment	weight	cut off week #
		2014J
TMA	0	3
TMA	10	8
TMA	20	16
TMA	35	22
TMA	35	29

Summary:
 Stage 1: 6-8 (CMA + TMA)
 Stage 2: 12-14 (CMA + TMA)
 Stage 3: 17-19 (CMA + TMA)
 Stage 4: 22-24 (CMA + TMA)
 Stage 5: 27-30 (CMA + TMA)

Figure 3. 11 Assessment data summary of the course BBB

3.4.2 Data Pre-processing

From this section, the course BBB data are further pre-processed. The data from datasets *studentVle* and *studentInfo* are processed by statistical analysis and data cleansing. The dataset *studentVle* is identified as a clean dataset with no missing values. Next, the *studentInfo* dataset is pre-processed as follows:

- According to the definition of the binary classification problem, two label classes are required. Therefore, the original labels are changed (see Figure 3. 12). Specifically, the label *distinction* is replaced with *pass* for the goal of the binary classification of this research. As a result, in the total of 5521 students, 68% (3754) are labelled *pass*, and 32% (1767) students are labelled *fail*.

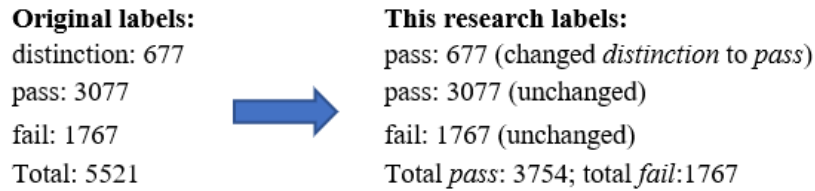


Figure 3. 12 Label changes of the course BBB *studentInfo* dataset

- The course BBB consists of four module presentations - 2013J, 2014J, 2013B and 2014B. From the dataset *studentInfo*, it is observed that some *id_student* codes appear in multiple module presentations. In other words, some students repeatedly studied the course BBB with different module presentations. To recognise each student in each module presentation, a new unique id (called *id* in this research) is created by concatenating *code_module*, *code_presentation*, and *id_students*.
- The column *imd-band* of the table *studentInfo* contains 50 missing values, which are replaced with the mean value.
- It is found that 180 students are included in the table *studentInfo* but excluded from the table *studentVle*. That means these 180 students have no click behaviours recorded in the table *studentVle*, although their demographics are included in the dataset. These 180 samples cannot be used for the research, so they are discarded.

In summary, 5341 students were used for the research. To better understand the student samples involved, the distribution of demographics is illustrated in Figure 3. 13.

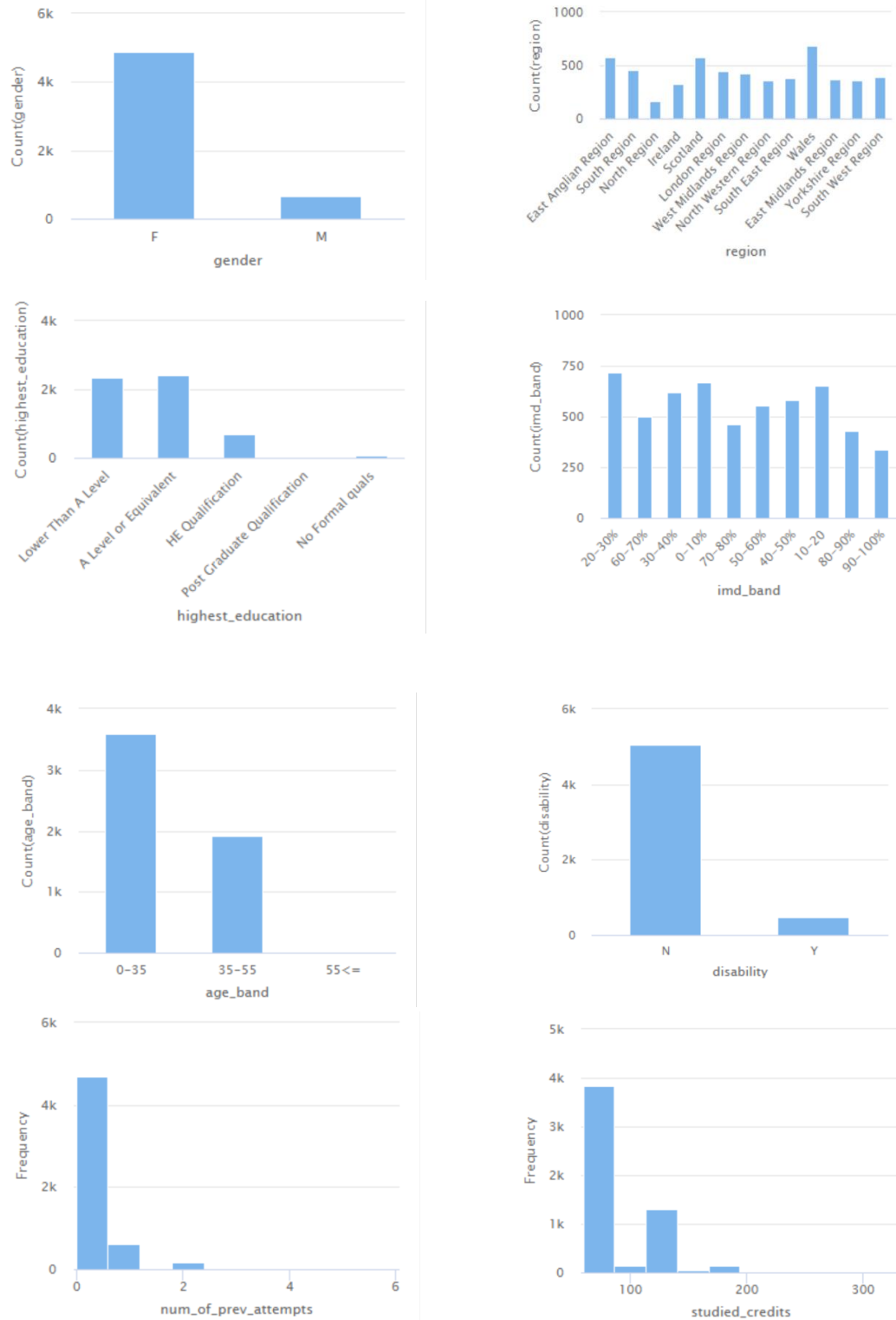


Figure 3. 13 Distribution of demographics

3.4.3 Feature Engineering Strategies

The goal of the feature engineering of this research is to transform the raw clickstream data into click behaviours, X , that can be used to find the function $f: X \rightarrow Y$. In the dataset *studentVle*, the features *date* and *id_site* indicate two dimensions of the click behavioural data – time (*date*: the date students clicked) and activity (*id_site*: the site category students clicked on). Before determining if time and activity category dimensions can be used for generating X , some preparation work was conducted. The preparation includes (1) transforming the data with the targeted dimension involved; and (2) determining if explicit behavioural patterns exist between two classes (pass/fail) when the targeted dimension is involved. This section describes the preparation work by closely exploring the table *studentVle* and *vle*, followed by describing the three feature engineering strategies used in this research.

Preparation of the Time Dimension

In the dataset *studentVle*, each row records click actions of a student on a date. For such data, time-based aggregation is appropriate. This research defines T as a series of time periods, $T = \{T_0, T_1, \dots, T_t\}$, and t refers to the number of periods (i.e., # time period). This research keeps using the original measures - click count (i.e., the number of clicks or # clicks) during the length of the course.

Next, the aggregation methods are determined. This is related to how to aggregate time-based click behaviours. Before the course starts, some students took the initiative to start clicking while other students did not. After the commencement date of the course, all students are expected to have click actions. Therefore, T is divided into two parts:

- the time period before the course officially commences (T_0): students' click actions in T_0 indicate that students have early access to the course. The number of clicks in T_0 in the raw data are demonstrated in negative numbers, and the number zero (-25, -24, ..., 0). For example, -25 means the 25 days prior to the start of the course. The number of clicks in T_0 is calculated by adding all the clicks from day -25 to day 0.
- the time period after the date of the course officially commences (T_1, \dots, T_t): the click count is aggregated in two ways - by week and by month. As a result, click behavioural views are generated weekly and monthly. The generation of different views aims to investigate which one is more likely to obtain the best prediction result in models. For the weekly

view, $t = 39$, meaning that the course BBB contains 39 weeks. For monthly view, $t = 9$, meaning that the course BBB contains 9 months.

After determining time-based aggregation methods, it is crucial to examine whether there are pattern differences between passed and failed students. The data are transformed then visualised with the following four steps:

- Step 1: a temporary dataset is generated, named *clickRecords* (Figure 3. 14). It has four features (*id*, *activity_type*, *date*, *sum_click*) and one label. The feature *activity_type* has 12 values referring to the 12 types of VLE sites of the course BBB that students click on in the course (Figure 3. 15). The dataset *clickRecords* is generated by merging *studentVle*, *vle*, and the label column from *studentInfo* together using *id_site* and *id* as the keys.

Row No.	id	activity_type	date	sum_click	final_result
1	BBB_2013B_1008675	forumng	-5	1	pass
2	BBB_2013B_1008675	homepage	-5	1	pass
3	BBB_2013B_1008675	forumng	-5	7	pass
4	BBB_2013B_1008675	forumng	-5	3	pass
5	BBB_2013B_1008675	forumng	-5	1	pass

Figure 3. 14 A screenshot of a part of the temporary dataset *clickRecords*

<i>Act1: forumng</i>	<i>Act2: oucontent</i>	<i>Act3: subpage</i>	<i>Act4: homepage</i>
<i>Act5: quiz</i>	<i>Act6: resource</i>	<i>Act7: wrl</i>	<i>Act8: oucollaborate</i>
<i>Act9: questionnaire</i>	<i>Act10: ouelluminate</i>	<i>Act11: glossary</i>	<i>Act12: sharedsubpage</i>

Figure 3. 15 The course BBB's 12 activity categories

- Step 2: the temporary dataset *clickRecords* is further transformed to be a new dataset called *studentWeeklyClick*. A screenshot of a part of this dataset is shown in Figure 3. 16. This

Row No.	week0	week1	week2	week3	week4	week5	week6	week7	week39	final_result
1	13	0	0	38	0	0	0	48	0	pass
2	7	25	6	1	10	16	4	2	0	pass
3	71	96	3	117	83	218	303	1	0	fail
4	117	21	36	3	1	24	120	0	0	pass
5	2	1	0	8	0	0	4	0	0	fail

Figure 3. 16 A screenshot of a part of the *studentWeeklyClick* dataset

is a time-based dataset structure with a weekly view. The transformation process aggregates click counts every 7 days. There are 40 features in *studentWeeklyClick*, *week0* to *week39* (click counts from week 0 to week 39), and the label (*pass* or *fail*).

- Step 3: *studentWeeklyClick* is visualised using a bar chart (Figure 3. 17). As can be seen, students with pass and fail have different click behaviour patterns over time.

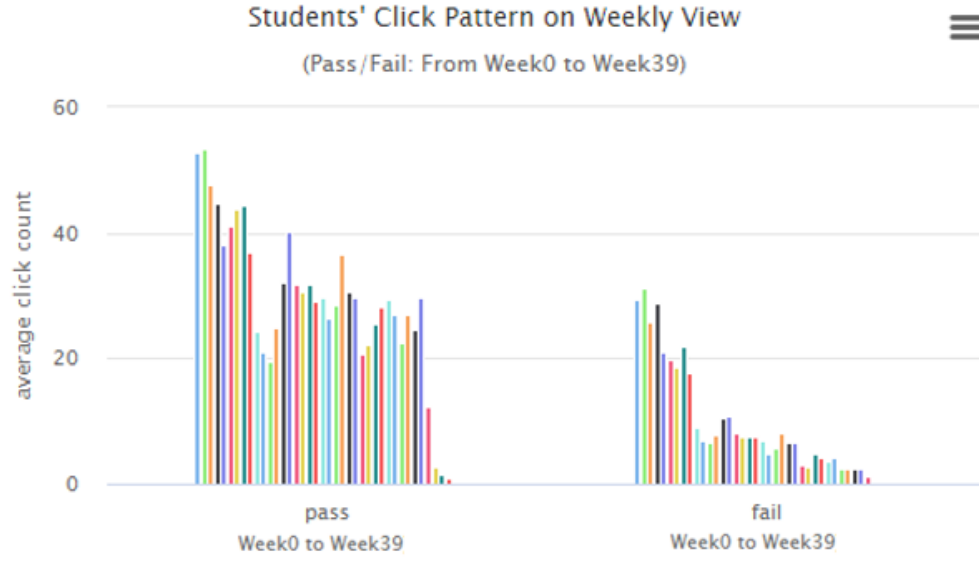


Figure 3. 17 Pattern differences generated by the *studentMonthlyClick* dataset

- Step 4: Using a similar method, the dataset *studentMonthlyClick* is generated by aggregating click numbers every 30 day to form the monthly view (see Figure 3. 18). These patterns are also illustrated with a bar chart (Figure 3. 19), showing a similar behavioural difference between passed and failed students.

Row No.	month0	month1	month2	month3	month4	month5	month6	month7	month8	month9	final_result
1	13	38	88	14	126	67	124	152	83	34	pass
2	7	42	22	7	101	46	14	62	0	0	pass
3	71	333	510	38	139	247	119	157	62	12	fail
4	117	72	211	131	244	483	193	409	138	0	pass
5	2	9	4	0	0	1	0	0	0	0	fail

Figure 3. 18 A screenshot of a part of the *studentMonthlyClick* dataset

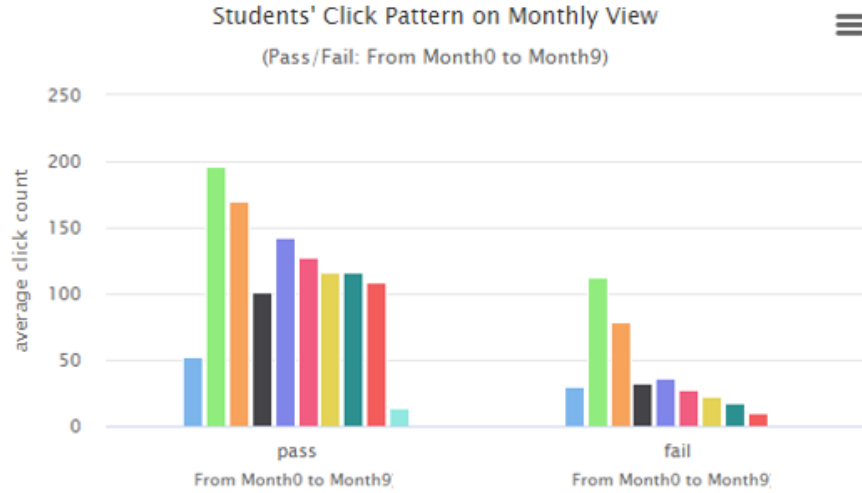


Figure 3. 19 Pattern differences generated by the *studentMonthlyClick* dataset

To sum up, temporal-based analyses in weekly and monthly views illustrate clear pattern differences between passed and failed students. Therefore, the time dimension can be developed as a form of X to reflect more informative values of click-behaviour patterns.

Preparation of the Activity Dimension

To examine whether the activity dimension benefits development as a form of X, a collection of activity categories is defined as Act, $Act = \{Act_1, Act_2, \dots, Act_v\}$, where $v = 12$, meaning that there are 12 activity categories in the course. Then, the following two steps examine pattern differences between passed and failed students from the activity categories perspective:

- Step 1, the *clickRecords* dataset is transformed into the dataset *activityClickRecords* (see a screenshot in Figure 3. 20) by aggregating the click number of each student on each activity type on each day. Act₁, Act₂, ..., and Act₁₂ are used as the names of the 12 activity features in the dataset *activityClickRecords*.

Row No.	id	date	Act1	Act2	Act3	Act4	Act5	Act6	Act7	Act8	Act9	Act10	Act11	Act12
1567534	BBB_2014J_2035574	262	0	0	0	1	0	0	0	0	0	0	0	0
1567535	BBB_2014J_2692969	262	0	0	0	1	0	0	0	0	0	0	0	0
1567536	BBB_2014J_2634319	262	0	1	0	0	0	0	0	0	0	0	0	0
1567537	BBB_2014J_2634319	262	0	28	0	0	0	0	0	0	0	0	0	0
1567538	BBB_2014J_2634319	262	0	1	0	0	0	0	0	0	0	0	0	0

Figure 3. 20 A screenshot of a part of the dataset *activityClickRecords*

- Step 2 is to merge the label column then generates graphs of click behaviour changes over time in 12 activity categories (Figure 3. 21 and Figure 3. 22). As can be seen, click behaviours on some activity categories show different patterns between students who passed and failed, such as *forumng*, *oucontent*, *subpage*, *homepage*, *quiz*. Therefore, activity category is another significant aspect that can be used to demonstrate informative patterns when generating X.

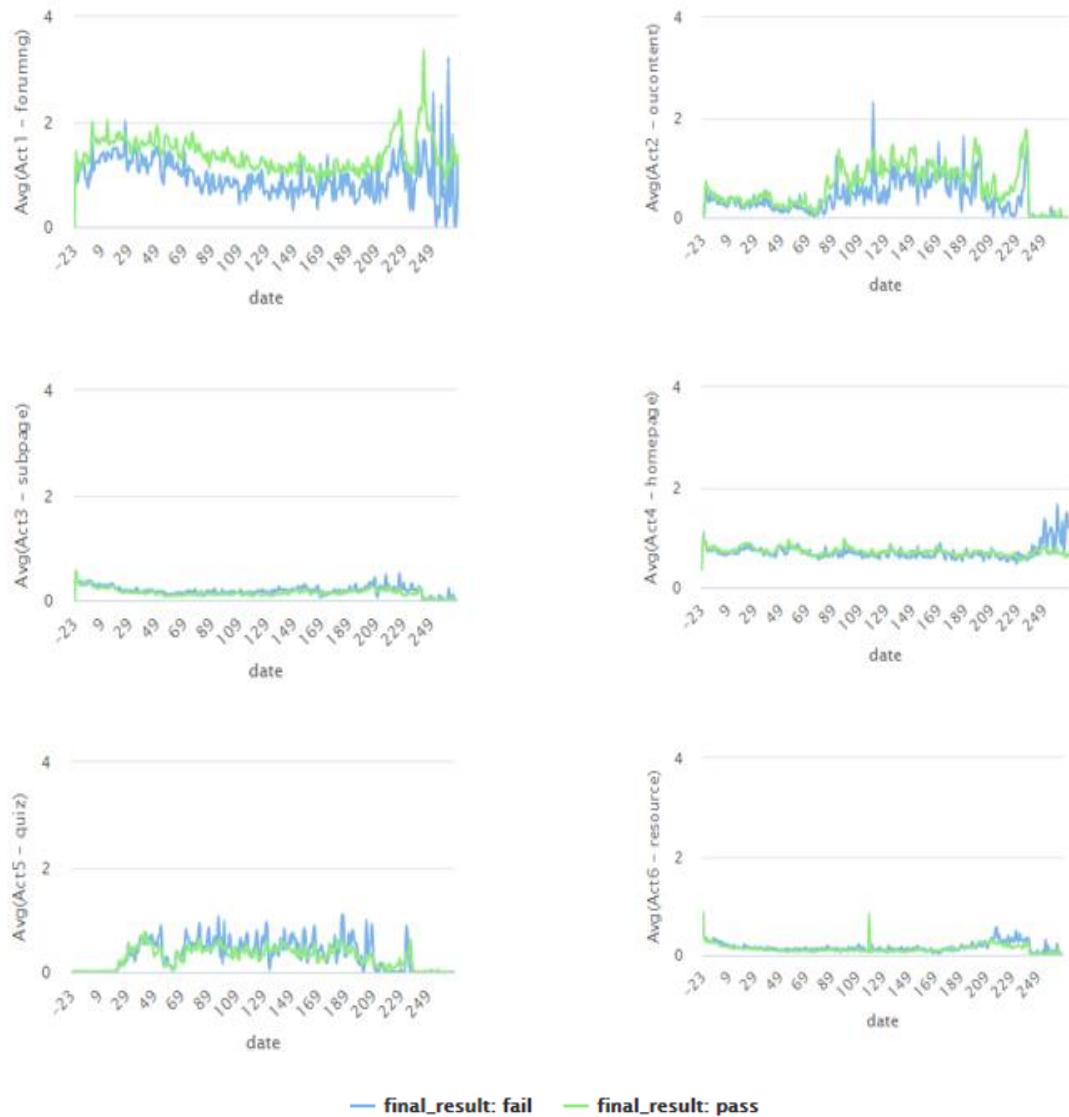


Figure 3. 21 Visualisation of students' click patterns on activity categories (part A)

