Towards Predicting Student Learning Outcomes from Learning Management System Interactions using Machine Learning

by

Kathryn L. Marcynuk

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg, Canada

To my Dad

Abstract

Advancements in classroom technology have resulted in new types of data collection in educational settings. Along with improvements in the fields of artificial intelligence and machine learning, this educational data can be used to study how we learn and create more personalised learning environments. Starting in March 2020 all in-person courses were abruptly moved to remote instruction in order to combat the COVID-19 pandemic. This influx of students taking remote courses presented a new opportunity to study how students interact with course materials. Remote learning courses at the University of Manitoba are offered using a learning management system (LMS) that centralizes all course activities and files and records user-activities.

The use of machine learning techniques with education-based data is an emerging discipline that offers an opportunity to provide new insights in this area. This thesis presents a code-based tool to create student timelines from raw LMS date-time stamp data and extract features describing student behaviours within a single-term online course. The successes and limitations of these features to predict student grade outcomes were investigated using supervised and unsupervised machine learning models. The LMS data was also explored using neural network-based CNNs and transformers.

The experiments presented in this thesis indicate that students predominately interact with the system at the same time on any given day relative to their previous interaction. The results further demonstrate that temporal features created from LMS interactions can predict student outcomes with greater than random accuracy. The neural network-based classifiers produced more accurate student outcome predictions than the feature-based ML models at the expense of interpretability. This thesis contributes to the body of knowledge on student modelling and prediction, as well as student behaviour within an LMS in an online course, and suggests that educators can help to reduce students' cognitive load and improve students' learning by updating the LMS at a consistent time of day.

Acknowledgments

Like all theses, this work would not have been possible without an entire network of people. To begin, I'd like to acknowledge Dr. Robert McLeod, Dr. Mark Torchia, Dr. Robert Renaud, and Dr. Laleh Behjat for your willingness to serve on my committee and for your contributions therein. Special recognition is due to my co-advisors Dr. Witold Kinsner and Dr. Jillian Seniuk-Cicek for their perspectives and encouragement.

Thank you as well to my friends who have listened, cheered, and commiserated with me along the way. I am deeply grateful that you have supported me to repeatedly put life on hold for this degree, and pick up right back where we left off. While I won't list names here, if you are wondering whether I am referring to you - I am.

I would also like to thank Liz, Judy, and Les for welcoming me into your family with open arms. We didn't know the twists and turns that life was about to throw our way, but you have never wavered in your support.

Thank you to my parents, Debbie and Don, for nurturing my curiosity and resilience (i.e. stubbornness), two necessary qualities to complete any doctoral program. I am so grateful that you taught me to love learning for its own sake, and believed in me every step of the way. Special thanks to Gracie and Edna for reminding me to face your fears, and make your own rules.

Finally, Matt - thank you for continuing to be my cheering squad and sounding board. I'm grateful to get to go on this, and every other, journey with you.

This work was supported by the Vanier Canada Graduate Scholarship, the University of Manitoba Faculty of Engineering, the University of Manitoba Department of Computer Science, and the University of Manitoba Centre for Engineering Professional Practice and Engineering Education.

Contents

List of Figures

List of Tables

List of Acronyms

AGI	*artificial general intelligence*	12, 20, 21, 26
AI	*artificial intelligence*	7, 9–26, 28–30, 33, 35–37, 39, 55, 195
ANN	*artificial neural network*	15, 17, 26
AR models	*auto-regressive models*	200
CAL	*computer-assisted learning*	39
CAPSI	*computer-aided personalized system of instruction*	40
CBT	*computer-based training*	39
CDTs	*cognitive digital twins*	42, A2
CEAB	*Canadian Engineering Accreditation Board*	37, 84, 102, 117
CGPA	*cumulative grade point average*	43
CLT	*Cognitive Load Theory*	35, 198
CMC	*computer mediated communication*	39
CNN	*convolutional neural network*	xvi, 23, 27, 91, 93, 95, 149–160, 171–173, 177, 182–185, 188, 189, 195, 197, 198
CTM	*computational theory of mind*	35
DFS	*Deep Feature Synthesis*	74
DL	*deep learning*	26