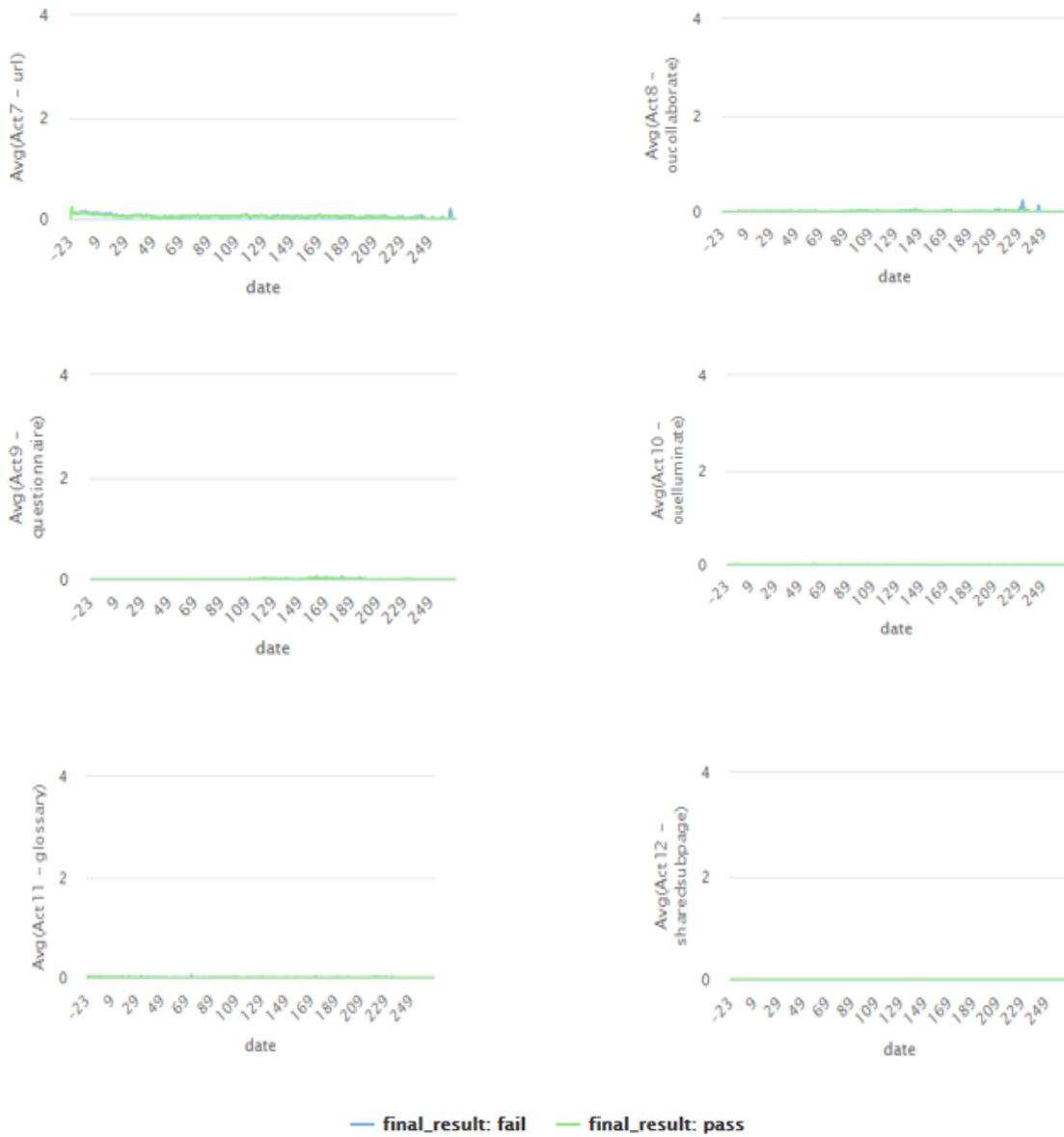


Figure 3.22 Visualisation of students' click patterns on activity categories (part B)

Based on all of the above, both time-based and activity category-based dataset structures demonstrate different patterns between passed and failed students. Therefore, this research posits that both the time and the activity dimensions are worth to be involved in the input data for prediction modelling.

Next, this research develops three feature engineering strategies through transforming data based on time and activity dimensions.

Feature Engineering Strategy 1

Strategy 1 considers using only time dimensions (without using the activity dimension) to transform the data into a structure of $X^{(T)}$, where (T) is a vector of the time period. Therefore, s_i is x_i that can be described as

$$s_i = x_i^{(T)} = \{x_i^{T_0}, x_i^{T_1}, \dots, x_i^{T_t}\}$$

In the equation, $x_i^{T_0}$ indicates the i^{th} student's click behaviour in T_0 ; $x_i^{T_1}$ indicates the i^{th} student's click behaviour in T_1 , etc. The dataset structure of this strategy is illustrated in Figure 3. 23. Next, integrating this strategy with the weekly and monthly views of T , two datasets are created, named T-WEE and T-MON (see details of the two datasets in Table 3. 10). In T-WEE, $t = 40$ (i.e., 40 features) and $n = 5341$ (i.e., 5341 samples or students). In T-MON, $t = 10$ (i.e., 10 features) and $n = 5341$ (i.e., 5341 samples or students). This strategy $X^{(T)}$ characterises a series of temporal features.

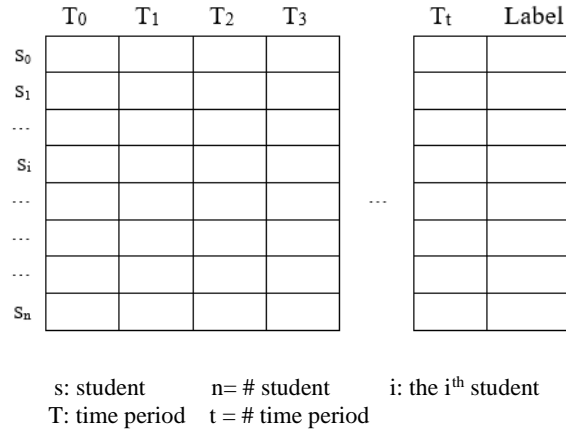


Figure 3. 23 Dataset structure of Strategy 1 - $X^{(T)}$

Table 3. 10 Details of the Strategy 1 datasets: T-WEE and T-MON

Dataset	Samples	Features	Shape
T-WEE	s0	T_0 : the number of clicks before the course start	5,341 * 40
	s1	T_1 : the number of clicks of the first week	
	
	s5340	T_{39} : the number of clicks of the 39 th week	
T-MON	s0	T_0 : the number of clicks before the course start	5,341 * 10
	s1	T_1 : the number of clicks of the first month	
	
	s5340	T_9 : the number of clicks of the 9 th month	

T-WEE and T-MON are formed by transforming the dataset *clickRecords*. The transforming process involves the following steps:

- pivot the temporary dataset *clickRecords* to generate the following dataset structure (Figure 3. 24) (grouping by ‘*id*’, grouping ‘*date*’, using ‘sum’ aggregation function). The pivoting leads to missing values, indicating that students had no clicks on some days. These missing values are replaced with 0;

	day-25	day-24	day0	day1		day267	day268
S_0							
S_1							
...						
S_n							

Figure 3. 24 Daily-based dataset structure

- generate T_0 : sum up the click numbers from day -25 to day 0 for both weekly-view and monthly-view generate T_1 to T_t : for weekly-view and monthly-view, aggregate click numbers every 7 days and 30 days respectively for T-WEE and T-MON.

Feature Engineering Strategy 2

Strategy 2 considers using both time and activity dimensions to transform the data into a structure of $X^{(TA)}$, where (TA) refers to a vector of the combination of time and activity category. Therefore, s_i , referring to the i^{th} student, is x_i that can be described as follows:

$$s_i = x_i^{(TA)} = \{x_i^{T_0Act_1}, \dots, x_i^{T_0Act_v}, \dots, x_i^{T_tAct_1}, \dots, x_i^{T_tAct_v}\}$$

In the equation, $x_i^{T_0Act_1}$ indicates the i^{th} student’s click behaviour in T_0 on the site that is related to Act_1 etc. The structure of $X^{(TA)}$ is shown in Figure 3. 25. Compared with $X^{(T)}$, the activity category dimension is integrated into features combined with the time dimension. The strategy $X^{(TA)}$ also characterises a series of temporal features. With weekly and monthly views of this strategy, two datasets called TA-WEE, and TA-MON are created, and their details are shown in

Table 3. 11. The dataset TA-WEE has 480 features, while the dataset TA-MON has 120 features. The sample number of both datasets remains 5341.

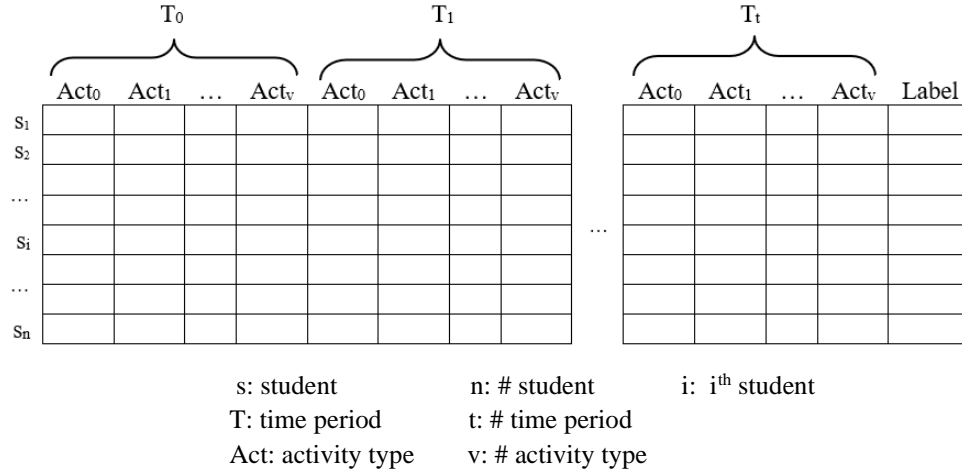


Figure 3. 25 Dataset structure of Strategy 2 - $X^{(\text{TA})}$

Table 3. 11 Details of the Strategy 2 datasets: TA-WEE and TA-MON

Dataset	Samples	Features	Shape
TA-WEE	S_0	$T_0\text{Act}_1$: # clicks in T_0 on activity1	5,341 * 480
	S_1	$T_0\text{Act}_2$: # clicks in T_0 on activity2	
	
	S_{5340}	$T_0\text{Act}_{12}$: # clicks in T_0 on activity12	
		$T_1\text{Act}_1$: # clicks in T_1 on activity1	
		$T_1\text{Act}_2$: # clicks in T_1 on activity2	
		...	
		$T_1\text{Act}_{12}$: # clicks in T_1 on activity12	
		$T_{39}\text{Act}_1$: # clicks in T_{39} on activity1	
		$T_{39}\text{Act}_2$: # clicks in T_{39} on activity2	
TA-MON	S_0	$T_0\text{Act}_1$: # clicks in T_0 on activity1	5,341 * 120
	S_1	$T_0\text{Act}_2$: # clicks in T_0 on activity2	
	
	S_{5340}	$T_0\text{Act}_{12}$: # clicks in T_0 on activity12	
		$T_1\text{Act}_1$: # clicks in T_1 on activity1	
		$T_1\text{Act}_2$: # clicks in T_1 on activity2	
		...	
		$T_1\text{Act}_{12}$: # clicks in T_1 on activity12	
		$T_9\text{Act}_1$: # clicks in T_9 on activity1	
		$T_9\text{Act}_2$: # clicks in T_9 on activity2	
		...	
		$T_9\text{Act}_{12}$: # clicks in T_9 on activity12	

TA-WEE, TA-MON are directly developed from the dataset *activityClickRecords*. The development process involves the following steps:

- for Act1, filter T0 data (from day -25 to day 0), then pivot data (group by ‘id’, grouping the column ‘date’, aggregation feature is *activity_type_1*, using ‘sum’ aggregation function), and replace all missing values with 0;
- use the same pivot method to deal with every 7 days and 30 days of the data. The output is weekly-view and monthly-view on Act1;
- repeat the first two points for the rest of the activity types (Act2 to Act12), merge all subsets of 12 activity types;
- merge with the label column of *studentInfo*, resulting in TA-WEE, TA-MON.

Feature Engineering Strategy 3

Strategy 3 considers transforming the dataset into a balanced panel dataset. The data structure of this strategy is illustrated in Figure 3. 26. A balanced panel refers to a structure where each panel member (i.e., student) is observed in a regular T period. This strategy transforms the data into a structure of $X^{(T) \times (A)}$, where $(T) \times (A)$ indicates a matrix consisting of the vector of T multiplying the vector of Act. Therefore, s_i is x_i that can be described as follows:

$$s_i = x_i^{(T) \times (A)} = \{x_i^{(T_0 T_1 \dots T_t) \times (Act1)}, x_i^{(T_0 T_1 \dots T_t) \times (Act2)}, \dots, x_i^{(T_0 T_1 \dots T_t) \times (Actv)}\},$$

where $x_i^{(T_0 T_1 \dots T_t) \times (Act1)}$ indicates the i^{th} student’s click behaviours on Act₁ during the whole time period, from T₀ to T_t. The balanced panel contains n students and t periods; therefore, the number

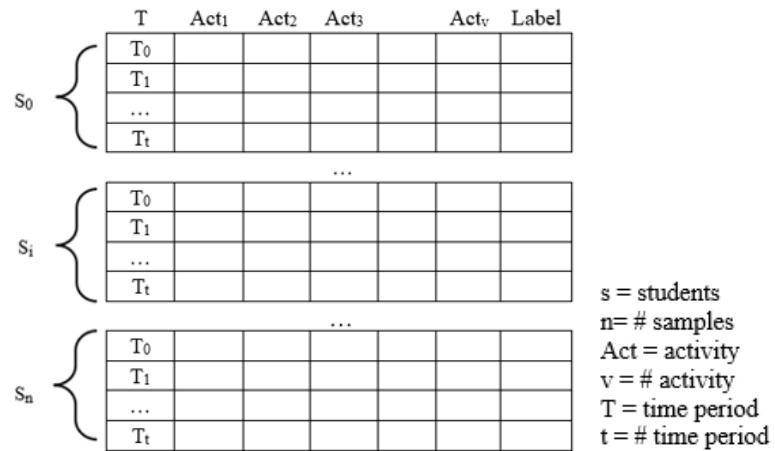


Figure 3. 26 Dataset structure of Strategy 3 - balanced panel data $X^{(T) \times (A)}$

of observations is $n \times t$. With weekly and monthly views of T , two panel datasets, P-WEE and P-MON, are generated. Their details are in Table 3. 12. Differentiated from $X^{(TA)}$, which integrates the time and activity dimensions into one vector on features, the strategy $X^{(T) \times (A)}$ integrates the time dimensions on samples. Therefore, this strategy has a sparser representation of samples. The total of 5341 sample students are extended to be 213,640 observations in the weekly view (P-WEE), and 53410 observations in the monthly view (P-MON). The rationale of the strategy is to avoid high-dimensional datasets while keeping both time and activity dimensions. Also, compared with the first two strategies showing the temporal features, this strategy leads to temporal observations within each student's subset.

Table 3. 12 Details of strategy 3 datasets: P-WEE and P-MON

Data name	Samples	Features	Shape of Matrix
P-WEE	<div> <div>S0 S0 S0</div> <div>}</div> <div>40 rows</div> </div>	<div> <div>T: # T period</div> <div>Act1: # clicks on Activity1</div> <div>Act2: # clicks on Activity2</div> <div>Act3: # clicks on Activity3</div> <div>Act4: # clicks on Activity4</div> </div>	213,640 * 12
	<div> <div>S5340 S5340 S5340 ...</div> <div>}</div> <div>40 rows</div> </div>	<div> <div>...</div> <div>Act12: # clicks on Activity12</div> </div>	
P-MON	<div> <div>S0 S0 S0</div> <div>}</div> <div>10 rows</div> </div>	<div> <div>T: # T period</div> <div>Act1: # clicks on Activity1</div> <div>Act2: # clicks on Activity2</div> <div>Act3: # clicks on Activity3</div> <div>Act4: # clicks on Activity4</div> </div>	53,410 * 12
	<div> <div>S5340 S5340 S5340 ...</div> <div>}</div> <div>10 rows</div> </div>	<div> <div>...</div> <div>Act12: # clicks on Activity12</div> </div>	

P-WEE and P-MON are created by transforming the dataset *activityClickRecords*. The generation process involves the following steps:

- aggregate all negative and zero records on each date of the dataset *activityClickRecords* from samples, forming a T_0 feature for both the weekly and monthly view datasets;
- aggregate each week of the dataset *activityClickRecords* from samples, forming T_1 - T_t features for the weekly view dataset;
- aggregate by month of the dataset *activityRecords* from samples, forming T_1 - T_t features for the monthly view dataset;

- merge T_0 and T_1-T_t of the weekly and monthly views and the label column of *studentInfo*, resulting in P-WEE and P-MON.

To conclude, three strategies are developed in this section, resulting in 6 datasets generated for later predictive modelling. Strategy 1 utilises the time dimension (without utilising the activity dimension). Strategies 2 and 3 utilise both time and activity dimensions of the clickstream data. Their sequential characteristics are compared in Table 3. 13.

Table 3. 13 Feature engineering strategies summary

Strategies	Structure	Dataset	Sequential
Strategy 1	$X^{(T)}$	T-WEE, T-MON	Temporal features
Strategy 2	$X^{(TA)}$	TA-WEE, TA-MON	Temporal features
Strategy 3	$X^{(T) \times (A)}$	P-WEE, P-MON	Panel data, temporal observations in each panel member

3.4.4 Feature Selection

Feature Selection Methods

Feature selection is one of the methods used to boost the models' performances (Rangkuti et al. 2018). The subset of features that lead to better model performance need to be evaluated. The filter method is the way to conduct the evaluation (Yu & Liu 2003). For the data used in this research, filter methods use statistical principles to evaluate a subset of features by measuring the relevance of features by their correlation with the label (Zhou et al. 2017). These methods are an independent process separated from the model training process. This research adopts three statistical methods: Information Gain, Pearson's Correlation and Principal Component Analysis (PCA).

Information Gain is an entropy-based method that can be used in feature evaluation (Lei 2012). Information gain used as a filter method referring to what extent of a feature can be used for classification (Lei 2012). Using the Information Gain statistical method, the relevance with the label of a feature is calculated and represented as a weight score (i.e., the value of Information Gain). The higher the weight score of a feature, the more relevant it is.

Correlation or Pearson's Correlation is also used as a feature evaluation method in this research due to its advantage of enhancing the model performance (Rangkuti et al. 2018). The principle is

to quantify the linear dependence of two continuous variables X and Y, using the formula below (Rangkuti et al. 2018):

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}$$

The correlation value is in a range from -1 to +1, where positive values indicate positive correlation, while negative values indicate inverse correlation (Rangkuti et al. 2018).

Principal Component Analysis (PCA), as a numerical analysis method, can be used into the feature selection process of classification (Song, Guo & Mei 2010). It is able to select important feature components from a dataset and transform high dimensional samples into low dimensional samples leading to accuracy improvement of the models (Song, Guo & Mei 2010). Involving the PCA in the filter method generates feature weight scores of the given dataset using a PCA-created component. The weight scores reflect the relevance of the features to their labels. The higher the weight scores, the more relevant it is considered to be.

In this research, feature selection involves (1) weighting each feature by the statistical methods (Information Gain, Correlation or PCA), and then (2) finding an optimal threshold of the weight score that enables selection of the best subset.

Demographic Features Selection

In student performance prediction tasks, demographics are commonly used in prediction models of real projects. It is worth mentioning that some papers found that demographics as features have a lower impact than behavioural features on predictive analysis of some cases. For instance, a study by Behr et al. (2020) points out that demographics do not significantly improve the prediction models' accuracy. Another example, the study of Tomasevic et al. (2020), adopts the same data source as this research (OULAD datasets) and found that demographic features do not significantly influence the prediction model performance. Therefore, this section intends to examine the significance of the demographic features of OULAD datasets and grounds the foundation of the modelling experiment – using only click behavioural features (without demographic features) to build prediction models.

The method is to build a series of models using both demographic and click behaviour features as input features. The process aligns with steps 3 to 5 of SPPA - feature selection, predictive modelling, and model evaluation. First, the feature selection method includes three filter methods: Information Gain, Correlation, and PCA. Second, modelling involves some easy and

straightforward ML algorithms - LR, RI (Rule Induction) and k-NN with four datasets T-WEE, T-MON, TA-WEE, TA-MON plus demographic features. Accuracy is the measure of model evaluation. Finally, the importance of demographic and behavioural features is assessed by observing filtered important features in the best model in each dataset-algorithm combination.

Due to the demographic features involving categorical data types, further data processing is conducted before modelling. Specifically, six demographic features from *studentInfo* dataset are processed with the method of *one-hot encoding* to ensure the consistency of data types between demographic features and behavioural features. As a result, a total of 35 demographic features are generated (Table 3. 14).

Table 3. 14 Demographic features processed with the *one-hot encoding* method

FEATURE CODE	FEATURE NAME	DATA TYPE
F1	age_band = 0-35	Numerical
F2	age_band = 35-55	Numerical
F3	age_band = 55<=	Numerical
F4	disability	Numerical
F5	gender	Numerical
F6	highest_education = A Level or Equivalent	Numerical
F7	highest_education = HE Qualification	Numerical
F8	highest_education = Lower Than A Level	Numerical
F9	highest_education = No Formal quals	Numerical
F10	highest_education = Post Graduate Qualification	Numerical
F11	imd_band = 0-10%	Numerical
F12	imd_band = 10-20	Numerical
F13	imd_band = 20-30%	Numerical
F14	imd_band = 30-40%	Numerical
F15	imd_band = 40-50%	Numerical
F16	imd_band = 50-60%	Numerical
F17	imd_band = 60-70%	Numerical
F18	imd_band = 70-80%	Numerical
F19	imd_band = 80-90%	Numerical
F20	imd_band = 90-100%	Numerical
F21	num_of_prev_attempts	Numerical
F22	region = East Anglian Region	Numerical
F23	region = East Midlands Region	Numerical
F24	region = Ireland	Numerical
F25	region = London Region	Numerical
F26	region = North Region	Numerical
F27	region = North Western Region	Numerical
F28	region = Scotland	Numerical
F29	region = South East Region	Numerical
F30	region = South Region	Numerical
F31	region = South West Region	Numerical
F32	region = Wales	Numerical
F33	region = West Midlands Region	Numerical
F34	region = Yorkshire Region	Numerical
F35	studied_credits	Numerical

These demographic features plus all the behavioural features in the four datasets T-WEE, T-MON, TA-WEE, and TA-MON, are fitted into models. The input data are scaled (0, 1), then split by 90% train and 10% test sets while modelling. All the weight scores are normalised to be (0, 1). To find the optimal subset using filter methods, the best weight score threshold is found by 21 iterations, indicating 21 weight score threshold numbers (Figure 3. 27). Moreover, LR and RI use the default parameters. As for k-NN models, the k value (k=7) is determined by trying a range of k values (16 values from 1 to 31) using the grid search.

Optimize Parameters (Grid) by Information Gain (4) (21 rows, 3 columns)

iteration	Select by Weights (14).weight ↑	accuracy	iteration	Select by Weights (14).weight ↑	accuracy
1	0	0.792	11	0.500	0.858
2	0.050	0.815	12	0.550	0.866
3	0.100	0.826	13	0.600	0.865
4	0.150	0.827	14	0.650	0.865
5	0.200	0.846	15	0.700	0.865
6	0.250	0.858	16	0.750	0.865
7	0.300	0.851	17	0.800	0.865
8	0.350	0.838	18	0.850	0.880
9	0.400	0.837	19	0.900	0.880
10	0.450	0.835	20	0.950	0.873
			21	1	0.873

Figure 3. 27 Using grid search to find the best accuracy through 21 iterations (an example)

There are 36 models built, and the results are shown in Table 3. 15. For each dataset and each algorithm, the 12 best models are highlighted in red, and these models do not involve any demographics. In other words, the result shows that all demographic features were filtered out in all the best models, meaning the demographic features are less important to achieving optimal performance than the click behaviour features in this research dataset. This result is consistent with the findings of Tomasevic, Gvozdenovic and Vranes (2020)'s study. That implies that it is possible to achieve the best model using only click behaviour features. This result grounds the basis of the experimental method by using only click behaviour data to train the models.

Table 3. 15 The result of experimental preparation modelling

Algorithm	Dataset	Feature selection methods	The best accuracy (among 21 iterations)	Weight score threshold	Whether demographics involved	Demographic Features involved
LR	D & T-WEE	InfoGain	86.89%	0.8	No	F8
		Corr	86.14%	0.7	No	
		PCA	84.27%	0.3	No	
	D & T-MON	InfoGain	87.45%	0.85	No	
		Corr	86.52%	0.45	Yes	
		PCA	86.52%	0.1	No	
	D & TA-WEE	InfoGain	87.27%	0.6	No	
		Corr	85.96%	0.7	No	
		PCA	74.53%	0.1	No	
	D & TA-MON	InfoGain	87.64%	0.85	No	
		Corr	87.08%	0.55	No	
		PCA	80.71%	0.1	No	
RI	D & T-WEE	InfoGain	86.89%	0.2	No	F1-35
		Corr	86.33%	0	Yes	
		PCA	85.39%	0.4	No	
	D & T-MON	InfoGain	87.45%	0.5	No	
		Corr	88.01%	0.6	No	
		PCA	87.08%	0.2	No	
	D & TA-WEE	InfoGain	87.27%	0.4	No	
		Corr	87.45%	0.8	No	
		PCA	85.96%	0.1	No	
	D & TA-MON	InfoGain	87.83%	0.4	No	
		Corr	87.64%	0.1	Yes	F1-2, F4, F6-13, F16, F19, F20-21, F25, F27, F29-0, F32, F35
		PCA	87.27%	0	Yes	F1-35
k-NN	D & T-WEE	InfoGain	86.89%	0.8	No	F1-35
		Corr	84.83%	0.9	No	
		PCA	79.40%	0	Yes	
	D & T-MON	InfoGain	87.45%	0.85	No	
		Corr	86.33%	1	No	
		PCA	77.34%	0	Yes	
	D & TA-WEE	InfoGain	86.89%	0.8	No	
		Corr	85.96%	0.9	No	
		PCA	79.59%	0	Yes	
	D & TA-MON	InfoGain	87.64%	0.85	No	
		Corr	86.52%	0.95	No	
		PCA	79.40%	0	Yes	

n.b.

D & T-WEE: the demographic dataset and the dataset T-WEE

D & T-MON: the demographic dataset and the dataset T-MON

D & TA-WEE: the demographic dataset and the dataset TA-WEE

D & TA-MON: demographic dataset and the dataset TA-MON

InfoGain: filter method Information Gain

Corr: filter method Correlation

PCA: filter method PCA

3.4.5 Machine Learning Algorithms

The analysis of a range of techniques in the literature review demonstrated that each classifier has its characteristics and specialities. For traditional machine learning, this research selects LR, k-NN, RF, and GBT. Also, this research intends to take advantage of deep learning's capabilities of extracting prominent features in neural network mechanisms and learning features from temporal data. Therefore, two deep learning algorithms 1D-CNN and LSTM are selected.

Traditional Machine Learning

First, this study selected **LR (Logistic Regression)** as the baseline algorithm because of its effectiveness in previous research (Aljohani, Fayoumi & Hassan 2019). The equation of LR is as follow:

$$Y = \frac{1}{1 + e^{-\theta^T \cdot X + b}}$$

In the equation, X is the click behavioural input features, and θ is the parameter the model aims to learn. When features are fed into the logistic function, the inputs are transferred to between 0 and 1 as the output, indicating the probability for Y (a given classification of fail or pass).

Second, **k-NN** is selected due to its simplicity. In the problem being addressed in this research, k-NN considers Y similar when students' click behaviour input features X are similar. Specifically, Y is computed by averaging k closest neighbour values. It is worth noting that k-NN is not suitable for extensive dimensional data such as the dataset TA-WEE involving 480 features. Therefore, this research expects significant impacts of k-NN when using feature selection (the feature numbers could be significantly reduced resulting in a low dimensional data).

Third, two ensembles **RF and GBT** are selected. The choice of these two algorithms is motivated by their boosted model performance when dealing with clickstream data based on the observations from literature review.

The four traditional machine learning algorithms selected are expected to perform well while using the datasets of strategies 1 and 2 but might not be suitable for Strategy 3. This is because traditional machine learning methods have a low capacity to learn features from sparse data, such as the panel data of Strategy 3. This hypothesis needs to be proven through the experiment of the research.

Deep Learning - 1D-CNN

This research proposes that the process of CNN dealing with images can be adjusted to be able to deal with data sequences. Specifically, a **1D-CNN** (One-dimensional convolutional neural network) model for student performance prediction is used. The architecture of the model is illustrated in Figure 3. 28. Apart from input and output layers, the model architecture includes one *1D Convolutional Layer*, followed by one *Pooling Layer* and one *Fully Connected Layer* as hidden layers.

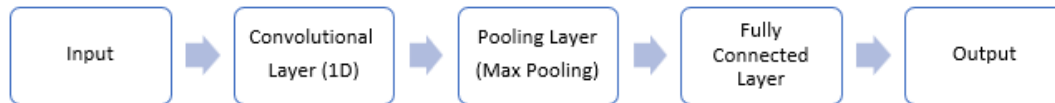


Figure 3. 28 The architecture of 1D-CNN model

The key to this model is the way the 1D convolutional layer works, which is demonstrated in Figure 3. 29. Like 2D-CNN dealing with images, extracting features from the input matrix, the 1D Convolutional Layer is devoted to extracting features from the input vector. It uses kernel and stride to control the feature extracting window (kernel size = the size of the window) and the window slice step (e.g., stride size = 2, meaning that it is moving two steps for each time slice).

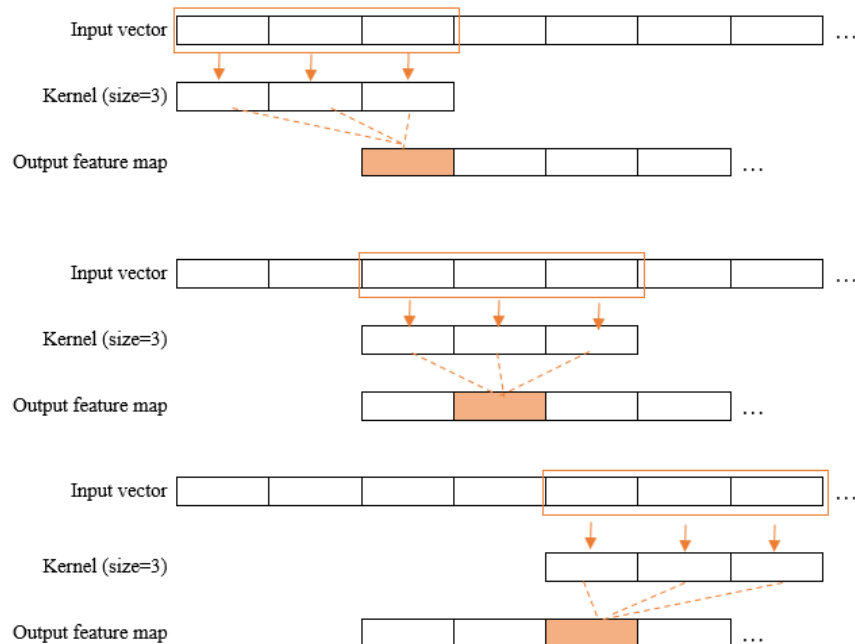


Figure 3. 29 1D Convolutional work process (kernel size = 3, stride = 2)

When the input vector involves temporal data, the moving windows can be thought of as extracting features over time, then projecting them onto the feature map. After this process, these features are sent to the *Pooling Layer* to be summarised. Then, the maximum element is selected as prominent features from the feature map by the *Max Pooling* layer. Finally, these prominent features are sent to the *Fully Connected Layer (dense)*, generating the output Y.

The constructed 1D-CNN model is expected to perform well using feature-based data sequences such as datasets of strategies 1 and 2. However, theoretically, this model cannot deal with sample-based or observation-based data sequences such as Strategy 3 datasets.

Deep Learning - LSTM

Next, LSTM is selected due to its well-known capacity to deal with temporal data and its outstanding performance in tabular clickstream data. This research uses a stacked LSTM architecture for student performance prediction (Figure 3. 30).

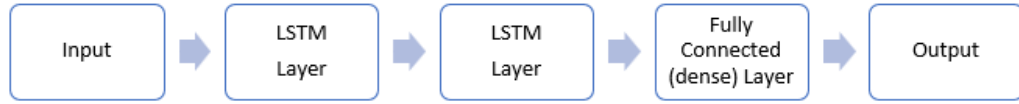


Figure 3. 30 The stacked LSTM architecture

The model adopts two LSTM hidden layers for feature extraction. Then a *Fully Connected (dense) Layer* is used to interpret the features extracted before an output layer makes predictions. Also, compared with 1D-CNN dealing with temporal data from features, LSTM is expected to learn features from panel data with temporal-involved observations of each panel member (i.e., each student).

3.4.6 Model Evaluation Methods

Three prediction performance measures are used in this research: accuracy, F1-score and AUC (the area-under-the-ROC-curve). To clarify the calculation methods, the following demonstrates the basic elements for calculating these three measures:

- TP (True Positive): the number of students whose final result was ‘pass’ in reality and were predicted as ‘pass’ by the model, which is correct positive prediction.
- TN (True Negative): the number of students whose final result was ‘fail’ in reality and were predicted as ‘fail’ by the model, which is correct negative prediction.

- FP (False Positive): the number of students whose final result is 'fail' in reality and were predicted as 'pass' by the model, which is incorrect positive prediction.
- FN (False Negative): the number of students whose final result is 'pass' in reality and were predicted as 'fail' by the model, which is incorrect negative prediction.
- TN + FP: the number of students whose final result is 'fail' in reality.
- TP + FN: the number of students whose final result is 'pass' in reality.
- TP + FP: the number of students whose final result is 'pass' in prediction.
- TN + FN: the number of students whose final result is 'fail' in prediction.
- TOTAL: total number of students.

Accuracy is one of the most common measures in data science tasks. Accuracy is calculated as demonstrated below:

$$Accuracy = \frac{TP + TN}{TOTAL}$$

F1-score is a comprehensive measure of Precision and Recall. The formula is provided below:

$$F1-score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

In the formula, Precision is the conditional probability that a passed student in reality is correctly classified by the model. A greater Precision indicates a better prediction performance for the positive class. Its calculation is demonstrated below:

$$Precision = \frac{TP}{TP + FP}$$

Recall is the probability of a passed student being classified correctly, which is calculated as:

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

Precision implies how accurately the classifier predicted the positive cases, while Recall refers to how completely the classifier predicted positive cases. Precision would be adopted when an investigation focuses on minimising FP, while Recall is used for an investigation that aims to minimise FN. However, an increase in Recall often leads to a decrease in Precision. As this

research focuses on measuring the model's ability in general (rather than minimising FN or FP), F1-score is appropriate.

AUC score is a single measure representing the predictive ability of models. AUC is employed because this measure is not sensitive to imbalanced data and the given classifying threshold. Theoretically, the AUC score is between 0 and 1. In practice, the AUC scores being greater than 0.5 means that the prediction is better than a random guess.

3.4.7 Experimental Design

The experimental design aims to integrate all aspects into the experimental methods, including the six datasets developed through feature engineering strategies, the chosen feature selection methods, and the selected traditional machine learning algorithms and constructed deep learning models. Experimentation involves a predictive modelling process using six datasets, one feature selection method and six algorithms with other aspects as follows:

- Dataset: T-WEE, T-MON, TA-WEE, TA-MON, P-WEE and P-MON;
- Feature selection: the filter method using Information Gain. The results of the demographics selection from section 3.4.4 of this chapter show that of the three feature selection methods, Information Gain shows the highest accuracy in most cases. Therefore, this method is used in modelling;
- Algorithms: four traditional machine learning - LR, k-NN, RF, GBT, and two deep learning - 1D-CNN and LSTM;

Table 3. 16 Building 60 models in the experiment design

Dataset	Machine Learning				Deep Learning	
	LR	k-NN	RF	GBT	1D-CNN	LSTM
T-WEE	2 models	2 models	2 models	2 models	1 model	1 model
T-MON	2 models	2 models	2 models	2 models	1 model	1 model
TA-WEE	2 models	2 models	2 models	2 models	1 model	1 model
TA-MON	2 models	2 models	2 models	2 models	1 model	1 model
P-WEE	2 models	2 models	2 models	2 models	1 model	1 model
P-MON	2 models	2 models	2 models	2 models	1 model	1 model

- Models: 60 models are built for this research, containing 48 traditional machine learning models and 12 deep learning models (Table 3. 16). Specifically, two models are built for

each algorithm LR, k-NN, RF, GBT with the condition of using and not using the feature selection method. For 1D-CNN and LSTM, only one model is built for each, which does not use the feature selection method. This is because 1D-CNN and LSTM have the capacity to weight features within the neural network mechanism;

- Modelling processes follow *step 3* to *step 5* of SPPA: feature selection, predictive modelling, model evaluation;
- Model evaluation: Accuracy, F1-score, AUC;
- Data validation: the experiment adopts 10-fold cross-validation to provide solid results as well as maximise the data proportion used for the training set (90% train set, 10% test set). Also, for the panel data (P-WEE and P-MON), the temporal nature is within each panel member, that is, within each individual student's data. Individual student data are independent. For this case, taking advantage of the independence of individual students to implement the 10-fold cross-validation is adopted in this research. That means that each individual student (40 rows in P-WEE and 10 rows in P-MON) is seen as a group, and the rows within the group are fixed. Only the whole group is randomly partitioned for each fold iteration (rather than the original rows);
- Experiment is implemented using RapidMiner and Python.

3.4.8 Feature Importance Analysis

Important features can be identified from the feature selection process and result for traditional machine learning. Also, important features are manually examined for deep learning models. The manual method is related to examining how the model performance changes when discarding those features that are judged as important.

Chapter 4

IMPLEMENTATION AND RESULTS

This chapter demonstrates the experiment implementation and analysis result in line with *step 3* to *step 5* of SPPA (as discussed in section 3.3.3) - feature selection, predictive modelling, and model evaluation. Firstly, for traditional machine learning, the implementation of modelling with and without feature selection is demonstrated in section 4.1. Secondly, for deep learning, model implementation is described in section 4.2. Next, model evaluation is demonstrated in section 4.3, followed by section 4.4, which focuses on feature importance analysis. In this chapter, some abbreviations regarding the datasets are used: T- refers to the datasets T-WEE and T-MON, TA- refers to the datasets TA-WEE and TA-MON, and P- refers to the datasets P-WEE and P-MON.

4.1 Building Models by Traditional Machine Learning

In this section, 48 models are built by traditional machine learning algorithms (LR, k-NN, RF and GBT) using six datasets (T-, TA- and P-). Each algorithm-dataset combination has two models to build. One model is built without feature selection methods (M1), and another is built with a feature selection method (M2). Section 4.1.1 discusses the implementation of 24 M1 models, and section 4.1.2 details 24 M2 models' implementation.

4.1.1 Modelling without Feature Selection

Overall, 24 M1 models use all the features of each dataset, along with parameter tuning for building optimal models. All implementations use 1992 as the random seed in RapidMiner. Traditional machine learning algorithms LR, k-NN, RF and GBT are involved. Their modelling details are as follow:

- Six LR M1 models adopt normalised input data (0, 1). These models are built using the default parameters of the Logistic Regression operator in RapidMiner.

- Six k-NN M1 models adopt normalised input data (0, 1) as well. For each M1 model, the k value is determined by attempting 16 possibilities in the range 1 to 31 using the Grid Search operator of RapidMiner. As a result, for T- and TA- datasets, the optimal k value is 11; for P- datasets, the optimal k value is 7. The parameter of distance measure is *MixedMeasures* for all models.
- Six RF M1 Models adopt original input data (i.e., without normalisation). Each M1 model involves tuning two parameters – *number_of_trees* and *maximal_depth*. The grid search is adopted by trying 11 possibilities (1-500, steps=10, scale is linear) for the parameter *number_of_trees* and three possibilities (1, 2, 3) for the parameter *maximal_depth*. Therefore, a total of 33 combination possibilities are attempted for each model. As a result, the models with datasets T- and TA-, *number_of_trees* = 151 and *maximal_depth* = 3. For the models with the datasets P-, *number_of_trees* = 350, *maximal_depth* = 3.
- Six GBT M1 models adopt original input data (i.e., without normalisation). Each M1 model involves tuning three parameters by the Grid Search operator of RapidMiner. They are *number_of_trees*, *maximal_depth* and *learning_rate*. There are 11 possibilities (from 1 to 200, scale is linear, steps = 10) attempted for the parameter *number_of_trees*. Three possibilities (1, 2, 3) are attempted for the parameter *maximal_depth*. Five possibilities (1, 0.1, 0.01, 0.001, 0.0001) are attempted for the parameter *learning_rate*. Altogether, 165 combination possibilities are attempted to find the best combinative parameter sets. As a result, all models adopt *number_of_trees* = 61, *maximal_depth* = 3 and *learning_rate* = 0.1.