EDM	*Educational Data Mining*	39, 41
EER	*Engineering Education Research*	41
ENIAC	*Electronic Numerical Integrator and Computer*	15
FGCS	*Fifth Generation Computer Systems*	21
GAN	*generative adversarial network*	200
GOFAI	*Good Old-Fashioned AI*	14
GPA	*grade point average*	43, 46
IP	*intellectual property*	A1
LMS	*learning management system*	2–5, 7, 10, 31, 38, 40, 41, 44–58, 60, 64–73, 75–82, 87–90, 93, 95–99, 102, 118, 124, 125, 129, 130, 132, 133, 138, 140, 142, 176–183, 187, 189–202
MAE	*mean average error*	120, 124, 125, 143, 144, 146, 147, 149, 181
MCC	*Microelectronics and Computer Technology Corporation*	22

ML	*machine learning*	7–10, 12, 13, 15, 19, 21, 23, 26, 27, 29, 31, 34, 37–39, 41–44, 48–50, 54, 55, 73, 75, 85–87, 94, 95, 118, 119, 135, 136, 143, 160, 177, 181, 182, 191–196, 198–201
MM models	*Markov models*	200
MOOC	*massive open online course*	40, 45, 52, 53, 200
NLP	*Natural Language Processing*	22, 26, 27, 91
NMT	*neural machine translation*	25
NN	*neural network*	22, 26, 27, 33, 182, 192, 195
PCA	*Principal Component Analysis*	86, 135, 138, 140
PDP	*parallel distributed processing*	20
PMF	*probability mass function*	180, 190, 196
RNN	*recurrent neural network*	27, 200
SMT	*statistical machine translation*	22
TELS	*technology-enhanced learning system*	52
VLE	*virtual learning environment*	53

VW deadline	*Voluntary Withdrawal deadline*	66, 67, 77, 78, 83, 88, 96, 102, 103, 140, 172, 173, 175, 189
ZPD	*Zone of Proximal Development*	36

Glossary

burstiness	the degree to which a pattern of activity can be characterized as short periods of time with high levels of activity, followed by longer stretches of time with decreased activity	78
features	numerical characteristics representing student behaviour within the learning management system	72
interaction	a student interaction within the learning management system, defined as a content page access, assignment submission, or quiz submission	65
inter-event time	the amount of time between two successive timeline items, also called an interval	71
interval	the amount of time between two successive timeline items, also called an inter-event time	71
multivariable	data consisting of multiple variables. In the context of this work, the multivariable data consists of Learning Management System timestamps and labels indicating the type of interaction that occured at that timestamp	90
probability mass function	a function of the probability, or likelihood, that a variable will be a particular value	80

spam	refers to unsociliated emails, also known as email spam or junk mail	27

Chapter 1

Introduction

"I'd take the awe of understanding over the awe of ignorance any day."

– Douglas Adams [Adam02]

The study of how we learn is a subject as complex as the human brain itself. It is widely accepted that there is no one-size-fits-all approach to learning, however in large classes there have traditionally been practical limitations for educators to provide individualized attention to students. Recent advancements in classroom technology have resulted in new types of data collection in educational settings. Along with improvements in the fields of machine learning, this educational data can be used to study how students learn, and predict student outcomes, in order to creating more personalised learning environments.

1.1 Problem Statement

In recent years more and different education-based data has become available as educational software, such as learning management systems, have become more prevalent.

At the same time, the fields of artificial intelligence and machine learning have advanced. Techniques from these fields can provide insight into student interactions with learning management software.

1.1.1 Motivation

Traditionally research in the field of education has included labour and time intensive data collection techniques such as interviews, classroom observations, and surveys [FiBR15] [RoVP11][SaFD18]. More recently, the adoption of new education software and technologies has offered new avenues for research. However, the data collected for educational research purposes often requires specialized education software [Kins88] [Kins90], or technology to track mouse movements or eye movements across a screen [HuZZ19] [MaMP20].

Starting in March 2020 many in-person courses, including all offered at the University of Manitoba, were abruptly moved to remote instruction in a response to the COVID-19 pandemic. Remote learning courses at the University of Manitoba were offered using a *learning management system* (LMS) that centralizes all course activities and files while recording user-activity data. The large influx of students taking remote courses presented a new opportunity to study how students interact with course materials.

In large undergraduate classes it can be difficult for a single instructor to provide personalized instruction. Even when every effort has been made to return student work with feedback in a timely manner during the course, students may still struggle without the instructor's knowledge or realize at the end of the term that they have only a narrow path to passing the course. This can result in negative situations such as increased stress and anxiety, temptation to engage in academic misconduct, and ultimately, poor or failing performance in the course.