4.1.2 Modelling with Feature Selection

Building process of 24 M2 models in RapidMiner is demonstrated in this section. This process involves (1) weighting features by Information Gain and acquiring weight scores of features and (2) finding the best threshold of weight scores that enables a selection of the best subset (features). Correspondingly, the implementation involves two parts in RapidMiner (Figure 4. 1). Part one is obtaining all click behaviour features' weight scores by the operator Weight by Information Gain. Part two is to build models with a range of possible weight scores threshold, conducted iteratively. A single iteration process involves using the operator Select by Weights to set up a threshold weight score (e.g., 0.7) and then building models with a traditional machine learning algorithm

using that threshold. Multiple iterations of this process enabled the best threshold weight score to be found, leading to the best subsets enabling the best model performance. Multiple iterations (11) are conducted in this process, using 11 possible threshold weight scores from 0, 0.1, 0.2, ..., 0.9, up to 1. Each-time iteration is a process of modelling, so tuning the parameters in the modelling is included.

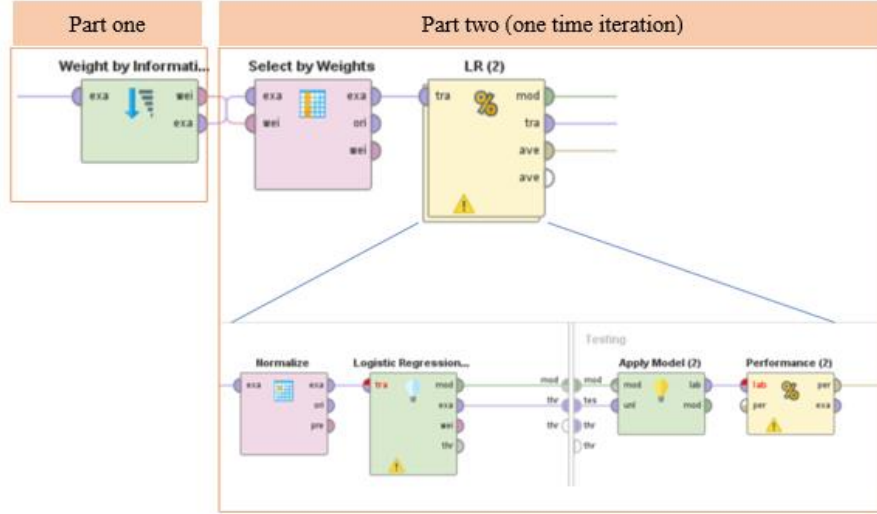


Figure 4. 1 M2 model implementation process in RapidMiner

The output of part one of the implementation is lists of weight scores of all features of the six datasets (see Table 4. 1 to Table 4. 6). All weight scores are normalised (0, 1), indicating the relevance of a feature to the label. As there are too many features in TA-WEE (480 features) and TA-MON (120 features), Table 4. 3 and Table 4. 4 only display a part of the results; all results can be found in Appendix 1 and Appendix 2. These results were used to calculate the number of the features in part two, as well as to identify the important features in section 4.4 feature importance analysis.

Table 4. 1 Weight scores of feature weighting by Information Gain in T-WEE

<i>Feature</i>	<i>Weight</i>	<i>Feature</i>	<i>Weight</i>	<i>Feature</i>	<i>Weight</i>	<i>Feature</i>	<i>Weight</i>
<i>T29</i>	1.00	<i>T27</i>	0.75	<i>T13</i>	0.51	<i>T5</i>	0.21
<i>T30</i>	0.88	<i>T26</i>	0.74	<i>T14</i>	0.45	<i>T36</i>	0.20
<i>T28</i>	0.86	<i>T21</i>	0.73	<i>T35</i>	0.43	<i>T37</i>	0.13
<i>T32</i>	0.85	<i>T25</i>	0.69	<i>T8</i>	0.35	<i>T4</i>	0.12
<i>T22</i>	0.85	<i>T19</i>	0.65	<i>T9</i>	0.29	<i>T2</i>	0.12
<i>T34</i>	0.84	<i>T20</i>	0.65	<i>T10</i>	0.28	<i>T38</i>	0.09
<i>T23</i>	0.82	<i>T18</i>	0.62	<i>T6</i>	0.27	<i>T0</i>	0.08
<i>T31</i>	0.80	<i>T17</i>	0.59	<i>T7</i>	0.25	<i>T1</i>	0.07
<i>T33</i>	0.79	<i>T16</i>	0.55	<i>T12</i>	0.23	<i>T3</i>	0.05
<i>T24</i>	0.77	<i>T15</i>	0.53	<i>T11</i>	0.21	<i>T39</i>	0.00

Table 4. 2 Weight scores of feature weighting by Information Gain in T-MON

<i>Feature</i>	Weight	<i>Feature</i>	Weight
<i>T7</i>	1.00	<i>T3</i>	0.28
<i>T8</i>	0.92	<i>T2</i>	0.20
<i>T6</i>	0.81	<i>T9</i>	0.18
<i>T5</i>	0.64	<i>T1</i>	0.05
<i>T4</i>	0.50	<i>T0</i>	0.00

Table 4. 3 Part of the weight scores of feature weighting by Information Gain in TA-WEE

<i>Feature</i>	Weight	<i>Feature</i>	Weight	<i>Feature</i>	Weight	<i>Feature</i>	Weight
<i>T29_Act4</i>	1	<i>T27_Act4</i>	0.79	<i>T15_Act4</i>	0.55	<i>T28_Act1</i>	0.43
<i>T30_Act4</i>	0.9	<i>T26_Act4</i>	0.75	<i>T13_Act4</i>	0.54	<i>T30_Act3</i>	0.43
<i>T28_Act4</i>	0.88	<i>T21_Act4</i>	0.75	<i>T34_Act1</i>	0.54	<i>T21_Act1</i>	0.41
<i>T32_Act4</i>	0.87	<i>T25_Act4</i>	0.71	<i>T33_Act1</i>	0.54	<i>T26_Act1</i>	0.4
<i>T22_Act4</i>	0.85	<i>T19_Act4</i>	0.68	<i>T29_Act1</i>	0.49	<i>T30_Act6</i>	0.39
<i>T34_Act4</i>	0.85	<i>T20_Act4</i>	0.67	<i>T14_Act4</i>	0.48	<i>T29_Act3</i>	0.39
<i>T23_Act4</i>	0.82	<i>T18_Act4</i>	0.63	<i>T31_Act1</i>	0.48	<i>T8_Act4</i>	0.39
<i>T31_Act4</i>	0.82	<i>T17_Act4</i>	0.61	<i>T30_Act1</i>	0.47	<i>T23_Act1</i>	0.38
<i>T24_Act4</i>	0.8	<i>T16_Act4</i>	0.59	<i>T22_Act1</i>	0.45	<i>T25_Act1</i>	0.38
<i>T33_Act4</i>	0.8	<i>T32_Act1</i>	0.57	<i>T35_Act4</i>	0.45	

Table 4. 4 Part of the weight scores of feature weighting by Information Gain in TA-MON

<i>Feature</i>	Weight	<i>Feature</i>	Weight	<i>Feature</i>	Weight	<i>Feature</i>	Weight
<i>T7_Act4</i>	1.00	<i>T7_Act1</i>	0.45	<i>T5_Act5</i>	0.28	<i>T6_Act7</i>	0.19
<i>T8_Act4</i>	0.93	<i>T6_Act3</i>	0.42	<i>T4_Act1</i>	0.27	<i>T3_Act1</i>	0.18
<i>T6_Act4</i>	0.82	<i>T7_Act5</i>	0.42	<i>T4_Act5</i>	0.27	<i>T9_Act1</i>	0.18
<i>T5_Act4</i>	0.65	<i>T6_Act2</i>	0.38	<i>T2_Act4</i>	0.25	<i>T8_Act2</i>	0.17
<i>T7_Act3</i>	0.63	<i>T6_Act1</i>	0.37	<i>T7_Act2</i>	0.25	<i>T2_Act1</i>	0.17
<i>T8_Act3</i>	0.58	<i>T5_Act1</i>	0.35	<i>T9_Act4</i>	0.24	<i>T2_Act5</i>	0.15
<i>T8_Act6</i>	0.58	<i>T3_Act4</i>	0.34	<i>T5_Act2</i>	0.23	<i>T5_Act7</i>	0.14
<i>T7_Act6</i>	0.57	<i>T6_Act5</i>	0.30	<i>T7_Act7</i>	0.21	<i>T3_Act5</i>	0.14
<i>T8_Act1</i>	0.56	<i>T5_Act3</i>	0.30	<i>T4_Act3</i>	0.21	<i>T8_Act7</i>	0.14
<i>T4_Act4</i>	0.54	<i>T6_Act6</i>	0.29	<i>T5_Act6</i>	0.21	

Table 4. 5 Weight scores of feature weighting by Information Gain in P-WEE

<i>Feature</i>	Weight	<i>Feature</i>	Weight	<i>Feature</i>	Weight
<i>Act4</i>	1	<i>Act7</i>	0.13	<i>Act10</i>	0.00
<i>Act1</i>	0.60	<i>Act5</i>	0.12	<i>Act12</i>	0.00
<i>Act2</i>	0.28	<i>Act8</i>	0.04	<i>T</i>	0.00
<i>Act3</i>	0.28	<i>Act9</i>	0.03		
<i>Act6</i>	0.22	<i>Act11</i>	0.03		

Table 4. 6 Weight scores of feature weighting by Information Gain in P-MON

<i>Feature</i>	Weight	<i>Feature</i>	Weight	<i>Feature</i>	Weight
<i>Act4</i>	1	<i>Act5</i>	0.32	<i>Act10</i>	0.01
<i>Act1</i>	0.63	<i>Act7</i>	0.19	<i>Act12</i>	0
<i>Act3</i>	0.54	<i>Act8</i>	0.09	<i>T</i>	0
<i>Act6</i>	0.43	<i>Act9</i>	0.06		
<i>Act2</i>	0.34	<i>Act11</i>	0.06		

The output of the part one implementation is related to the 24 M2 models. These models' parameters, iteration implementation, the best thresholds and the number of features involved in the models are listed in Table 4. 7. The number of features involved in each model refers to the total features whose weight score is greater than the best threshold. They are calculated based on the figures from Table 4. 1 - Table 4. 6. The iteration results regarding P-WEE and P-MON are not provided (listed as 'N/A' in Table 4. 7) because these models' performances are suboptimal (low AUC from 0.50 to 0.78, analysed in the later section 4.3 of this chapter). Therefore, their iteration results are meaningless for this research.

Table 4. 7 Outputs of building 24 M2 models

Algorithm	Dataset	Best threshold (# features involved in the model)	Iteration implementation figures	M2 models' parameters
LR	T-WEE	0.8 (8)	Figure 4. 2	All models' parameters are the same as LR M1 models
	T-MON	0.9 (2)	Figure 4. 2	
	TA-WEE	0.8 (10)	Figure 4. 2	
	TA-MO	0.9 (2)	Figure 4. 2	
	P-WEE	0.7 (1)	N/A	
	P-MON	0.7 (1)	N/A	
k-NN	T-WEE	0.8 (8)	Figure 4. 3	All models' parameters are the same as k-NN M1 models
	T-MON	0.9 (2)	Figure 4. 3	
	TA-WEE	0.8 (10)	Figure 4. 3	
	TA-MO	0.9 (2)	Figure 4. 3	
	P-WEE	0.7 (1)	N/A	
	P-MON	0.7 (1)	N/A	
RF	T-WEE	0.3 (24)	Figure 4. 4	T-WEE, T-MON, TA-WEE and TA-MON: <i>number_of_trees</i> = 251, <i>maximal_depth</i> = 3 P-WEE and P-MON: <i>number_of_trees</i> = 350, <i>maximal_depth</i> = 3.
	T-MON	0.8 (3)	Figure 4. 4	
	TA-WEE	0.6 (18)	Figure 4. 4	
	TA-MO	0.1 (46)	Figure 4. 4	
	P-WEE	0.3 (2)	N/A	
	P-MON	0.3 (4)	N/A	
GBT	T-WEE	0.0 (40)	Figure 4. 5	T-WEE, T-MON, TA-WEE, TA-MON: <i>number_of_trees</i> = 140, <i>maximal_depth</i> = 2, <i>learning_rate</i> = 0.1 P-WEE, P-MON: <i>number_of_trees</i> = 61, <i>maximal_depth</i> = 3, <i>learning_rate</i> = 0.1
	T-MON	0.0 (10)	Figure 4. 5	
	TA-WEE	0.0 (480)	Figure 4. 5	
	TA-MON	0.0 (120)	Figure 4. 5	
	P-WEE	0.3 (2)	N/A	
	P-MON	0.3 (4)	N/A	

iteration	Threshold.weight	accuracy ↓	iteration	Threshold.weight	accuracy ↓	iteration	Threshold.weight	accuracy ↓	iteration	Threshold.weight	accuracy ↓
9	0.800	0.866	10	0.900	0.881	9	0.800	0.865	10	0.900	0.880
4	0.300	0.862	11	1	0.875	7	0.600	0.859	11	1	0.873
3	0.200	0.857	1	0	0.871	8	0.700	0.856	8	0.700	0.861
8	0.700	0.856	8	0.700	0.860	6	0.500	0.855	9	0.800	0.861
7	0.600	0.854	9	0.800	0.860	4	0.300	0.847	3	0.200	0.851
5	0.400	0.853	3	0.200	0.860	5	0.400	0.839	6	0.500	0.849
1	0	0.850	2	0.100	0.859	10	0.900	0.822	4	0.300	0.848
6	0.500	0.849	5	0.400	0.847	3	0.200	0.819	7	0.600	0.843
2	0.100	0.834	4	0.300	0.847	2	0.100	0.792	5	0.400	0.832
11	1	0.702	7	0.600	0.836	1	0	0.773	2	0.100	0.831
10	0.900	0.702	6	0.500	0.836	11	1	0.702	1	0	0.817

LR & T-WEE

LR & T-MON

LR & TA-WEE

LR & TA-MON

Figure 4. 2 Results of the 11 iterations - threshold weight scores in LR models

iteration	Threshold.weight	accuracy ↓	iteration	Threshold.weight	accuracy ↓	iteration	Threshold.weight	accuracy ↓	iteration	Threshold.weight	accuracy ↓
9	0.800	0.866	10	0.900	0.881	9	0.800	0.865	10	0.900	0.880
8	0.700	0.853	11	1	0.875	8	0.700	0.854	11	1	0.873
7	0.600	0.834	8	0.700	0.860	7	0.600	0.829	9	0.800	0.861
6	0.500	0.806	9	0.800	0.860	10	0.900	0.822	8	0.700	0.861
5	0.400	0.800	7	0.600	0.834	6	0.500	0.808	7	0.600	0.835
4	0.300	0.780	6	0.500	0.834	5	0.400	0.802	5	0.400	0.807
3	0.200	0.744	4	0.300	0.806	3	0.200	0.762	6	0.500	0.807
11	1	0.725	5	0.400	0.806	4	0.300	0.762	4	0.300	0.787
10	0.900	0.725	3	0.200	0.744	11	1	0.735	3	0.200	0.751
2	0.100	0.718	2	0.100	0.744	2	0.100	0.717	2	0.100	0.710
1	0	0.704	1	0	0.703	1	0	0.714	1	0	0.706

k-NN & T-WEE

k-NN & T-MON

k-NN & TA-WEE

k-NN & TA-MON

Figure 4. 3 Results of the 11 iterations - threshold weight scores in k-NN models

iteration	Threshold.weight	accuracy ↓	iteration	Threshold.weight	accuracy ↓	iteration	Threshold.weight	accuracy ↓	iteration	Threshold.weight	accuracy ↓
4	0.300	0.884	9	0.800	0.885	7	0.600	0.881	2	0.100	0.887
7	0.600	0.883	8	0.700	0.885	4	0.300	0.878	5	0.400	0.886
8	0.700	0.883	5	0.400	0.885	5	0.400	0.877	3	0.200	0.886
2	0.100	0.883	4	0.300	0.885	8	0.700	0.877	6	0.500	0.886
1	0	0.883	3	0.200	0.884	6	0.500	0.876	4	0.300	0.886
3	0.200	0.882	2	0.100	0.884	3	0.200	0.871	1	0	0.877
5	0.400	0.881	6	0.500	0.884	9	0.800	0.870	7	0.600	0.876
6	0.500	0.876	7	0.600	0.884	2	0.100	0.853	10	0.900	0.875
9	0.800	0.873	1	0	0.884	10	0.900	0.822	8	0.700	0.875
11	1	0.741	10	0.900	0.883	11	1	0.735	9	0.800	0.875
10	0.900	0.741	11	1	0.875	1	0	0.702	11	1	0.873

RF & T-WEE

RF & T-MON

RF & TA-WEE

RF & TA-MON

Figure 4. 4 Results of the 11 iterations - threshold weight scores in RF models

iteration	Threshold.weight	accuracy ↓	iteration	Threshold.weight	accuracy ↓	iteration	Threshold.weight	accuracy ↓	iteration	Threshold.weight	accuracy ↓
1	0	0.886	1	0	0.889	1	0	0.887	1	0	0.890
2	0.100	0.886	6	0.500	0.887	4	0.300	0.884	2	0.100	0.888
8	0.700	0.882	7	0.600	0.887	3	0.200	0.884	3	0.200	0.888
4	0.300	0.882	9	0.800	0.886	2	0.100	0.882	5	0.400	0.887
5	0.400	0.882	8	0.700	0.886	5	0.400	0.880	4	0.300	0.886
3	0.200	0.882	3	0.200	0.885	6	0.500	0.880	6	0.500	0.886
7	0.600	0.882	2	0.100	0.884	8	0.700	0.879	7	0.600	0.882
6	0.500	0.879	4	0.300	0.883	9	0.800	0.878	9	0.800	0.881
9	0.800	0.877	5	0.400	0.883	7	0.600	0.877	8	0.700	0.881
11	1	0.722	10	0.900	0.883	10	0.900	0.822	10	0.900	0.880
10	0.900	0.722	11	1	0.869	11	1	0.723	11	1	0.869

GBT & T-WEE

GBT & T-MON

GBT & TA-WEE

GBT & TA-MON

Figure 4. 5 Results of the 11 iterations - threshold weight scores in GBT models

4.2 Building Models by Deep Learning

Twelve models were built, including six 1D-CNN models and six LSTM models. They are implemented using Python in Jupyter Notebook with 42 as a random seed. The input data of each dataset are normalised to (0, 1) before fitting into models. The following section describes the detailed architecture of 1D-CNN (section 4.2.1) and LSTM (section 4.2.2) models and their implementations.

4.2.1 1D-CNN Models

The detailed architecture of the 1D-CNN model used in this project is as follows. First, a *Sequential Keras* model is defined. Next is one 1D-CNN layer, followed by an *activation layer*, a *dropout layer* and a *pooling layer*. The *dropout layer* helps CNN avoid overfitting. The *pooling layer* consolidates the learned features to the essentials. After pooling, a *flatten layer* turns the learned features into a vector, and passes them to a *dense layer*. Finally, the learned features are dealt with in the output layer, making a prediction.

Some hyperparameters of each model are determined. First, the number of filters is tuned from 32, 64, 128 or 256. The kernel size is tuned from 2, 3, or 5. The two hyperparameters are 128 and 3 for the best results. Also, the *Adam* stochastic gradient descent is used to optimise NNs, and the loss function is *categorical binary cross-entropy*.