If data from the *learning management system* (LMS) can be used to identify students'

patterns of behaviour that correlate with or predict failing the course, then those patterns can be intercepted earlier in the term, and additional support offered when it is needed. Supports can be offered and interventions can be planned earlier in the course to potentially set the student back on track, rather than leaving students with limited remedial options near the end of the term. Techniques from artificial intelligence and machine learning have already been successfully applied to a wide variety of fields including commerce and medicine [Kins15]. Similarly, these techniques may be able to use the growing repositories of education data to detect patterns in how students' interact with course material and offer insights into how to set up the LMS to better support students.

1.1.2 Problem Definition

The use of machine learning techniques with education-based data is an emerging discipline that offers an opportunity to provide new insights into learning. The purpose of this work is to study how undergraduate students interact with a learning management system over time during a single-term course. The first objective is to determine if there are time-based features from the LMS data that correlate with and predict student behaviours and learning outcomes in the course, and if it is possible to identify patterns to trigger earlier interventions to provide a more individualized learning experience. The second objective is to explore student outcome prediction from the perspective of time series classification using machine learning techniques without feature engineering.

1.1.3 Proposed Solution

In remote learning environments a student's entire experience in a course may be facilitated through an LMS that is organized by their instructor. Some of the ways that instructors set up the LMS for a course are by posting and organizing content, providing

dropboxes for assignment submissions, creating tests and quizzes, and linking marked work to a grade book. Students interact with the LMS by logging on to the site at times and durations of their choosing, opening files, and submitting assignments and tests. The LMS also records the scores of any marked assessments, and the weighted final grade for each student in the course.

These interactions create a set of user-activity data that is recorded by the LMS as date-time stamps. The dates of the student interactions can be viewed and then analysed using signal processing and machine learning techniques, to analyse learners' patterns of behaviours and correlations to learning outcomes with the ultimate goal of increasing the personalization of learning. The raw data can be transformed into time-based features in code, to view behavioural patterns, which can then be correlated with final grade outcomes. The features will be used in machine learning models to predict student outcomes in the course, and the strength of those predictions will be evaluated. In particular, the interval of time between successive student interactions with the LMS will be calculated and studied. Although previous research has been conducted in the area of student outcome prediction, little is known about LMS intervals and their predictive value in online, synchronous courses [HeHT19] [DeBr20]. As well, the time series created for each student by their LMS interactions will be classified using neural network-based time series classification techniques, as an alternative course outcome prediction method.

1.2 Thesis Formulation

To address the following thesis statement and objectives, this thesis is comprised of three parts: (i) a background and literature review; (ii) an explanation of the experiment design and results; and (iii) a discussion of the results as they relate to the research questions and broader implications.

1.2.1 Thesis Statement

The goal of this thesis is to use the information collected by an LMS in three undergraduate computer science courses to analyse student patterns of behaviour for course outcome prediction. This will be done by developing a tool to model a student in code as a set of numeric features, and performing time series classification on the timelines created by student interactions with the LMS.

1.2.2 Thesis Objectives

There are four main objectives in this thesis:

1. study students' patterns of behaviour in terms of features based on their interactions over time within a learning management system;
2. study the correlations and relationships between students' patterns of behaviour over the entire course and student course outcomes;
3. assess the predictive capabilities of the patterns of behaviour on student success in the course, including how early within the term the student patterns of behaviour can predict student learning outcomes; and
4. assess the predictive capabilities of neural network-based time series classification techniques using with timelines of LMS interactions.

In addition, the following broader questions are included for further discussion:

- What can we learn about students' behaviour by analysing an LMS?
- How can student behaviours on an LMS inform instructors?

1.2.3 Research Questions

Predicting learner behaviour and outcomes is a challenging problem. The following is a list of the topics and research questions addressed in full or part in this thesis.