To acquire the best model, the tuned hyperparameters include the *batch size* (32-512), *dropout rate* (0.1-0.6), *learning rate* (0.1-0.00001) and *epochs* (50-1000). The *batch size* has little impact

on model performance in this case. The *dropout rate* slightly impacts the model's performance. However, the *learning rate* is the most significant one to tune because it significantly influences the loss of the models. Different *learning rates* were manually set up from 0.1 to 0.00001, the loss changes were observed for finding the best *learning rate*. *Epochs* is the second significant hyperparameter influencing the model performance. As a result, for each model, a *batch size* of 128, an *epochs* of 400, a *learning rate* of 0.001, and 0.4 as *dropout rate* shows the best prediction results. In building 1D-CNN models, input shapes for each dataset for modelling are different according to the structure of each dataset. Each model's input shape and its number of trainable parameters is shown in Table 4. 8.

Table 4. 8 1D-CNN trainable parameters for each dataset

Dataset	Input shape	Trainable params
T-WEE	(40, 1)	1665
T-MON	(10, 1)	769
TA-WEE	(480, 1)	15745
TA-MON	(120, 1)	4225
P-WEE	(13, 1)	897
P-MON	(13, 1)	897

4.2.2 LSTM Models

The detailed architecture of the implemented LSTM model is as follows. First, a *Sequential Keras* model is defined. Then, two LSTM hidden layers are added with the dropout functions. To stack two LSTM layers, the prior LSTM layer's output is expected as the input for the second LSTM layer. Therefore, the *return_sequence* argument on the first LSTM layer is set to *True* (defaults to *False*). The *dropout* function is added to decrease overfitting. Finally, a *dense layer* is added, and then a final output layer is used to make predictions. The optimisation function is *Adam* stochastic gradient descent, the loss function is the *categorical binary cross-entropy*.

The tuned hyperparameters in modelling include the number of hidden units of LSTM layers, *batch size*, *learning rate*, *epochs*, *dropout rate*. First, the *learning rate* is the most sensitive hyperparameter. After trying all the possibilities (0.1, 0.01, 0.001, 0.0001, 0.00001), all LSTM model's *learning rate* are set to the best rate, which is 0.0001. Also, the number of hidden units significantly impacts the model capability. Considering the small size of the dataset, when tuning the hyperparameter, a small number for hidden units is used to start (start from 8, then 16, 32, up

to 64). This is because more hidden units need more samples to train. It was discovered that the set-up of hidden units for each dataset differed. *Epochs* are slightly different as well (selecting from 50, 100, 200, until 1000). Finally, this model's *dropout rate* and *batch size* are not sensitive hyperparameters. A number of 0.2 as the *dropout rate* (selecting from 0.1, 0.2, until 0.6) and 128 as the batch size (selecting from 32, 64, 128, 256, 512) are used for all models. Each dataset's input shape, hidden units of the two LSTM layers and each model's trainable parameter numbers are also shown in Table 4. 9.

Table 4. 9 LSTM hyperparameters and trainable parameters for each dataset

Dataset	Input shape	The lstm1 hidden units	The lstm2 hidden units	Epochs	Trainable params
T-WEE	(1, 40)	32	16	500	12497
T-MON	(1, 10)	32	16	500	8657
TA-WEE	(1, 480)	32	16	500	68817
TA-MON	(1, 120)	32	16	500	22737
P-WEE	(40, 13)	32	8	700	7209
P-MON	(10, 13)	32	16	700	9041

4.3 Model Evaluation

In this section, 60 models' performances, including Accuracy, F1-score and AUC, are summarised (Table 4. 10). As 10-fold cross-validation is used in modelling, the variance is also demonstrated in the model performance (+/-). To analyse the result of the experiment, Table 4. 10 is divided into three areas based on AUC measures. They are well-performing models (highlighted with yellow), moderate-performed models (no highlighted colour), and suboptimal models (highlighted with grey colour).

Some models perform well. The RF, GBT, 1D-CNN, LSTM algorithms with T-WEE/T-MON/TA-WEE/TA-MON achieve high AUC results, ranging from 0.892 to 0.918. These models are highlighted in yellow in Table 4. 10. Within this area, the best model is LSTM & P-WEE (highlighted texts with green colour), showing an accuracy of 89.25% (+/- 0.97%), F1-score of 92.71% (+/- 0.62%) and AUC of 0.913 (+/- 0.014). Also, the model's variances are relatively low. The second best is GBT & TA-MON (highlighted texts with blue colour), showing an accuracy

Table 4. 10 All models' performance summaries

Dataset		LR Accuracy	k-NN Accuracy	RF Accuracy	GBT Accuracy	1D-CNN Accuracy	LSTM Accuracy
T-WEE	M1	84.97% (+/- 1.60%)	70.44% (+/- 0.20%)	88.41% (+/- 1.62%)	88.62% (+/- 1.35%)	87.61% (+/- 1.69%)	88.58% (+/- 1.86%)
	M2	86.59% (+/- 2.11%)	86.59% (+/- 2.11%)	88.45% (+/- 1.62%)	88.62% (+/- 1.35%)		
T-MON	M1	87.14% (+/- 1.88%)	70.27% (+/- 0.08%)	88.60% (+/- 1.85%)	88.88% (+/- 1.64%)	88.49% (+/- 1.60%)	88.07% (+/- 1.64%)
	M2	88.09% (+/- 2.17%)	88.09% (+/- 2.17%)	88.50% (+/- 1.74%)	88.88% (+/- 1.64%)		
TA-WEE	M1	77.34% (+/- 1.83%)	71.35% (+/- 0.48%)	70.25% (+/- 0.06%)	88.73% (+/- 1.54%)	85.79% (+/- 1.07%)	88.41% (+/- 1.12%)
	M2	86.52% (+/- 1.14%)	86.52% (+/- 1.14%)	88.09% (+/- 1.29%)	88.73% (+/- 1.54%)		
TA-MON	M1	81.71% (+/- 1.88%)	70.64% (+/- 0.22%)	87.36% (+/- 1.28%)	88.95% (+/- 1.48%)	88.19% (+/- 0.59%)	88.47% (+/- 0.66%)
	M2	87.98% (+/- 1.26%)	87.98% (+/- 1.26%)	88.67% (+/- 1.11%)	88.95% (+/- 1.48%)		
P-WEE	M1	70.24% (+/- 0.02%)	66.29% (+/- 0.30%)	70.25% (+/- 0.00%)	70.25% (+/- 0.00%)	70.25% (+/- 0.27%)	88.25% (+/- 0.07%)
	M2	70.25% (+/- 0.00%)	66.10% (+/- 0.32%)	70.25% (+/- 0.00%)	69.02% (+/- 1.99%)		
P-MON	M1	70.24% (+/- 0.03%)	76.46% (+/- 0.48%)	73.82% (+/- 3.83%)	76.47% (+/- 0.49%)	77.55% (+/- 0.88%)	88.67% (+/- 1.27%)
	M2	70.25% (+/- 0.00%)	76.46% (+/- 0.49%)	76.39% (+/- 0.47%)	76.47% (+/- 0.49%)		

Dataset		LR F-score	k-NN F-score	RF F-score	GBT F-score	1D-CNN F-score	LSTM F-score
T-WEE	M1	70.89% (+/- 3.83%)	1.25% *	77.27% (+/- 3.78%)	79.51% (+/- 2.84%)	91.34% (+/- 1.22%)	92.18% (+/- 1.30%)
	M2	71.26% (+/- 5.93%)	71.26% (+/- 5.93%)	77.65% (+/- 3.86%)	79.51% (+/- 2.84%)		
T-MON	M1	74.32% (+/- 4.32%)	0.13% *	77.74% (+/- 4.27%)	80.24% (+/- 3.17%)	92.15% (+/- 1.10%)	91.80% (+/- 1.16%)
	M2	75.89% (+/- 5.31%)	75.89% (+/- 5.31%)	77.63% (+/- 3.94%)	80.24% (+/- 3.17%)		
TA-WEE	M1	85.06% (+/- 1.21%)	83.06% (+/- 0.24%)	82.53% (+/- 0.04%)	92.32% (+/- 1.02%)	90.03% (+/- .77%)	92.05% (+/- 0.77%)
	M2	91.16% (+/- 0.69%)	91.16% (+/- 0.69%)	91.88% (+/- 0.86%)	92.32% (+/- 1.02%)		
TA-MON	M1	87.89% (+/- 1.27%)	82.72% (+/- 0.12%)	91.66% (+/- 0.77%)	92.41% (+/- 1.00%)	91.84% (+/- 0.46%)	92.11% (+/- 0.56%)
	M2	91.98% (+/- 0.81%)	91.98% (+/- 0.81%)	92.38% (+/- 0.72%)	92.41% (+/- 1.00%)		
P-WEE	M1	82.51% (+/- 0.01%)	72.94% (+/- 0.25%)	82.52% (+/- 0.00%)	82.52% (+/- 0.00%)	82.52% (+/- 0.18%)	92.71% (+/- 0.62%)
	M2	82.53% (+/- 0.00%)	72.67% (+/- 0.27%)	82.52% (+/- 0.00%)	79.58% (+/- 4.75%)		
P-MON	M1	82.51% (+/- 0.02%)	84.06% (+/- 0.37%)	83.88% (+/- 1.65%)	84.04% (+/- 0.38%)	85.24% (+/- 0.82%)	92.37% (+/- 0.81%)
	M2	82.53% (+/- 0.00%)	84.06% (+/- 0.38%)	83.96% (+/- 0.37%)	84.04% (+/- 0.38%)		

Dataset		LR AUC	k-NN AUC	RF AUC	GBT AUC	1D-CNN AUC	LSTM AUC
T-WEE	M1	0.855 (+/- 0.013)	0.707 (+/- 0.031)	0.909 (+/- 0.014)	0.918 (+/- 0.011)	0.910 (+/- 0.019)	0.906 (+/-0.019)
	M2	0.879 (+/- 0.016)	0.793 (+/- 0.035)	0.910 (+/- 0.013)	0.918 (+/- 0.011)		
T-MON	M1	0.842 (+/- 0.028)	0.634 (+/- 0.042)	0.905 (+/- 0.014)	0.911 (+/- 0.016)	0.897 (+/-0.019)	0.895 (+/-0.020)
	M2	0.852 (+/- 0.025)	0.810 (+/- 0.035)	0.901 (+/- 0.018)	0.911 (+/- 0.016)		
TA-WEE	M1	0.499 (+/- 0.054)	0.737 (+/- 0.030)	0.901 (+/- 0.015)	0.914 (+/- 0.014)	0.892 (+/-0.011)	0.904 (+/-0.007)
	M2	0.686 (+/- 0.040)	0.823 (+/- 0.019)	0.905 (+/- 0.013)	0.914 (+/- 0.014)		
TA-MON	M1	0.549 (+/- 0.037)	0.670 (+/- 0.024)	0.904 (+/- 0.021)	0.917 (+/- 0.014)	0.909 (+/-0.008)	0.903 (+/-0.008)
	M2	0.500 (+/- 0.000)	0.811 (+/- 0.024)	0.905 (+/- 0.020)	0.917 (+/- 0.014)		
P-WEE	M1	0.597 (+/- 0.007)	0.670 (+/- 0.006)	0.690 (+/- 0.004)	0.698 (+/- 0.004)	0.720 (+/-0.005)	0.913 (+/-0.014)
	M2	0.607 (+/- 0.006)	0.666 (+/- 0.007)	0.674 (+/- 0.004)	0.690 (+/- 0.004)		
P-MON	M1	0.482 (+/- 0.008)	0.671 (+/- 0.008)	0.751 (+/- 0.008)	0.763 (+/- 0.007)	0.786 (+/-0.006)	0.906 (+/-0.013)
	M2	0.500 (+/- 0.000)	0.670 (+/- 0.010)	0.734 (+/- 0.009)	0.763 (+/- 0.007)		

1.25% *: precision is 100.00% but recall is 0.63% (+/- 0.66%)

0.13% *: precision is 100.00% but recall is 0.06% (+/- 0.20%)

of 88.95% (+/- 1.48%), F1-score of 92.41% (+/- 1.00%) and AUC of 0.911 (+/- 0.016).

Some models, such as LR with T-WEE/T-MON and k-NN with T-WEE/T-MON/TA-WEE/TA-MON have moderate performances, shown without any highlight colour in Table 4. 10. Two models stand out in this area. The first is LR & T-MON & M2 (marked texts with underline) with 88.09% (+/- 2.17%) accuracy, 75.89% (+/- 5.31%) F1-score and 0.852 (+/- 0.025) AUC, which is the best LR model among six datasets. This model has a relatively large variance in accuracy and F1-score. The second model is k-NN & T-MON & M2 (marked texts with underline), as the best k-NN model among the six datasets, and it has a similar performance: 88.09% (+/- 2.17%) accuracy, 75.89% (+/- 5.31%) F1-score and 0.810 (+/- 0.035) AUC.

Some models do not perform well and are highlighted in the grey colour in Table 4. 10. The models k-NN, RF, GBT, 1D-CNN with P-WEE/P-MON show low AUC scores, between 0.50 to 0.78. This range of AUC means that the model's ability is a random guess or slightly better. Also, their corresponding accuracy ranges from 66% to 76% with F1-score is between 72% to 85%. Due to the imbalanced data (68% Pass, 32% Fail), this performance range is not ideal in practice. Other suboptimal models are LR with TA-WEE/TA-MON. Their AUC scores, from 0.499 to 0.686, are also equivalent to random prediction or slightly better, although some of their accuracy and F1-score results seem not too bad (accuracy in a range from 77.24 % to 87.98%; F1-score in a range from 85.06% to 91.98%).

In summary, as this research expected (as mentioned in machine learning algorithm selection in section 3.4.5), traditional machine learning and 1D-CNN are not suitable for learning features derived from panel data. LSTM showed advantages in dealing with panel data, as expected.

4.4 Feature Importance Analysis

According to the experimental results, the best model for this research uses the dataset P-WEE. This dataset structure uses activity categories as features with a weekly view. Therefore, the feature importance of this research aims to examine which weeks and activity categories are essential to prediction. To do this, the feature importance is analysed for the best models that use each dataset in feature engineering strategies in the weekly view. In other words, the three best models, using datasets T-WEE, TA-WEE and P-WEE, are examined. These models' performances are highlighted texts with red colour in Table 4.10.

Table 4. 11 shows the feature importance analysis goals and their analytical methods. For Strategies 1 and 2, due to the best model being related to GBT, this research uses feature selection iteration results that were generated in section 4.1.2 to identify important features. This is demonstrated in section 4.4.1 and 4.4.2. For Strategy 3, this research examines the dominant features of the LSTM model by discarding each feature of P-WEE to observe the model performance (accuracy) changes. This is demonstrated in section 4.4.3.

Table 4. 11 Feature importance analysis summary

Feature engineering strategy	Model	Feature Importance analysis goal	Analysis Methods
Strategy 1	GBT & T-WEE & M2	Which weeks are important	Observing the feature selection iteration
Strategy 2	GBT & TA-WEE & M2	Which combination of weeks and activities are important	Observing the feature selection iteration
Strategy 3	LSTM & P-WEE	Which activities are important	Observing model performance changes when discarding each feature

4.4.1 Important Features of Weekly-view Strategy 1

For Strategy 1, in the model GBT & T-WEE & M2, the best threshold weight score is 0 (indicating using all features); it is hard to identify the dominant feature. Therefore, this research seeks the second and the third-best accuracies in the iterations of the feature selection process (Figure 4. 6). The second-best is when the threshold is 0.1, and its features involve T2, T4-T37. The third-best is when the threshold is 0.7 and the features are T21-24, T26-34. From this point of view, T21-24 and T26-34 are the consistent features in the second and third-best accuracies. Therefore, these are the important features of this model. To sum up, among 40 weeks of the course, week 2, and week 4-week 37 are significant. In those weeks, weeks 21-24 and weeks 26-34 are the most significant.

iteration	Threshold.weight	accuracy ↓	Threshold	Accuracy	Input features
1	0	0.886	0	88.62%	all 40 features
2	0.100	0.886	0.1	88.56%	T2, T4-T37 (35 features)
8	0.700	0.882	0.7	88.22%	T21-24, T26-34 (13 features)

Figure 4. 6 Important features analysis in GBT & T-WEE & M2

4.4.2 Important Features of Weekly-view Strategy 2

For Strategy 2, the important features are examined in the model GBT & TA-WEE & M2. Figure 4. 7 shows the iteration results of the model. The best threshold weight score in the experiment is 0 for the best accuracy, 88.73%. Therefore, the second-best is examined: with a threshold weight score of 0.3, the accuracy achieves 88.43% with a total of 62 features (Figure 4. 8) There are 28 features involving *Act4*, while 21 features involve *Act1*, six features involve *Act6*, five features involve *Act3* and two features involve *Act2*. Therefore, from the perspective of activity, it can be concluded that *Act4* and *Act1* are the most important features in the model. From the combinative activity category and time perspectives, *Act4* in T6-T14, *Act1*, *Act4* in T15-T21, *Act1*, *Act2*, *Act4*

Iteration	Threshold.weight	accuracy ↓	Threshold	Accuracy	Input features
1	0	0.887	0	88.73%	all 480 features
4	0.300	0.884			
3	0.200	0.884	0.3	88.43%	see Figure 4. 8 (62 features)

Figure 4. 7 Important features analysis in GBT & TA-WEE & M2

in T22-T24, *Act1*, *Act3*, *Act4*, *Act6* in T28-T32 and *Act1*, *Act4* in T33-T35 are important. Finally, these important features are summarised in Table 4. 12.

T6_Act4	T15_Act1	T22_Act1	T25_Act1	T28_Act1	T33_Act1
T7_Act4	T15_Act4	T22_Act2	T25_Act4	T28_Act3	T33_Act4
T8_Act4	T16_Act1	T22_Act4	T26_Act1	T28_Act4	T34_Act1
T9_Act4	T16_Act4	T23_Act1	T26_Act4	T28_Act6	T34_Act4
T10_Act4	T17_Act1	T23_Act4	T27_Act1	T29_Act1	T34_Act6
T13_Act4	T17_Act4	T24_Act1	T27_Act4	T29_Act3	T35_Act1
T14_Act4	T18_Act1	T24_Act2		T29_Act4	T35_Act4
	T18_Act4	T24_Act4		T29_Act6	
	T19_Act1			T30_Act1	
	T19_Act4			T30_Act3	
	T20_Act1			T30_Act4	
	T20_Act4			T30_Act6	
	T21_Act1			T31_Act1	
	T21_Act4			T31_Act3	
				T31_Act4	
				T31_Act6	
				T32_Act1	
				T32_Act3	
				T32_Act4	
				T32_Act6	

Figure 4. 8 Features in the model GBT & TA-WEE & M2 (threshold = 0.3)

Table 4. 12 Important features in the model GBT & TA-WEE & M2

Feature name	Activity name	Time period (T=week)
Act4	homepage	T6-T14, T15-T21, T22-T24, T28-T32, T33-35
Act1	forumng	T15-T21, T22-T24, T28-T32, T33-35
Act3	subpage	T28-T32
Act6	resource	T28-T32

4.4.3 Important Features of Weekly-view Strategy 3

For Strategy 3, important features are examined from the model LSTM & P-WEE. The model using all 12 features achieved 89.25%. The result of removing each feature of the model is shown Table 4. 13. It is observed that without *Act4*, the accuracy reduced the most significantly (from 89.25% to 88.90%). *Act3* shows a similar phenomenon. Therefore, *Act4* and *Act3* are the most dominant features.

Table 4. 13 Feature Importance analysis of LSTM & P-WEE

Input features	Accuracy
All features	89.25% (+/- 0.97%)
without <i>Act4</i> (<i>homepage</i>)	88.90% (+/- 1.27%)
without <i>Act3</i> (<i>subpage</i>)	88.90% (+/- 0.78%)
without <i>Act2</i> (<i>oucontent</i>)	89.03% (+/- 0.68%)
without <i>Act5</i> (<i>quiz</i>)	89.07% (+/- 0.93%)
without <i>Act7</i> (<i>url</i>)	89.12% (+/- 0.89%)
without <i>Act6</i> (<i>resource</i>)	89.18% (+/- 0.87%)
without <i>Act8</i> (<i>oucollaborate</i>)	89.16% (+/- 0.97%)
without <i>Act10</i> (<i>ouelluminate</i>)	89.16% (+/- 0.83%)
without <i>Act1</i> (<i>forumng</i>)	89.22% (+/- 0.79%)
without <i>Act11</i> (<i>glossary</i>)	89.22% (+/- 0.77%)
without <i>Act12</i> (<i>sharedsubpage</i>)	89.22% (+/- 0.69%)
without <i>Act9</i> (<i>questionnaire</i>)	89.23% (+/- 0.89%)

Chapter 5

DISCUSSION AND CONCLUSION

This chapter discusses the implementational and analytical results from the previous section, along with the two research questions. The discussion of RQ1 and RQ2 are represented in 5.1 and 5.2, respectively. The research limitation and future research recommendations are provided in section 5.3. Section 5.4 presents the conclusion of the thesis.

5.1 Discussion of RQ1

This section intends to answer the first research question: what feature engineering strategies can be used in student performance prediction using clickstream data? An earlier section (3.4.3) explored feature engineering strategies to generate click behaviour features to predict student performance. In this research, the click data are aggregated based on weekly and monthly click count. Feature engineering strategies utilise time and activity dimensions. Strategies 1 and 2 are vector-based, while Strategy 3 is matrix-based. Temporal-based aggregation is discussed in section 5.1.1. The weekly and monthly views are discussion in section 5.1.2. The three strategies are discussed in section 5.1.3.

5.1.1 Temporal-Based Aggregation

It is believed that temporal-based aggregation is an effective way to generate variables in student performance prediction tasks. Clickstream data characterises sequential click action; therefore, temporal-based aggregation is utilised in this research. According to the experimental results, the temporal-based aggregation method leads to an effective predictive model. Also, such an aggregation method allows for a straightforward feature interpretation from educational perspectives (Rodriguez et al. 2021). Due to the course coordinators opening the VLE (Virtual Learning Environment) and releasing access to the course contents before the course officially starts, T is divided into two parts – T0 and T1-Tt. First, T0 indicates the addition of all click counts

on the dates before the course starts. T_0 is interpreted as early learning efforts, such as previewing learning materials or course instructions before the course starts (Park et al. 2017). An early effort in learning is one of the learning strategies that impacts student performance (Gasevic et al. 2017). Second, T_1 - T_t indicates students' click numbers during the course, which is interpreted as students' learning effort over time (Li, Baker & Warschauer 2020).

5.1.2 Weekly and Monthly Views

In this research, the period $T_1 - T_t$ involves the weekly and monthly views, reflecting two collection sizes of click behaviours. The data in the weekly view are sparser than in the monthly view. For example, in T - and TA - datasets, each student is represented in a longer vector (more features) in the weekly view, but a shorter vector (fewer features) in the monthly view. In P -datasets, the weekly view presents one student with a matrix of 40 rows (40 weeks) multiplied by 12 columns. The monthly view presents one student with a matrix of 12 rows (12 months) multiplied by 12 columns.

The best practice in this research case is adopting the weekly view although datasets in this view did not perform as well as the monthly view in some cases. For Strategy 1, in the k -NN, RF, GBT and 1D-CNN models, the monthly view datasets show better accuracy than weekly view datasets. For LSTM, the weekly view reaches 88.58% accuracy, higher than the monthly view 88.07%. For Strategy 2, the monthly view is better than the weekly view when using k -NN, RF, GBT and 1D-CNN. The weekly and monthly view show a similar accuracy performance (88.41% and 88.47%) when using LSTM. For Strategy 3, panel data are unsuitable for all traditional machine learning and 1D-CNN, so the results' weekly and monthly views are not worth comparing. However, for LSTM, the weekly view shows the best model with 89.25% accuracy, significantly higher than the monthly view at 88.67%. According to all the above, although traditional machine learning models learn features better from the monthly view, their accuracies are still lower than the model using weekly view panel dataset with the LSTM algorithm. This result is consistent with some current studies that use a weekly view to represent clickstream data and successfully conduct their prediction tasks (Aljohani, Fayoumi & Hassan 2019).

5.1.3 Feature Engineering Strategies

The three strategies are developed from the involvement of time and activity dimensions. Strategy 1 involves only the time dimension, while Strategies 2 and 3 integrate both time and activity

dimensions. Both dimensions are integrated into a vector of features in Strategy 2 but a 2D matrix in Strategy 3.

First, the experimental result shows that feature engineering strategies involving both time and activity dimensions in datasets leads to better prediction results than only using the time dimension. Strategy 3 using LSTM performs the best among all the experiment models. The second-best is Strategy 2 with GBT. Therefore, Strategies 2 and 3 outperform Strategy 1. That implies that using the information about when students click, and what activity categories students click on, yields better models than only using the information about when students click. This is consistent with the argument of Hung et al. (2020) - features from multiple dimensions are more likely to achieve better predictive results.

It is worth mentioning that Strategy 2 and 3 models are not significantly better than Strategy 1 models in this research case. The best model from Strategies 2 and 3 gives 89.25% accuracy, while the best model from Strategy 1 results in 88.88% accuracy. They are only different by 0.37%. Fewer activity categories being dominant features could be the reason for this slight difference in the accuracy measure. According to feature importance analysis of Strategy 3, only two out of twelve are dominant activity categories – *Act4 (homepage)* and *Act3 (subpage)*. According to feature importance analysis of Strategy 2, only four out of twelve are dominant activity categories – *Act4 (homepage)*, *Act1 (forumng)*, *Act3 (subpage)* and *Act6 (resource)*.

Second, a matrix-based input data structure produces better models than a form of flattening vector while using LSTM. According to the experimental result, LSTM is the only one capable of dealing with all three strategies while producing well-performing models. Comparing the three strategies with LSTM, matrix-based panel data as a reflection of students' click behaviours is the best. Therefore, feature engineering is not simply about creating variables; it is also a process of developing an optimal dataset structure for click-behaviour collection, which significantly impacts prediction results.

5.2 Discussion of RQ2

In this section, the findings of this research are analysed and discussed to answer the second research question: How can machine learning be used to predict student performance to improve teaching and learning? To achieve an effective prediction using clickstream data for students' academic results – fail or pass – this research investigates how to build a predictive model. There are three aspects to the answer to RQ2. The first is feature selection while building predictive

models, demonstrated in section 5.2.1. The second is related to algorithms and models used in modelling with the three feature engineering strategies for clickstream data. This is discussed in section 5.2.2. In this thesis, some significant insights were generated from the findings of important features analysis (section 4.4), which is now used to form suggestions to improve teaching and learning in section 5.2.3.

5.2.1 Feature Selection

Feature selection is a powerful method to considerably reduce the number of features, which allows the model to be easier to understand (Guyon & Elisseeff 2003). For high-dimensional datasets, feature selection is commonly-used for modelling (Zhou et al. 2017). This research involves two parts of feature selection.

The first feature selection part aims to decide whether demographic features should be used in modelling. As a result, demographic features are all filtered out in the best model. That implies that it is possible to achieve the best model using only click behavioural features. The study of Tomasevic, Gvozdenovic and Vranes (2020) adopted the same data source as this research and found that demographic features have little impacts on the prediction model performance, which is consistent with this thesis. According to this finding, there is a possibility of using only behavioural data to effectively predict student performance. The implication of this possibility is to provide evidence of prediction analysis success in education with avoiding potential demographic-based judging, resulting in stereotyping in educational practices (Seidel & Kutieleh 2017).

The second feature selection part involves feature selection methods to acquire optimal models. The Information Gain is used to build all M2 models in the experiment. It is found that the effectiveness of feature selection is varied in different strategies. Table 5. 1 shows a comparison of M1 and M2 models' prediction accuracy and the number of features used. In Strategies 1 and 2, the feature selection method effectively builds a model with improved performance. **In Strategy 1**, with LR and k-NN, all M2 models perform significantly better than their corresponding M1 models using fewer features (highlighted in green in Table 5. 1). With RF and GBT, all the features result in the best accuracy; therefore, M1 and M2 models show a similar accuracy result (highlighted in blue in Table 5. 1). In these cases, the feature selection method helps effectively determine the features for building optimal models.

Table 5. 1 Feature Selection Result Comparison

Strategy	Dataset	Model		LR	k-NN	RF	GBT	1D-CNN	LSTM
1	T-WEE	M1	Accuracy	84.97%	70.44%	88.41%	88.62%	87.61%	88.58%
			(# feature)	(40)	(40)	(40)	(40)	(40)	(40)
	T-MON	M2	Accuracy	86.59%	86.59%	88.45%	88.62%	--	--
			(# feature)	(8)	(8)	(24)	(40)		
		M1	Accuracy	87.14%	70.27%	88.60%	88.88%	88.49%	88.07%
			(# feature)	(10)	(10)	(10)	(10)	(10)	(10)
2	TA-WEE	M1	Accuracy	77.34%	71.35%	70.25%	88.73%	85.79%	88.41%
			(# feature)	(480)	(480)	(480)	(480)	(480)	(480)
	TA-MON	M2	Accuracy	86.52%	86.52%	88.09%	88.73%	--	--
			(# feature)	(10)	(10)	(18)	(480)		
		M1	Accuracy	81.71%	70.64%	87.36%	88.95%	88.19%	88.47%
			(# feature)	(120)	(120)	(120)	(120)	(120)	(120)
3	P-WEE	M1	Accuracy	70.24%	66.29%	70.25%	70.25%	70.25%	89.25%
	P-MON	M2	Accuracy	70.25%	66.10%	70.25%	69.02%	--	--
		M1	Accuracy	70.24%	76.46%	73.82%	76.47%	77.55%	88.67%

In Strategy 2, a similar pattern emerges. With LR, k-NN and RF, all M2 models perform better than their corresponding M1 models and use fewer features (highlighted in green in Table 5. 1). With GBT, M1 and M2 models show a similar accuracy result (highlighted in blue in Table 5. 1). This is because the feature selection method helps effectively determine the use of all features that are able to build the best models.

However, feature selection does not influence the model performance in **Strategy 3**. Traditional machine learning algorithms with P-WEE and P-MON perform the worst (highlighted in grey in Table 5. 1). Although using the feature selection method, these models do not boost the result as the dataset structure of this strategy is not suitable for all selected traditional machine learning algorithms. However, the LSTM & P-WEE achieves 89.25% accuracy without using any feature selection method.

To conclude, feature selection is not always necessary in student performance prediction with clickstream data, although a feature selection method is able to increase model performance in some cases. Some combinations of datasets and algorithms show that feature selection facilitates acquiring optimal subsets to further determine the better models (e.g., the models in the green and blue coloured areas in Table 5. 1). However, when the model's input data is transferred into panel data, using LSTM to conduct a predictive model can achieve an excellent accuracy even without feature selection.

5.2.2 Algorithms and Models

The model performance strongly depends on what input data and algorithms are used. Therefore, it is significant to compare different algorithms to generate insights into the best practice of predictive modelling. Because the weekly view is the best dataset view for this research case, this section compares each strategy-algorithm combination's model performances under this view. **For Strategy 1**, the structure of the dataset is $X^{(T)}$. Each vector indicates each student. The model's accuracy shows $GBT > RF > 1D-CNN > LR, k-NN, LSTM$. **For Strategy 2**, the structure of the dataset is $X^{(TA)}$. Each vector is a combination of time and activity category, indicating each student. The model's accuracy shows this trend: $GBT > RF > LSTM > 1D-CNN > LR, k-NN$. **For Strategy 3**, the dataset structure is $X^{(T) \times (A)}$. Each matrix indicates each student. The model performance shows $LSTM > GBT, RF, 1D-CNN, LR, k-NN$. From these result, the following findings about the algorithms and models are discussed.

k-NN and 1D-CNN

k-NN and 1D-CNN are not ideal algorithms in the case of this research. First, k-NN is one of the most popular algorithms for a course-specific prediction using event-stream data (Marbouti, Diefes-Dux & Madhavan 2016). However, in the case of using clickstream data, k-NN does not show great results. In the three strategies with the weekly view, k-NN models perform the same as the baseline models. After transforming clickstream data into vector-based and matrix-based data structures, k-NN models perform worse than other models. Therefore, k-NN is not the most powerful algorithm for this clickstream data investigation. This finding is similar to some clickstream data studies using k-NN in student performance prediction showing a poor performance (Tomasevic, Gvozdenovic & Vranes 2020).

The experiment results show 1D-CNN's moderate performance in three dataset structures of the three feature engineering strategies – $X^{(T)}$, $X^{(TA)}$ and $X^{(T) \times (A)}$. CNN-based models are commonly used to build models with clickstream data, but little existing research involve 1D-CNN models. Therefore, this research is a novel attempt. This research intends to take advantage of the mechanism of 1D-CNN to deal with sequential features. However, the result of the models in the three strategies is far lower than the best LSTM model.

GBT and RF

Using the clickstream data to conduct the classification task of this research, ensemble classifiers GBT and RF are more capable than base classifiers LR and k-NN. It is observed that GBT and RF

perform better than LR and k-NN in all the strategies in this research. Some current student performance prediction investigations show that boosting classifiers focus on the patterns that are difficult to classify correctly with base classifiers, and they increase accuracy and reduce the misclassification ratio (Imran et al. 2019). This thesis provides empirical evidence regarding the power of ensembles compared with base learners k-NN and LR when using clickstream data. Moreover, GBT is more capable of learning features from the clickstream data of this research case than RF. GBT models' accuracies are always higher than RF models in the three strategies.

LSTM

LSTM is suitable for learning features from panel data with the matrix structure indicating student click behaviours. The capability of LSTM to learn features in predictive modelling using clickstream data relies heavily on the input data structure. With the input data structures of Strategies 1 and 2, the LSTM models do not perform their best. However, when the panel data form is fitted into the LSTM model, it achieves the best performance. This could be related to the capability of LSTM to deal with sequential samples in each panel member. There are similar findings in some other papers. For example, a study of Sarkar and De Bruyn (2021) concludes that LSTM are one of the outstanding candidates for modelling customer prediction using a form of panel data (Sarkar & De Bruyn 2021).

Also, the use of LSTM achieves the effect of using traditional machine learning with complex feature selection processes. LSTM has no feature selection method involved; however, it is better than the result of traditional machine learning with large feature selection workloads. Therefore, when using LSTM to predict student results, feature selection can be optional.

5.2.3 Teaching and Learning Improvement

Based on the feature importance analysis of the three strategies, the findings in terms of important learning sites (activity categories) in the VLE and important study periods during the course BBB are summarised as follows. First, the students' click behaviours on the course's homepage and forum activities are the most important to the prediction tasks. The second-most important are the course's contents and subpages. The rest of the activity categories are of minor importance, meaning that students' click behaviour patterns on those sites have a minor impact on predicting their academic results. These activity categories are quiz, resource, URL, oucollaborate, questionnaire, ouelluminate, glossary and sharedsubpage. Second, from the study period perspective, weeks 4 to 37 are the dominant study periods that impacts students' academic

performance in the course. Within this timeframe, weeks 22-35 are the most significant period. It is observed that this period is in line with stages 4 and 5 of the course assessment (as mentioned in Figure 3. 11 Assessment data summary of the course BBB in Chapter 3), involving the last two assessments in the modules 2013B, 2013J, 2014B, and 2014B. Week 0 to week 3 and week 38 to week 39 are less important. It implies that students' early access to the course BBB is not a significant influential factor in their final performance.

The above findings generate some suggestions to improve teaching and learning, from enhancing the learning environments to facilitating teaching intervention practices. First, the important activity categories can be used to guide the learning environment design. For example, the click behaviours on the course homepage and its subpages might reflect students' habits of interaction with the learning environment of the course – starting from the course's homepage when they start studying. Therefore, some vital course information could be placed on the homepages and subpages to align with students' behaviour habits.

Additionally, the important activity categories and study periods can be used to guide teaching intervention practices. For instance, as the forum is identified as an important category, educators

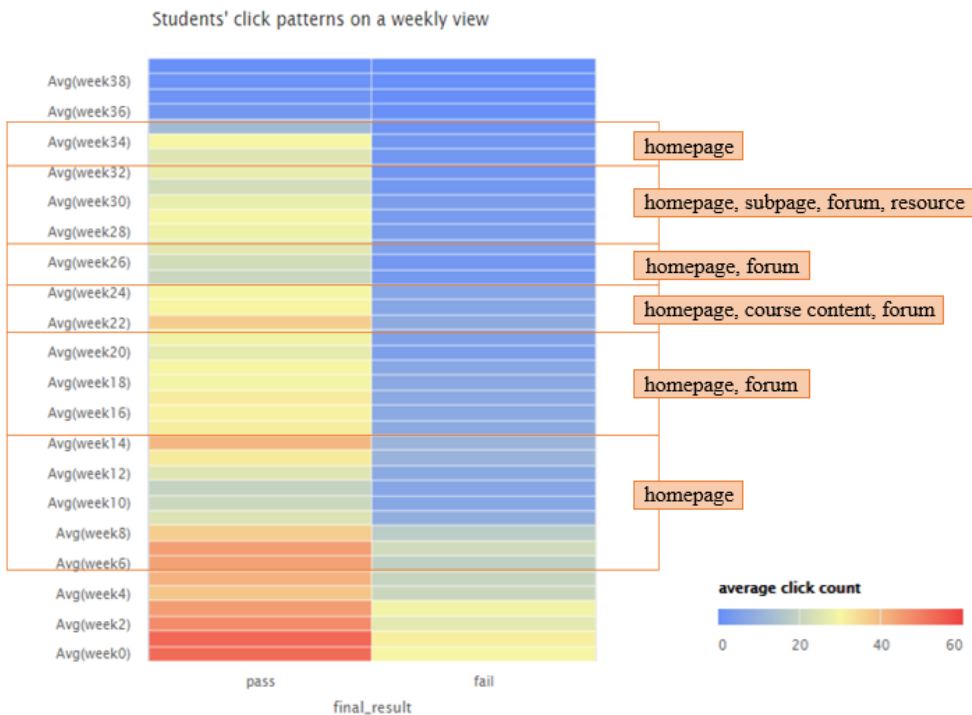


Figure 5.1 An example of teaching intervention practices

or instructors can support their students to engage in the forum activities of the course BBB. Also, educators or instructors may consider assisting students before or during the crucial period of week

22 to week 35, around the time of the last two assessments of the course. Finally, the teaching intervention practices could consider different activity categories for different periods, as Figure 5.1 shows. Such concrete student support from teachers is more likely to help at-risk students pass the course.

5.3 Limitations and Future Research

5.3.1 Limitations

There are limitations to this thesis. First, this research has limited practical demonstrations of data preparation. The OULAD datasets used in this research is relatively clean and there is no enormous data cleansing work involved. In some other cases, first-hand raw clickstream data can be very noisy and might need considerable work dealing with time-stamped clickstream data, and this research lacks such demonstrations.

Second, teaching and learning improvement suggestions are generated based on the understanding of the course BBB from the OULAD description and the original datasets. The suggestions for improving the learning environment design and teaching intervention practices are based on the research's personal beliefs and judgements.

Third, this OULAD clickstream dataset applies to a higher education course structure. The research may be applicable for student performance prediction tasks involving similar tertiary course structures, but some school course structures or informal learning environments (e.g., short-term courses in workplaces) may require adjustments.

5.3.2 Future Research

This thesis informs future practices from three aspects. First, using data including a shorter course period can be explored in future work. This research uses click behaviour data during the entire length of the course to predict students' final results. It is found that passed and failed students' click behavioural patterns are different from week 4 of the course. Therefore, future studies' focus could be on using an earlier course period of clickstream data (e.g., the first ten weeks) to build predictive models to predict at-risk students as early as possible.

Second, feature generation may consider the relationships between weeks. This research assumes that the click behaviours between weeks and months are independent; therefore, independently aggregating click counts for each week/month is adopted. Future research could consider that each week's click behaviours might be connected to the previous weeks' click. The rationale for this consideration is that students' learning seems like a knowledge cultivation process. Therefore, integrating lag sequential analysis can be investigated in future work, such as using LAG to transform time-based clickstream data. For example, each week's click measure can be the sum of the click counts from the previous three weeks.

Third, future research could examine how feature selection influences deep learning models. This research only involves the feature selection method in traditional machine learning algorithms. As a result, LSTM performs the great prediction result without using feature selection. Whether additional advantages of feature selection exist in LSTM and 1D-CNN can be examined in future work.

5.4 Conclusion of the Thesis

This research investigates the potential of clickstream data in student performance prediction, the development of feature engineering strategies, and modelling with machine learning algorithms. This thesis comprehensively examines clickstream data in student performance prediction tasks, from a big picture of related concepts (LA and EDM) to critical aspects of building predictive models. This research explores clickstream data by examining different feature engineering strategies and 60 models using traditional machine learning and deep learning algorithms, as well as feature selection.

Student performance prediction is a sub-topic of LA and EDM. One of the benefits of predicting student performance is identifying at-risk students and then providing them with learning supports. Also, student performance predictive modelling often generates educational insights for teaching and learning improvement. One type of learning behaviour data, clickstream data, has had inadequate investigations in student performance tasks. The motivation of this research is to fill this gap. The objective of this research is to build student performance prediction models with an open-source clickstream data and effective feature engineering strategies.