1. What are students patterns of behaviour on an LMS in an undergraduate computer science course in terms of:
    1.1. Time (over time periods, within a single interaction/time, between interactions, time spent on assignments/quizzes, in relation to course events such as assignment deadlines and VW dates)
    1.2. LMS Interaction and intervals (regular intervals, bursts, consistency,)
    1.3. What are the predominate patterns of behaviour?
2. How are patterns of behaviour over an entire course correlated?
    2.1. Are time and pattern correlated? If so, how?
    2.2. Are time, pattern and events/content (e.g., quizzes, assignments, exams) correlated? How?
    2.3. Are time, pattern, and student grades correlated? How?
    2.4. Are time, pattern, events/content, and student grades correlated? How?
    2.5. Can these patterns be described in terms of high or low engagement?
    2.6. Are there patterns of behaviour that are related with outcomes in the course assuming students are grouped as pass/fail?
3. Which variables and features of behaviours (time, pattern, events) have the greatest predictive capabilities of/are the most highly correlated with student outcomes?
    3.1. What factors predict student success in an online environment?

3.2. Can we define a set of archetypes (student behaviour + course outcome)?

3.3. How early within the term can these factors predict final grades?

3.4. Can these variables and features be used to predict course outcomes of pass or fail before the end of the term, to indicate when intervention may be required?

4. Can student course outcomes be predicted from the timelines of LMS interactions without feature engineering using neural network-based time series classification techniques?

4.1. Can time series classification be used with the full timelines to predict students into one of two groups: passing or failing?

4.2. Can time series classification be used with the full timelines to predict students into one of three groups: passing with a high mark, passing with a low mark, or failing?

4.3. Can time series classification be used for early prediction of students who will fail the course?

1.3 Thesis Organization

This thesis addresses the problem of student modelling and prediction using *machine learning* (ML) models. An introduction to the evolution of *artificial intelligence* (AI), ML, educational learning theories, and how these fields have informed each other is presented in Chapter 2. This chapter also presents the current state of education research using ML methods. Chapter 3 provides an overview of the methodologies used including data collection, the designing and building of a tool that will create student timelines with an LMS and extract representative features of their interactions with the LMS over time, and

how ML models were trained and tested. Experiments were designed to test the features with multiple ML models, and the results are presented in Chapter 4. The analysis of the results is continued into Chapter 5, with a discussion of the findings in view of the research questions and in relationship to the broader fields of education and ML. Chapter 6 provides a summary of the results, suggestions for future work, and concluding remarks.

Chapter 2

Literature Review

This chapter presents the evolution of *artificial intelligence* (AI) and *machine learning* (ML), describing the historically fluctuating success characterised by promising 'waves' and stagnant 'winters' of the leading approaches and technologies, as well as current directions within the fields. Following this, the development of three main learning theories, behaviourism, cognitivism, and constructivism, are presented with parallels drawn to advancements within AI.

The current trends and some future directions for using AI and ML for research in education and personalised learning are then discussed, made possible by the proliferation of digital education data, increased access to educational technology, and growing numbers of online or remote delivery courses. Applications of ML using education data are examined, including student modelling and prediction, decision support systems, and adaptive learning environments. The various types of prediction within education are introduced, including course or cumulative grade prediction and dropout prediction. In particular, the application of ML using different educational data sets for the purpose of student course outcome prediction as seen in the literature is explored. Finally, the sub-area of research into student

course outcome prediction using LMS interaction data is reviewed in more depth and the gap in the literature that this doctoral thesis address are identified.

2.1 Understanding How We Learn

The quest to create "intelligent" computers is really the quest to understand human intelligence: to explore how we think, learn, and rationalize; to define intelligence and be able to conclusively determine whether something, or someone, has that quality or not; and to uncover the fundamental building blocks of the human mind to support human learning, and build machines that mimic, or possibly improve upon human intelligence.

The concepts of learning, cognition, and intelligence have long fascinated scientists across many disciplines, owing to the sheer complexity of these subjects. Yet it is not an esoteric or specialist topic: all of us make assessments in our everyday lives as to whether something has intelligence and, if so, a judgement as to how much. Although it is difficult to articulate how we make these assessments of intelligence, there is a sense of "we know it when we see it". Work in *artificial intelligence* (AI) challenges these assumptions. What we learn from these attempts to create mechanical and digital thinking machines in our own mental likeness can help us to not only advance technology, but shine a mirror on humanity to challenge biases on human intelligence. Further, AI systems can expose, and potentially counteract, implicit biases in human decision-making that occur due to insufficient experience, faulty memory, and mental shortcuts [Bara20].

Understanding the history of AI and ML can help us to better appreciate where the fields are at today, and how they are influencing research in other areas. No longer are AI techniques and tools available only to the largest industrial or academic research laboratories. Now, hardware and software advancements can support the widespread use of many AI algorithms. AI can offer a fresh perspective in many disciplines, including the field of

education.