To build models, firstly, a predictive modelling approach for student performance prediction, named SPPA, is developed. The research methods are in line with the steps of SPPA. Through the first step, data preparation, 5341 students' click behavioural data are determined for modelling. In

the second step, feature generation, time and activity dimensions of the data are examined to ensure the pattern differences between failed and passed students while transforming the data based on the two dimensions. These ground the basic ideas of the feature engineering strategies' development. Sequentially, the data are transformed in three ways, that is, Strategies 1, 2 and 3. Each strategy generates two datasets of the weekly and monthly views so that six datasets (Strategy 1: T-WEE, T-MON; Strategy 2: TA-WEE, TA-MON; and Strategy 3: P-WEE, P-MON) are used in the experiment. Also, demographic features are found to have fewer values, so the use of only click behaviours is designed for the modelling experiment.

In the experiment, 60 models are built, involving the six datasets, six algorithms (LR, k-NN, RF, GBT, 1D-CNN and LSTM), and two options of whether to use feature selection methods (M1 or M2 models). Based on the model performances, this research found that weekly-based click count aggregation with the form of panel data is the best feature engineering strategy. The best practice for this research's prediction case is fitting the panel data into the stacked LSTM model. It achieves up to 90.22% accuracy (89.25% with 0.97% variance), 93.33 F1-score (92.71% with 0.62% variance) and 0.927 AUC (0.913 with 0.014 variance). GBT algorithm achieves the second-best model using P-MON. RF and 1D-CNN perform in medium-level accuracies. However, k-NN is not an ideal machine learning algorithm. Also, feature selection can be optional in the prediction case.

Based on the findings of feature importance analysis of Strategies 1, 2 and 3 in a weekly view, this research found that click behaviours on the homepage, subpage, forum and resources are significant; and the study period from week 22 to week 35 is significant. The course BBB learning environment could be improved by utilising the advantage of the homepage and subpages of this course. Therefore, teaching intervention practices are suggested to provide student support around the significant study period and consider supporting student access or engagement with different activity categories during different study periods.

Appendix 1: TA-WEE – feature weighting result

Feature	Weight	Feature	Weight	Feature	Weight	Feature	Weight	Feature	Weight	Feature	Weight
T29_Act4	1.00	T22_Act2	0.37	T36_Act4	0.24	T16_Act2	0.15	T21_Act7	0.11	T3_Act1	0.07
T30_Act4	0.90	T32_Act3	0.37	T10_Act1	0.23	T19_Act3	0.14	T3_Act4	0.11	T13_Act5	0.07
T28_Act4	0.88	T31_Act3	0.36	T20_Act2	0.23	T21_Act5	0.14	T31_Act7	0.10	T18_Act5	0.07
T32_Act4	0.87	T32_Act6	0.36	T25_Act3	0.23	T15_Act3	0.14	T12_Act2	0.10	T13_Act7	0.07
T22_Act4	0.85	T35_Act1	0.36	T7_Act1	0.23	T15_Act2	0.14	T15_Act5	0.10	T9_Act3	0.07
T34_Act4	0.85	T31_Act6	0.36	T27_Act6	0.22	T19_Act6	0.14	T17_Act5	0.10	T35_Act7	0.06
T23_Act4	0.82	T16_Act1	0.35	T9_Act1	0.22	T22_Act6	0.14	T16_Act5	0.10	T7_Act6	0.06
T31_Act4	0.82	T18_Act1	0.35	T21_Act3	0.22	T16_Act7	0.14	T7_Act3	0.10	T34_Act7	0.06
T24_Act4	0.80	T19_Act1	0.34	T24_Act3	0.21	T30_Act7	0.14	T17_Act3	0.10	T34_Act5	0.06
T33_Act4	0.80	T17_Act1	0.34	T27_Act2	0.20	T32_Act2	0.14	T20_Act7	0.10	T8_Act6	0.06
T27_Act4	0.79	T24_Act1	0.34	T11_Act1	0.19	T23_Act7	0.14	T20_Act5	0.10	T4_Act2	0.06
T26_Act4	0.75	T29_Act6	0.34	T19_Act2	0.19	T13_Act2	0.13	T38_Act1	0.09	T5_Act6	0.06
T21_Act4	0.75	T15_Act1	0.33	T12_Act1	0.19	T28_Act7	0.13	T15_Act6	0.09	T10_Act7	0.06
T25_Act4	0.71	T9_Act4	0.33	T25_Act6	0.19	T4_Act1	0.13	T35_Act6	0.09	T10_Act6	0.06
T19_Act4	0.68	T20_Act1	0.33	T26_Act3	0.18	T38_Act4	0.13	T7_Act2	0.09	T33_Act7	0.05
T20_Act4	0.67	T10_Act4	0.32	T5_Act1	0.18	T23_Act5	0.13	T0_Act2	0.09	T6_Act6	0.05
T18_Act4	0.63	T28_Act3	0.31	T26_Act2	0.18	T24_Act6	0.13	T14_Act7	0.08	T10_Act3	0.05
T17_Act4	0.61	T24_Act2	0.31	T23_Act6	0.18	T21_Act6	0.13	T6_Act5	0.08	T17_Act6	0.05
T16_Act4	0.59	T6_Act4	0.31	T20_Act3	0.17	T18_Act7	0.13	T13_Act3	0.08	T23_Act12	0.05
T32_Act1	0.57	T7_Act4	0.30	T37_Act4	0.17	T27_Act7	0.13	T8_Act2	0.08	T0_Act7	0.05
T15_Act4	0.55	T34_Act6	0.30	T30_Act2	0.17	T29_Act5	0.13	T6_Act3	0.08	T12_Act6	0.05
T13_Act4	0.54	T28_Act6	0.30	T26_Act5	0.17	T15_Act7	0.13	T14_Act5	0.08	T34_Act9	0.05
T34_Act1	0.54	T14_Act1	0.29	T17_Act2	0.17	T2_Act1	0.13	T10_Act2	0.08	T11_Act5	0.05
T33_Act1	0.54	T22_Act3	0.29	T36_Act1	0.17	T18_Act3	0.12	T5_Act2	0.08	T11_Act7	0.05
T29_Act1	0.49	T29_Act2	0.28	T4_Act4	0.17	T24_Act5	0.12	T16_Act6	0.08	T6_Act7	0.05
T14_Act4	0.48	T34_Act3	0.28	T18_Act2	0.17	T22_Act5	0.12	T25_Act5	0.08	T11_Act2	0.05
T31_Act1	0.48	T34_Act2	0.27	T26_Act6	0.17	T17_Act7	0.12	T8_Act3	0.08	T12_Act7	0.05
T30_Act1	0.47	T23_Act2	0.27	T14_Act2	0.17	T31_Act2	0.12	T13_Act6	0.08	T18_Act12	0.05
T22_Act1	0.45	T12_Act4	0.27	T24_Act7	0.16	T32_Act7	0.12	T24_Act12	0.08	T11_Act3	0.05
T35_Act4	0.45	T23_Act3	0.26	T25_Act2	0.16	T22_Act7	0.12	T0_Act6	0.07	T5_Act7	0.05
T28_Act1	0.43	T5_Act4	0.26	T33_Act2	0.16	T1_Act1	0.12	T9_Act2	0.07	T32_Act9	0.05
T30_Act3	0.43	T11_Act4	0.26	T2_Act4	0.16	T14_Act6	0.12	T18_Act6	0.07	T39_Act4	0.05
T21_Act1	0.41	T21_Act2	0.26	T27_Act5	0.16	T35_Act3	0.12	T0_Act3	0.07	T12_Act5	0.04
T26_Act1	0.40	T13_Act1	0.26	T25_Act7	0.16	T0_Act4	0.12	T19_Act5	0.07	T33_Act9	0.04
T30_Act6	0.39	T33_Act6	0.25	T14_Act3	0.16	T30_Act5	0.12	T5_Act5	0.07	T7_Act5	0.04
T29_Act3	0.39	T6_Act1	0.25	T26_Act7	0.16	T19_Act7	0.11	T9_Act6	0.07	T0_Act14	0.04
T8_Act4	0.39	T33_Act3	0.24	T29_Act7	0.15	T37_Act1	0.11	T2_Act3	0.07	T28_Act9	0.04
T23_Act1	0.38	T28_Act2	0.24	T16_Act3	0.15	T6_Act2	0.11	T12_Act3	0.07	T4_Act6	0.04
T25_Act1	0.38	T8_Act1	0.24	T20_Act6	0.15	T0_Act1	0.11	T5_Act3	0.07	T1_Act2	0.04
T27_Act1	0.38	T27_Act3	0.24	T28_Act5	0.15	T1_Act4	0.11	T2_Act2	0.07	T10_Act5	0.04

Appendix 1- continued

Feature	Weight	Feature	Weight	Feature	Weight	Feature	Weight	Feature	Weight	Feature	Weight
T2_Act6	0.04	T30_Act9	0.02	T6_Act14	0.01	T4_Act14	0.00	T39_Act3	0.00	T1_Act12	0.00
T15_Act9	0.04	T14_Act14	0.02	T11_Act9	0.01	T22_Act13	0.00	T17_Act13	0.00	T10_Act12	0.00
T1_Act3	0.04	T17_Act12	0.02	T36_Act6	0.01	T35_Act5	0.00	T33_Act13	0.00	T11_Act12	0.00
T25_Act12	0.04	T29_Act9	0.02	T32_Act12	0.01	T37_Act9	0.00	T10_Act13	0.00	T11_Act19	0.00
T3_Act2	0.04	T33_Act12	0.02	T9_Act9	0.01	T5_Act19	0.00	T7_Act19	0.00	T12_Act12	0.00
T39_Act1	0.04	T27_Act12	0.02	T4_Act7	0.01	T1_Act5	0.00	T19_Act13	0.00	T13_Act12	0.00
T16_Act9	0.04	T18_Act14	0.02	T32_Act13	0.01	T3_Act13	0.00	T10_Act19	0.00	T14_Act12	0.00
T20_Act9	0.04	T19_Act14	0.02	T8_Act14	0.01	T12_Act13	0.00	T13_Act13	0.00	T15_Act12	0.00
T8_Act7	0.03	T17_Act14	0.02	T3_Act5	0.01	T26_Act13	0.00	T13_Act19	0.00	T15_Act19	0.00
T21_Act9	0.03	T13_Act9	0.02	T5_Act9	0.01	T38_Act2	0.00	T14_Act19	0.00	T2_Act12	0.00
T33_Act5	0.03	T31_Act14	0.02	T4_Act9	0.01	T38_Act3	0.00	T19_Act19	0.00	T23_Act19	0.00
T22_Act9	0.03	T20_Act12	0.02	T1_Act9	0.01	T38_Act9	0.00	T21_Act19	0.00	T29_Act19	0.00
T24_Act14	0.03	T27_Act14	0.02	T31_Act12	0.01	T31_Act13	0.00	T22_Act19	0.00	T3_Act12	0.00
T11_Act6	0.03	T25_Act9	0.02	T36_Act2	0.01	T36_Act7	0.00	T23_Act13	0.00	T34_Act19	0.00
T19_Act12	0.03	T31_Act5	0.02	T37_Act3	0.01	T1_Act13	0.00	T25_Act13	0.00	T35_Act12	0.00
T3_Act3	0.03	T33_Act14	0.02	T11_Act14	0.01	T37_Act6	0.00	T26_Act19	0.00	T36_Act12	0.00
T7_Act7	0.03	T21_Act14	0.02	T9_Act14	0.01	T8_Act13	0.00	T27_Act19	0.00	T36_Act13	0.00
T31_Act9	0.03	T35_Act2	0.02	T35_Act14	0.01	T36_Act5	0.00	T30_Act19	0.00	T36_Act14	0.00
T3_Act6	0.03	T26_Act14	0.02	T5_Act14	0.01	T0_Act19	0.00	T31_Act19	0.00	T36_Act19	0.00
T26_Act12	0.03	T19_Act9	0.02	T6_Act13	0.01	T18_Act13	0.00	T32_Act19	0.00	T37_Act12	0.00
T32_Act5	0.03	T8_Act5	0.02	T12_Act9	0.01	T3_Act7	0.00	T35_Act13	0.00	T37_Act13	0.00
T28_Act14	0.03	T8_Act9	0.02	T12_Act14	0.01	T38_Act7	0.00	T35_Act19	0.00	T37_Act19	0.00
T2_Act7	0.03	T24_Act9	0.02	T2_Act13	0.01	T9_Act13	0.00	T38_Act14	0.00	T38_Act12	0.00
T9_Act7	0.03	T26_Act9	0.02	T3_Act14	0.01	T37_Act7	0.00	T39_Act7	0.00	T38_Act13	0.00
T25_Act14	0.03	T4_Act5	0.02	T10_Act14	0.01	T25_Act19	0.00	T4_Act13	0.00	T38_Act19	0.00
T17_Act9	0.03	T23_Act14	0.02	T30_Act12	0.01	T34_Act13	0.00	T4_Act19	0.00	T38_Act5	0.00
T29_Act14	0.03	T23_Act9	0.01	T35_Act9	0.01	T39_Act6	0.00	T5_Act13	0.00	T39_Act12	0.00
T7_Act9	0.03	T10_Act9	0.01	T0_Act13	0.00	T7_Act13	0.00	T6_Act19	0.00	T39_Act13	0.00
T9_Act5	0.03	T3_Act9	0.01	T24_Act13	0.00	T14_Act13	0.00	T9_Act19	0.00	T39_Act14	0.00
T14_Act9	0.03	T13_Act14	0.01	T29_Act13	0.00	T16_Act13	0.00	T21_Act13	0.00	T39_Act19	0.00
T32_Act14	0.02	T22_Act14	0.01	T36_Act9	0.00	T2_Act19	0.00	T1_Act19	0.00	T39_Act2	0.00
T6_Act9	0.02	T15_Act14	0.01	T16_Act14	0.00	T20_Act13	0.00	T17_Act19	0.00	T39_Act5	0.00
T4_Act3	0.02	T2_Act14	0.01	T0_Act5	0.00	T27_Act13	0.00	T20_Act19	0.00	T39_Act9	0.00
T18_Act9	0.02	T36_Act3	0.01	T30_Act13	0.00	T28_Act13	0.00	T11_Act13	0.00	T4_Act12	0.00
T22_Act12	0.02	T21_Act12	0.01	T2_Act9	0.00	T28_Act19	0.00	T12_Act19	0.00	T5_Act12	0.00
T27_Act9	0.02	T28_Act12	0.01	T29_Act12	0.00	T3_Act19	0.00	T16_Act19	0.00	T6_Act12	0.00
T30_Act14	0.02	T20_Act14	0.01	T16_Act12	0.00	T33_Act19	0.00	T18_Act19	0.00	T7_Act12	0.00
T34_Act14	0.02	T7_Act14	0.01	T37_Act2	0.00	T37_Act14	0.00	T24_Act19	0.00	T8_Act12	0.00
T1_Act6	0.02	T1_Act14	0.01	T15_Act13	0.00	T37_Act5	0.00	T0_Act12	0.00	T8_Act19	0.00
T34_Act12	0.02	T1_Act7	0.01	T2_Act5	0.00	T38_Act6	0.00	T0_Act9	0.00	T9_Act12	0.00

Appendix 2: TA-MON - feature weighting result

Feature	Weight	Feature	Weight	Feature	Weight	Feature	Weight
T7_Act4	1.00	T6_Act7	0.19	T0_Act2	0.05	T8_Act13	0.01
T8_Act4	0.93	T3_Act1	0.18	T1_Act2	0.05	T9_Act9	0.00
T6_Act4	0.82	T9_Act1	0.18	T7_Act14	0.05	T7_Act13	0.00
T5_Act4	0.65	T8_Act2	0.17	T3_Act7	0.04	T2_Act13	0.00
T7_Act3	0.63	T2_Act1	0.17	T8_Act14	0.04	T6_Act13	0.00
T8_Act3	0.58	T2_Act5	0.15	T0_Act6	0.04	T0_Act13	0.00
T8_Act6	0.58	T5_Act7	0.14	T0_Act3	0.04	T9_Act14	0.00
T7_Act6	0.57	T3_Act5	0.14	T1_Act3	0.04	T1_Act13	0.00
T8_Act1	0.56	T8_Act7	0.14	T6_Act14	0.04	T0_Act5	0.00
T4_Act4	0.54	T2_Act3	0.14	T2_Act7	0.04	T9_Act5	0.00
T7_Act1	0.45	T4_Act7	0.13	T9_Act6	0.04	T8_Act19	0.00
T6_Act3	0.42	T4_Act2	0.13	T2_Act9	0.04	T4_Act13	0.00
T7_Act5	0.42	T1_Act4	0.12	T5_Act14	0.04	T7_Act19	0.00
T6_Act2	0.38	T4_Act6	0.11	T6_Act9	0.03	T2_Act19	0.00
T6_Act1	0.37	T3_Act2	0.11	T1_Act6	0.03	T0_Act19	0.00
T5_Act1	0.35	T3_Act3	0.10	T8_Act12	0.03	T5_Act13	0.00
T3_Act4	0.34	T3_Act6	0.09	T0_Act7	0.03	T5_Act19	0.00
T6_Act5	0.30	T2_Act2	0.09	T4_Act14	0.03	T3_Act13	0.00
T5_Act3	0.30	T6_Act12	0.09	T9_Act7	0.03	T6_Act19	0.00
T6_Act6	0.29	T2_Act6	0.09	T0_Act14	0.02	T1_Act19	0.00
T5_Act5	0.28	T1_Act1	0.08	T3_Act9	0.02	T3_Act19	0.00
T4_Act1	0.27	T8_Act9	0.08	T7_Act12	0.02	T4_Act19	0.00
T4_Act5	0.27	T0_Act4	0.07	T1_Act5	0.02	T0_Act12	0.00
T2_Act4	0.25	T0_Act1	0.06	T3_Act14	0.02	T0_Act9	0.00
T7_Act2	0.25	T5_Act12	0.06	T2_Act14	0.02	T1_Act12	0.00
T9_Act4	0.24	T8_Act5	0.06	T1_Act14	0.02	T2_Act12	0.00
T5_Act2	0.23	T5_Act9	0.06	T4_Act12	0.01	T3_Act12	0.00
T7_Act7	0.21	T4_Act9	0.05	T1_Act9	0.01	T9_Act12	0.00
T4_Act3	0.21	T9_Act3	0.05	T9_Act2	0.01	T9_Act13	0.00
T5_Act6	0.21	T7_Act9	0.05	T1_Act7	0.01	T9_Act19	0.00

References

Akçapınar, G, Altun, A & Aşkar, P 2019, 'Using learning analytics to develop early-warning system for at-risk students', *International Journal of Educational Technology in Higher Education*, vol. 16, no. 40.

Akram, A, Fu, C, Li, Y, Javed, MY, Lin, R, Jiang, Y & Tang, Y 2019, 'Predicting students' academic procrastination in blended learning course using homework submission data', *IEEE Access*, vol. 7, pp. 102487-102498.

Al-Shabandar, R, Hussain, A, Laws, A, Keight, R, Lunn, J & Radi, N 2017, 'Machine learning approaches to predict learning outcomes in massive open online courses', in *Proceedings of the 2017 IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 713-720.

Aldowah, H, Al-Samarraie, H & Fauzy, WM 2019, 'Educational data mining and learning analytics for 21st century higher education: a review and synthesis', *Telematics and Informatics*, vol. 37, pp. 13-49.

Aljohani, NR, Fayoumi, A & Hassan, SU 2019, 'Predicting at-risk students using clickstream data in the virtual learning environment', *Sustainability*, vol. 11, no. 7238, pp. 1-12.

Antunes, C 2008, 'Acquiring Background Knowledge for Intelligent Tutoring Systems', in *Proceedings of the 1st International Conference on Educational Data Mining (EDM 2008)*, pp. 18-27.

Araque, F, Roldán, C & Salguero, A 2009, 'Factors influencing university drop out rates', *Computers and Education*, vol. 53, no. 3, pp. 563-574.

Bader-El-Den, M, Teitei, E & Perry, T 2019, 'Biased Random Forest For Dealing With the Class Imbalance Problem', *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 7, pp. 2163-2172.

Baumann, A, Haupt, J, Gebert, F & Lessmann, S 2018, 'Changing perspectives: Using graph metrics to predict purchase probabilities', *Expert Systems with Applications*, vol. 94, pp. 137-148.