Just as the dominant theories of intelligence have developed over time, ideas on why and how learning occurs have also evolved. There are multiple educational learning theories under consideration today that are still yet to be resolved into a single unified theory. What is guaranteed, however, is that changes in technology have and will continue to influence learning environments.

More data is being collected than ever before from classrooms, both from software specifically designed for research purposes and from the software that facilitates everyday classroom activities. These data provide a new way of observing what happens in the classroom, and can be analysed in new ways with the growing prevalence and accessibility of AI.

Educators, researchers, and students all stand to benefit from a better understanding of how we learn, and under what conditions we learn best.

2.2 Evolution of Artificial Intelligence & Machine Learning

The fields of artificial intelligence and machine learning have evolved significantly over time. There have been distinct eras of rapid growth in ideas, advocates, and research dollars to tackle big problems. However, when the implied or promised results of AI fail to materialize quickly enough these booms are followed by disillusionment, stagnation, and outspoken critics, with only a few supporters left to carry the field forward. In addition to the boom-and-bust cycles, the history of AI has been heavily influenced by the available hardware, leading psychological theories, and societal values of the time.

Artificial intelligence can be a polarizing topic, one that sparks both passionate curiosity and intense fear. Robot uprisings and HAL 9000 make for compelling fiction [Clar68], but

remain fiction nonetheless. By all credible accounts we are nowhere close to a theoretical point in time in which superintelligent computers threaten the survival of human life, often called the singularity, if such a time is coming at all [Kurz05]. However, concerns about the ethical use and implementation of AI are grounded in reality and will need to be addressed [Kasp17][MaDa19][Mitc19]. Ultimately, AI is a tool and, like any tool, it will be up to those who use it to do so responsibly.

There are many terms used to describe the study of machines that act with some form of intelligence: artificial intelligence, machine learning, computational intelligence, and thinking machines, among others. Each of these terms has risen and fallen in popularity and has its proponents and detractors to this day.

Currently, research into machines that exhibit intelligent behaviour and algorithms to do the same can generally be classified under the term AI. The field includes the lofty goal of *artificial general intelligence* (AGI), which is to artificially create an intelligence on par or exceeding human-level intelligence in all areas. However, most work in AI focuses on solving specific problems, such as machine vision, speech recognition, or translation. Although often used synonymously with AI, the term *machine learning* (ML) is intended to distinguish the branch of AI research in which machines analyse large quantities of raw data for patterns to acquire their own knowledge, rather than being given pre-written rules to follow by programmers [GoBC16]. There are many different names for ML as well, such as pattern recognition, statistical modelling, knowledge discovery, predictive analytics, data science, adaptive systems, and self-organizing systems [Domi15]. Each of these terms has also varied in popularity over time, and within different research communities.

Within this work, the terms “artificial intelligence” and “AI” will be used to broadly describe any efforts to produce algorithms or machines that have or mimic intelligent behaviours. The terms “machine learning” and “ML” will be used distinguish the subset of

AI tools and techniques in which machines acquire their own knowledge. Specific subfields within AI, or within ML, will be named as such.

Section 2.2 provides an overview of the evolution of AI and ML, from the foundations of the field, through the three 'waves' of AI prosperity and two AI 'winters' of slow progress, showing how they have influenced the current state of the field.

2.2.1 Foundations of Artificial Intelligence

The foundations of artificial intelligence and machine learning began long before the first digital computer was invented. Although not typically associated with engineering and computer science today, ancient Greek scholars such as Plato, Socrates, and Aristotle worked to develop philosophy and formalized logic that would later underpin these fields. Meanwhile, the ancient Greek mathematician Euclid is credited with creating the first algorithm, which was to find the greatest common divisor of two numbers [Knut14].

The first widely known machine created in the likeness of a human, called an automaton, was a small mechanical monk built in the 1560s said to be a gift to the church on behalf of King Phillip II of Spain [Wool20]. Attributed to Spanish clockmaker Juanelo Turriano, the automaton was able to repeatedly perform a limited number of human-like movements.

Although the automatons of this time were modelled after their creators, with faces and limbs that moved in human-like gestures, they were still distinctively clockwork machines and not people. However, by the mid-1600s philosophers were starting to ask questions about what it means to be human, and what distinguishes us from other animals or machines. English philosopher Thomas Hobbes described human reasoning as computation, and his French contemporary Ren Descartes declared "cognito, ergo sum" or "I think, therefore I am" [Brit16]. These ideas would lay the groundwork for many branches of philosophy,

as well as future work in cognitive science. German philosopher and mathematician Gottfried Wilhelm von Leibniz directly linked us to machines, postulating that a human is a complex machine in which the mind is a container holding instructions (or "the soul"). In the early 1670s he described and built a new calculating machine called the step reckoner which ran on primitive logic and hypothesized general purpose computers [SwFr17].

However, the next major step towards implementing general purpose computers would not be until the early 1800s. In 1821 English mathematician and engineer Charles Babbage invented his first Difference Engine. He would later go on to theorize the Analytical Engine. English mathematician Ada Lovelace worked with Babbage, and wrote an algorithm during this time for the Analytical Engine that is considered to be the first computer program [Wool20]. Shortly thereafter in the 1840s, English mathematician George Boole introduced the truth variable-based algebra which would later be named after him and would form the foundation for digital computers [Bool47]. In the 1930s, the theoretical work of the previous century was realized into a physical computer with Alan Turing's invention of the universal machine [Turi36].

2.2.2 First wave of Artificial Intelligence

The first wave of AI research began in the 1940s, and the work that came out of this period of time is often referred to as "Classical AI" or *Good Old-Fashioned AI* (GOFAI). Classical AI focused primarily on symbolic, rule-based systems in which knowledge was distilled into axioms by human experts and logically combined by machines to make assertions or decisions. These systems were focused on solving problems that were relatively easy for humans, but more difficult for machines.

However, the first wave of AI research also included work that was inspired by the wetware used in biological learning, particularly neurons and their connections in the brain.

This led to the development of *artificial neural network* (ANN) models. In contrast to rule-based systems with pre-programmed axioms, ANNs attempted to emulate the physical functionality of the brain, albeit on a much smaller scale. The main thrust of the argument behind ANNs is that learning and intelligence in the brain does not need to be completely understood to be modelled artificially, and perhaps intelligent behaviour can emerge from a system created from artificial neurons as it does from a brain comprised of biological neurons. One of the earliest models of brain functionality as an artificial neuron was the McCulloch-Pitts neuron, introduced in 1943 [McPi43][Hebb49]. However, models with early artificial neurons were much simpler and had far fewer connections than the models that exist today due to the computational constraints of the time.

John von Neumann and Oskar Morgenstern published Theory of Games and Economic Behavior in 1944 about game theory as it relates to economics, which would lead to von Neumann's collaboration on the *Electronic Numerical Integrator and Computer* (ENIAC) where he developed a way to store programs on the computer itself [voMo44]. This development was considered the transition from the era of computing as tabulation to the programming paradigm that exists today.

In contrast to neural-inspired algorithms, another early branch of ML was based on creating explicit rules for a system to follow. In 1947, Warren Weaver said:

> One naturally wonders if the problem of translation could conceivably be treated as a problem in cryptography. When I look at an article in Russian, I say: 'This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode' [LoBo56].

Like much of AI, early attempts at machine translation used rule-based expert systems in an attempt to mimic the implicit or explicit rules humans follow in order to complete certain tasks. Both neural nets and rule-based systems would see many successes, and roadblocks,

in the following decades.

Even though articles related to AI topics had been published previously, the article "Computing Machinery and Intelligence" by Alan Turing is considered the first AI publication. It was in this article, published in *Mind*, that Turing proposed the "imitation game" which would later more commonly be referred to as the "Turing Test". Turing wrote: "I propose to consider the question, 'Can machines think?'" [Turi50]. However, Turing noted that part of the difficulty with such a question is in defining the terms "machine" and "think", and therefore he wrote, "instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words." He then proposed an imitation game with three participants consisting of an interrogator, a man, and a woman. The man and woman are hidden from the interrogator, and their voices are obscured. The interrogator must try to determine which of the other participants is the man and which is the woman by asking them questions. If one of the participants is replaced by a machine, Turing asks "Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman?" Followed by, "are there imaginable digital computers which would do well in the imitation game?" By re-framing the discussion in this way, Turing created a mechanism to empirically assess AI systems without first requiring answers to difficult philosophical questions about the nature of thought and intelligence.