Behr, A, Giese, M, Tegum K, HD & Theune, K 2020, 'Early prediction of university dropouts - a Random Forest approach', *Jahrbücher für Nationalökonomie und Statistik*, vol. 240, no. 6, pp. 743-789.

Breiman, L 2001, 'Random Forests', *Machine Learning*, vol. 45, pp. 5-32.

Brinton, CG & Chiang, M 2015, 'MOOC performance prediction via clickstream data and social learning networks', in *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 2299-2307.

Broadbent, J & Poon, WL 2015, 'Self-regulated learning strategies & academic achievement in online higher education learning environments: A systematic review', *Internet and Higher Education*, vol. 27, pp. 1-13.

Bucklin, RE, Lattin, JM, Ansari, A, Gupta, S, Bell, D, Coupey, E, Little, JD, Mela, C, Montgomery, A & Steckel, J 2002, 'Choice and the Internet: From clickstream to research stream', *Marketing letters*, vol. 13, no. 3, pp. 245-258.

Burgos, C, Campanario, ML, Peña, Ddl, Lara, JA, Lizcano, D & Martínez, MA 2018, 'Data mining for modeling students' performance: a tutoring action plan to prevent academic dropout', *Computers and Electrical Engineering*, vol. 66, pp. 541-556.

Calders, T & Pechenizkiy, M 2012, 'Introduction to the special section on educational data mining', *ACM SIGKDD Explorations Newsletter*, vol. 13, no. 2, pp. 3-6.

Chan, W-L & Yeung, D-Y 2021, 'Clickstream Knowledge Tracing: Modeling How Students Answer Interactive Online Questions', in *Proceedings of the 8th International Conference on Learning Analytics & Knowledge (LAK21)*, Irvine, CA, USA, pp. 99-109.

Chen, F & Cui, Y 2020, 'Utilizing student time series behaviour in Learning Management Systems for early prediction of course performance', *Journal of Learning Analytics*, vol. 7, no. 2, pp. 1-17.

Choi, SPM, Lam, SS, Li, KC & Wong, BTM 2018, 'Learning analytics at low cost at-risk student prediction with clicker data and systematic proactive interventions', *Educational Technology and Society*, vol. 21, no. 2, pp. 273-290.

Doleck, T, Lemay, DJ, Basnet, RB & Bazelais, P 2019, 'Predictive analytics in education: a comparison of deep learning frameworks', *Education and Information Technologies*, vol. 25, no. 3, pp. 1951-1963.

Dutt, A, Ismail, MA & Herawan, T 2017, 'A systematic review on educational data mining', *IEEE Access*, vol. 5, pp. 15991-16005.

El Fouki, M, Akinin, N & El Kadiri, KE 2019, 'Multidimensional approach based on deep learning to improve the prediction performance of DNN models', *International Journal of Emerging Technologies in Learning (iJET)*, vol. 14, no. 2, pp. 30-41.

Ferguson, R & Clow, D 2017, 'Where is the evidence? A call to action for learning analytics', in *Proceedings of the 7th International Conference on Learning Analytics & Knowledge (LAK'17)*, New York, USA, pp. 56-65.

Filvà, DA, Forment, MA, García-Peñalvo, FJ, Escudero, DF & Casañ, MJ 2019, 'Clickstream for learning analytics to assess students' behavior with Scratch', *Future Generation Computer Systems*, vol. 93, pp. 673-686.

Gao, Y, Cui, Y, Bulut, O, Zhai, X & Chen, F 2022, 'Examining adults' web navigation patterns in multi-layered hypertext environments', *Computers in Human Behavior*, vol. 129, no. 107142.

Gašević, D, Dawson, S, Rogers, T & Gasevic, D 2016, 'Learning analytics should not promote one size fits all: the effects of instructional conditions in predicting academic success', *Internet and Higher Education*, vol. 28, pp. 68-84.

Gasevic, D, Jovanovic, J, Pardo, A & Dawson, S 2017, 'Detecting learning strategies with analytics: links with self-reported measures and academic performance', *Journal of Learning Analytics*, vol. 4, no. 2, pp. 113-128.

Gasevic, D, Tsai, Y-S, Dawson, S & Pardo, A 2019, 'How do we start? an approach to learning analytics adoption in higher education', *International Journal of Information and Learning Technology*, vol. 36, no. 4, pp. 342-353.

Graves, A 2012, 'Supervised sequence labelling', in *Supervised sequence labelling with recurrent neural networks*, Springer, pp. 5-13.

Gupta, A, Gusain, K & Popli, B 2016, 'Verifying the value and veracity of extreme gradient boosted decision trees on a variety of datasets', in *2016 11th International Conference on Industrial and Information Systems (ICIIS)*, pp. 457-462.

Guyon, I & Elisseeff, A 2003, 'An introduction to variable and feature selection', *Journal of machine learning research*, vol. 3, no. 3, pp. 1157-1182.

Helal, S, Li, J, Liu, L, Ebrahimie, E, Dawson, S & Murray, DJ 2019, 'Identifying key factors of student academic performance by subgroup discovery', *International Journal of Data Science and Analytics*, vol. 7, no. 3, pp. 227-245.

Helal, S, Li, J, Liu, L, Ebrahimie, E, Dawson, S, Murray, DJ & Long, Q 2018, 'Predicting academic performance by considering student heterogeneity', *Knowledge-Based Systems*, vol. 161, pp. 134-146.

Hung, J-L, Rice, K, Kepka, J & Yang, J 2020, 'Improving predictive power through deep learning analysis of K-12 online student behaviors and discussion board content', *Information Discovery and Delivery*, vol. 48, no. 4, pp. 199-212.

Hussain, S, Muhsion, ZF, Salal, YK, Theodoru, P, Kurtoğlu, F & Hazarika, GC 2019, 'Prediction model on student performance based on internal assessment using deep learning', *International Journal of Emerging Technologies in Learning (iJET)*, vol. 14, no. 8, pp. 4-22.

Ifenthaler, D & Yau, JY-K 2020, 'Utilising learning analytics to support study success in higher education: a systematic review', *Educational Technology Research and Development*, vol. 68, no. 4, pp. 1961-1990.

Imran, M, Latif, S, Mehmood, D & Shah, MS 2019, 'Student academic performance prediction using supervised learning techniques', *International Journal of Emerging Technologies in Learning (iJET)*, vol. 14, no. 14, pp. 92-104.

Jaggars, SS & Xu, D 2016, 'How do online course design features influence student performance?', *Computers and Education*, vol. 95, pp. 270-284.

Jiang, T, Chi, Y & Gao, H 2017, 'A clickstream data analysis of Chinese academic library OPAC users' information behavior', *Library & Information Science Research*, vol. 39, no. 3, pp. 213-223.

Jonassen, DH 1991, 'Objectivism versus constructivism: do we need a new philosophical paradigm?', *Educational Technology Research and Development*, vol. 39, no. 3, pp. 5-14.

Karim, F, Majumdar, S & Darabi, H 2019, 'Insights Into LSTM Fully Convolutional Networks for Time Series Classification', *IEEE Access*, vol. 7, pp. 67718-67725.

Kemper, L, Vorhoff, G & Wigger, BU 2020, 'Predicting student dropout: a machine learning approach', *European Journal of Higher Education*, vol. 10, no. 1, pp. 28-47.

Khan, A & Ghosh, SK 2020, 'Student performance analysis and prediction in classroom learning: a review of educational data mining studies', *Education and Information Technologies*, vol. 26, no. 1, pp. 205-240.

Kim, Y 2014, 'Convolutional Neural Networks for Sentence Classification', in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Koehn, D, Lessmann, S & Schaal, M 2020, 'Predicting online shopping behaviour from clickstream data using deep learning', *Expert Systems with Applications*, vol. 150, p. 113342.

Kuzilek, J, Hlostá, M & Zdráhal, Z 2017, 'Open University Learning Analytics dataset', *Scientific Data*, vol. 4, no. 170171.

Lee, L-K, Cheung, SKS & Kwok, L-F 2020, 'Learning analytics: current trends and innovative practices', *Journal of Computers in Education*, vol. 7, no. 1, pp. 1-6.

Lei, S 2012, 'A Feature Selection Method Based on Information Gain and Genetic Algorithm', in *2012 International Conference on Computer Science and Electronics Engineering*, vol. 2, pp. 355-358.

Lemay, DJ & Doleck, T 2020, 'Predicting completion of massive open online course (MOOC) assignments from video viewing behavior', *Interactive Learning Environments*, pp. 1-12.

Li, Q, Baker, R & Warschauer, M 2020, 'Using clickstream data to measure, understand, and support self-regulated learning in online courses', *Internet and Higher Education*, vol. 45, no. 100727.

Liu, H & Yu, L 2005, 'Toward Integrating Feature Selection Algorithm for Classification and Clustering', *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491-502.

Livieris, IE, Kotsilieris, T, Tampakas, V & Pintelas, P 2018, 'Improving the evaluation process of students' performance utilizing a decision support software', *Neural Computing and Applications*, vol. 31, no. 6, pp. 1683-1694.

Long, P, & Siemens, G. 2011, 'Penetrating the fog: Analytics in learning and education', *EDUCAUSE review*, vol. 46, no. 5, pp. 31-40.

Lopez-Zambrano, J, Lara Torralbo, JA & Romero, C 2021, 'Early prediction of student learning performance through data mining: a systematic review', *Psicothema*, vol. 33, no. 3, pp. 456-465.

Macfadyen, LP & Dawson, S 2010, 'Mining LMS data to develop an "early warning system" for educators: A proof of concept', *Computers and Education*, vol. 54, pp. 588-599.

Mangaroska, K & Giannakos, M 2019, 'Learning analytics for learning design: a systematic literature review of analytics-driven design to enhance learning', *IEEE Transactions on Learning Technologies*, vol. 12, no. 4, pp. 516-534.

Marbouti, F, Diefes-Dux, HA & Madhavan, K 2016, 'Models for early prediction of at-risk students in a course using standards-based grading', *Computers and Education*, vol. 103, pp. 1-15.

Marbouti, F, Diefes-Dux, HA & Strobel, J 2015, 'Building course-specific regression-based models to identify at-risk students', in *2015 ASEE Annual Conference and Exposition*, pp. 26-304.

Mengash, HA 2020, 'Using data mining techniques to predict student performance to support decision making in university admission systems', *IEEE Access*, vol. 8, pp. 55462-55470.

Miguéis, VL, Freitas, A, Garcia, PJV & Silva, A 2018, 'Early segmentation of students according to their academic performance: a predictive modelling approach', *Decision Support Systems*, vol. 115, pp. 36-51.

Moe, WW 2003, 'Buying, Searching, or Browsing: Differentiating Between Online Shoppers Using In-Store Navigational Clickstream', *Journal of Consumer Psychology*, vol. 13, no. 1, pp. 29-39.

Montgomery, AL, Li, S, Srinivasan, K & Liechty, JC 2004, 'Modeling Online Browsing and Path Analysis Using Clickstream Data', *Marketing Science*, vol. 23, no. 4, pp. 579-595.

Nahar, K, Shova, BI, Ria, T, Rashid, HB & Islam, AHMS 2021, 'Mining educational data to predict students performance', *Education and Information Technologies*, vol. 26, no. 5, pp. 6051-6067.

Najafabadi, MM, Villanustre, F, Khoshgoftaar, TM, Seliya, N, Wald, R & Muharemagic, E 2015, 'Deep learning applications and challenges in big data analytics', *Journal of Big Data*, vol. 2, no. 1.

Namoun, A & Alshanqiti, A 2020, 'Predicting student performance using data mining and learning analytics techniques: a systematic literature review', *Applied Sciences*, vol. 11, no. 1.

Natek, S & Zwillling, M 2014, 'Student data mining solution?—knowledge management system related to higher education institutions', *Expert Systems with Applications*, vol. 41, no. 14, pp. 6400-6407.

Nishimura, N, Sukegawa, N, Takano, Y & Iwanaga, J 2018, 'A latent-class model for estimating product-choice probabilities from clickstream data', *Information Sciences*, vol. 429, pp. 406-420.

Nistor, N & Hernández-García, Á 2018, 'What types of data are used in learning analytics? an overview of six cases', *Computers in Human Behavior*, vol. 89, pp. 335-338.

Oliva-Cordova, LM, Garcia-Cabot, A & Amado-Salvatierra, HR 2021, 'Learning analytics to support teaching skills: a systematic literature review', *IEEE Access*, pp. 1-1.

Pardo, A, Han, F & Ellis, RA 2017, 'Combining University Student Self-Regulated Learning Indicators and Engagement with Online Learning Events to Predict Academic Performance', *IEEE Transactions on Learning Technologies*, vol. 10, no. 1, pp. 82-92.

Park, J, Denaro, K, Rodriguez, F, Smyth, P & Warschauer, M 2017, 'Detecting changes in student behavior from clickstream data', in *Proceedings of the 7th International Conference on Learning Analytics & Knowledge (LAK'17)*, Vancouver, British Columbia, Canada, pp. 21–30.

Park, J, Yu, R, Rodriguez, F, Baker, RB, Smyth, P & Warschauer, M 2018, 'Understanding Student Procrastination via Mixture Models', in *Proceedings of the 11th International Conference on Educational Data Mining (EDM 2018)*, Raleigh, NC.

Park, Y, Yu, JH & Jo, I-H 2016, 'Clustering blended learning courses by online behavior data: a case study in a Korean higher education institute', *Internet and Higher Education*, vol. 29, pp. 1-11.

Pishtari, G, Rodríguez-Triana, MJ, Sarmiento-Márquez, EM, Pérez-Sanagustín, M, Ruiz-Calleja, A, Santos, P, Prieto, L, Serrano-Iglesias, S & Våljetaga, T 2020, 'Learning design and learning analytics in mobile and ubiquitous learning: a systematic review', *British Journal of Educational Technology*, vol. 51, no. 4, pp. 1078-1100.

Prada, MA, Dominguez, M, Vicario, JL, Alves, PAV, Barbu, M, Podpora, M, Spagnolini, U, Pereira, MJV & Vilanova, R 2020, 'Educational data mining for tutoring support in Higher Education: a web-Based tool case study in engineering degrees', *IEEE Access*, vol. 8, pp. 212818-212836.

Qiu, L, Liu, Y, Hu, Q & Liu, Y 2019, 'Student dropout prediction in massive open online courses by convolutional neural networks', *Soft Computing*, vol. 23, no. 20, pp. 10287-10301.

Rangkuti, FRS, Fauzi, MA, Sari, YA, Sari, EDL & Ieee 2018, 'Sentiment Analysis on Movie Reviews Using Ensemble Features and Pearson Correlation Based Feature Selection', in *3rd International Conference on Sustainable Information Engineering and Technology (SIET)*, Malang, INDONESIA, pp. 88-91.

Rodriguez, F, Lee, HR, Rutherford, T, Fischer, C, Potma, E & Warschauer, M 2021, 'Using Clickstream Data Mining Techniques to Understand and Support First-Generation College Students in an Online Chemistry Course', in *Proceedings of the 11th International Conference on Learning Analytics & Knowledge (LAK21)*, Irvine, CA, USA, pp. 313–322.

Romero, C & Ventura, S 2013, 'Data mining in education', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 3, no. 1, pp. 12-27.

Romero, C & Ventura, S 2020, 'Educational data mining and learning analytics: an updated survey', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. e1355.

Sarkar, M & De Bruyn, A 2021, 'LSTM Response Models for Direct Marketing Analytics: Replacing Feature Engineering with Deep Learning', *Journal of Interactive Marketing*, vol. 53, pp. 80-95.

Seidel, E & Kutieleh, S 2017, 'Using predictive analytics to target and improve first year student attrition', *Australian Journal of Education*, vol. 61, no. 2, pp. pp.200-218.

Seo, K, Dodson, S, Harandi, NM, Roberson, N, Fels, S & Roll, I 2021, 'Active learning with online video: The impact of learning context on engagement', *Computers and Education*, vol. 165, no. 104132.

Shimada, A, Taniguchi, Y, Okubo, F, Konomi, Si & Ogata, H 2018, 'Online change detection for monitoring individual student behavior via clickstream data on E-book system', in *Proceedings of the 8th International Conference on Learning Analytics & Knowledge (LAK'18)*, Sydney, New South Wales, Australia, pp. 446–450.

Song, F, Guo, Z & Mei, D 2010, 'Feature Selection Using Principal Component Analysis', in *2010 International Conference on System Science, Engineering Design and Manufacturing Informatization*, vol. 1, pp. 27-30.

Tomasevic, N, Gvozdenovic, N & Vranes, S 2020, 'An overview and comparison of supervised data mining techniques for student exam performance prediction', *Computers and Education*, vol. 143.

Tsiakmaki, M, Kostopoulos, G, Kotsiantis, S & Ragos, O 2021, 'Fuzzy-based active learning for predicting student academic performance using autoML: a step-wise approach', *Journal of Computing in Higher Education*.

Viberg, O, Hatakka, M, Bälter, O & Mavroudi, A 2018, 'The current landscape of learning analytics in higher education', *Computers in Human Behavior*, vol. 89, pp. 98-110.

Vo, C & Nguyen, HP 2019, 'A class-cluster k-Nearest Neighbors method for temporal in-trouble student identification', in *The 11th Asian Conference on Intelligent Information and Database Systems (ACIIDS 2019)*, pp. 219-230.

Vo, C & Nguyen, HP 2020, 'An enhanced CNN model on temporal educational data for program-level student classification', in *The 12th Asian Conference on Intelligent Information and Database Systems (ACIIDS 2020)*, pp. 442-454.

Waheed, H, Hassan, S-U, Aljohani, NR, Hardman, J, Alelyani, S & Nawaz, R 2020, 'Predicting academic performance of students from VLE big data using deep learning models', *Computers in Human Behavior*, vol. 104, no. 106189.

Werner, LL, McDowell, CE & Denner, J 2013, 'A First Step in Learning Analytics: Pre-processing Low-Level Alice Logging Data of Middle School Students', in *The 6th International Conference on Educational Data Mining (EDM 2013)*.

Xu, J, Moon, KH & Schaar, Mvd 2017, 'A machine learning approach for tracking and predicting student performance in degree programs', *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 5, pp. 742-753.

Yang, Y, Hooshyar, D, Pedaste, M, Wang, M, Huang, Y-M & Lim, H 2020a, 'Predicting course achievement of university students based on their procrastination behaviour on Moodle', *Soft Computing*, vol. 24, no. 24, pp. 18777-18793.

Yang, Y, Hooshyar, D, Pedaste, M, Wang, M, Huang, Y-M & Lim, H 2020b, 'Prediction of students' procrastination behaviour through their submission behavioural pattern in online learning', *Journal of Ambient Intelligence and Humanized Computing*.

Yousafzai, BK, Hayat, M & Afzal, S 2020, 'Application of machine learning and data mining in predicting the performance of intermediate and secondary education level student', *Education and Information Technologies*, vol. 25, no. 6, pp. 4677-4697.

Yu, L-C, Lee, C-W, Pan, HI, Chou, C-Y, Chao, P-Y, Chen, ZH, Tseng, SF, Chan, CL & Lai, KR 2018, 'Improving early prediction of academic failure using sentiment analysis on self-evaluated comments', *Journal of Computer Assisted Learning*, vol. 34, no. 4, pp. 358-365.

Yu, L & Liu, H 2003, 'Feature selection for high-dimensional data: A fast correlation-based filter solution', in *The 20th international conference on machine learning (ICML-03)*, pp. 856-863.

Zhang, J, Gao, M & Zhang, J 2021, 'The learning behaviours of dropouts in MOOCs: A collective attention network perspective', *Computers and Education*, vol. 167, no. 104189.

Zhou, Q, Quan, W, Zhong, Y, Xiao, W, Mou, C & Wang, Y 2017, 'Predicting high-risk students using Internet access logs', *Knowledge and Information Systems*, vol. 55, no. 2, pp. 393-413.

Zhu, G, Wu, Z, Wang, Y, Cao, S & Cao, J 2019, 'Online purchase decisions for tourism e-commerce', *Electronic Commerce Research and Applications*, vol. 38, no. 100887.

Zollanvari, A, Kizilirmak, RC, Kho, YH & HernáNdez-Torrano, D 2017, 'Predicting students' GPA and developing intervention strategies based on self-regulatory learning behaviors', *IEEE Access*, vol. 5, pp. 23792-23802.

Zou, X, Hu, Y, Tian, Z & Shen, K 2019, 'Logistic Regression Model Optimization and Case Analysis', in *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, pp. 135-139.