In 1955, John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon proposed the Dartmouth Summer Research Project on Artificial Intelligence for the following year [McMR55]. At this gathering, a group of experts in various fields got together to work on Turings questions about what it means to think, how our minds work, and how to teach machines to think like humans. It was also during this meeting that the phrase artificial intelligence was first used to describe this relatively new, and interdisciplinary field. The participants included Allen Newell and Herbert Simon who, along with McCarthy

and Minsky, would later come to be known as the "big four" pioneers of AI. However, all the members officially invited to join the meeting were white men, and so the gathering served as an early example of bias in AI work. In the coming decades, the intentional and unintentional biases introduced into AI systems by their designers, and amplified by a lack of diversity among members of the field, would become an area of study in its own right [Gebr20].

The Golden Age of Artificial Intelligence: The 1956 Dartmouth Summer Research Project on Artificial Intelligence ushered in the "Golden Age" of Artificial Intelligence, which lasted until approximately 1974. During this period of time, the dominant approach to tackling the major issues facing AI was to use a divide and conquer strategy. The idea was that researchers would identify the necessary set of capabilities for intelligent behaviour and work on them separately, rather than trying to build a complete, general AI all at once. Some of these capabilities were: perception (as well as development of sensors and their interpretation); machine learning (i.e. learning from and making predictions about data) [Wool20]; problem solving and planning; reasoning; and natural language understanding. The assumption was that once these individual problems were solved, putting all the pieces into one complete system would be relatively easy. Although that assumption later turned out to be incorrect, it stimulated many advances in artificial intelligence research.

During this time, Allen Newell, Herbert Simon, and Cliff Shaw wrote a program in 1956 called "Logic Theorist" which is considered to be the first artificial intelligence program. Another major milestone during the Golden Age was the introduction of the perceptron by psychologist Frank Rosenblatt [Rose58][Rose62]. The perceptron is considered the first ANN, as it could learn weights iteratively rather than requiring all connective weights to be pre-determined.

A number of significant AI projects were developed during this time. Using the divide and conquer strategy of the era, each project focused on a particular problem within AI. Some of these projects include:

- ELIZA, a novel chat program created by Joseph Weizenbaum at MIT in 1964-1966 that allowed users to type input and receive a response from the computer [Weiz66]. It was one of the first programs that was able to demonstrate the Turing Test, and inspired future generations of chatbot programs;
- SHAKEY, created at the Stanford Research Institute from 1966-1972, was considered the first autonomous mobile robot [KuFH17]. The robot was designed to move around and interact with a controlled environment. As computers were still too big to fit on a moving robot at this time, it used radio to interact with a mainframe that did all the processing; and
- Terry Winograd developed SHRDLU in 1971 to demonstrate problem solving and natural language understanding using an experimental scenario called the "Blocks World" [Wino71].

First AI Winter: In 1965 Gordon Moore published the article Cramming More Components onto Integrated Circuits, in which he observed that the number of transistors doubled within the same area on an integrated circuit approximately every two years [Moor65]. What came to be known as "Moore's Law" provided a quantitative way of showing the pace of technological progress, and how computer hardware was becoming simultaneously computationally faster and physically smaller. However, despite the rapid advances in computing technology, the golden age of AI was failing to live up to the high expectations of the previous decade, which resulted in the reputation of the field taking a serious hit and funds for AI research being significantly cut. As well, after years of hype and fantastical

predictions around AI, "2001: A Space Odyssey" was released in 1968 [Clar68]. Arthur C. Clarke and Stanley Kubrick's thought experiment into how an AI could go wrong was brought to the big screen and began to negatively affect public perception towards the field. These factors led to a period of time now known as the "first AI winter" which lasted from the early 1970s to the early 1980s, during which progress and innovation in AI slowed significantly.

Research in artificial neural nets was hit particularly hard during the first AI winter, so this period of time is often also referred to as the first "neural net winter". Although today ML is considered a branch of AI, many ML researchers consider it a completely separate field, and are even offended to be grouped together with AI. This is likely because of the Minsky and Papert book, *Perceptrons* published in 1969, that was highly critical of neural nets and is credited with the ensuing funding cuts to neural net research [MiPa69]. From that point on, some research communities adopted alternate terms for their work or extended the use of ML to refer to all work related to non-biological intelligence, no longer reserving it just for fields in which machines acquire their own knowledge. To these ML researchers, the field of AI was too closely associated with a list of failed ideas, whereas re-branding to ML put the focus on the techniques that have continued to show promise [Wool20].

The Expert Systems Boom: Research in the late 1970s brought AI back into fashion by focusing on what the previous decades, with a piecemeal problem-solving approach, had ignored: knowledge. Knowledge-based expert systems, which codified domain-specific expert human knowledge to solve very narrow problems, dominated the decade. In part, this approach was popular because expert systems could be profitable when packaged and sold as problem-solving tools with industry-specific applications. An example of an expert system is MYCIN, which was developed at Stanford in the 1970s to assist doctors in diagnosing blood diseases in humans [Shor75].

Typically, expert systems are designed to resemble an interaction with a conversational human expert who is willing to explain their reasoning. They are characterized by the following behaviours: the system will ask a series of questions, to which the user provides responses; the system uses the input to arrive at a conclusion by following a set of pre-programmed rules; the system has some ability to cope with uncertainty in the input. The expert system is able to show how it arrived at an answer by displaying the path through the pre-programmed rules. This transparency is a key characteristic that distinguishes expert systems from other types of AI such as neural networks. Expert systems were followed in the late 1970s by logic-based AI systems, also called logical reasoners, as a way to reason with and make deductions from the knowledge base rules.

The largest and longest-running knowledge-based AI system is Cyc, which was started in 1984 by Douglas Lenat. The Cyc hypothesis was that the problem of artificial general intelligence (AGI) was mostly one of knowledge, and therefore could be solved by a knowledge-based system [LePS86]. Although Cyc continues to this day, by the end of the 1980s the majority of these types of systems fell out of fashion because of the difficulty of extracting human knowledge and encoding it into rules.

2.2.3 Second wave of Artificial Intelligence

Whereas the first wave of artificial intelligence was dominated by systems that attempted to mimic human behaviours and knowledge representation, the second wave was dominated by statistical learning and inspired by human biology - particularly, our brains. Connectionism, also called *parallel distributed processing* (PDP) was popular during the 1980s to 1990s. The idea behind PDP systems is to network many simple computational units together in order to achieve intelligent behaviours, like many neurons working together in a brain.

The late 1980s also brought robotics back into the spotlight in AI. Unlike the previously popular knowledge-based approaches, such as expert systems, which created logical systems completely independent of their environments, in this new paradigm the behaviour of the robot was directly related to its environment or situation. However, the drawback was that this type of AI does not scale well, due to the physicality of the robots. This led to the rise of agent-based AI in the early 1990s. Combining behavioural AI and logic-based AI, agents had to be reactive, proactive, and social. Agent-based AI was the precursor to modern software agents like Apple's *Siri* and Amazon's *Alexa*. However, in this paradigm the goal was not necessarily to make choices that a human would make, but to make the optimal choice. The main thrust of AI research today remains rooted in agent-based AI, in which agents are built to help people and act on their behalf, rather than focusing only on the lofty goal of artificial general intelligence (AGI). Better ways to optimize decision-making, especially in the face of uncertainty, is still an active area of research, such as self-driving cars or recommender systems. In the 1990s, the main way to deal with this uncertainty was by using Bayesian inference to repeatedly update statistical models and calculate the likelihood of relevant events as new information became available to the system.

It is worth noting that progress in AI and ML has always been directly connected to technological advances in physical computer systems. Increases in available computing speed and memory open the door to new ML research possibilities, whereas other branches of research can be impeded or halted if the current hardware is unable to support them. Computer hardware was continuing to advance rapidly throughout the twentieth century and in 1981, near the beginning of the connectionism wave, Japan announced the *Fifth Generation Computer Systems* (FGCS) project which was launched a year later [Garv19]. Each generation of computer systems represents a significant step forward in computing capabilities. The first four generations of computers were vacuum tubes, transistors, integrated circuits, and very large integrated circuits. The FGCS was aimed specifically at

creating computers that could support work in artificial intelligence. In response, the USA formed the *Microelectronics and Computer Technology Corporation* (MCC) in 1982, which lasted for twenty-two years. The MCC advanced computer systems research in the West on several fronts, one of which was intelligent systems.

Along with advancements in computer hardware, another development that supported neural nets during the second wave of artificial intelligence was the introduction of back-propagation to systematically adjust the weights of connections in a multi-layer *neural network* (NN), and therefore train them more efficiently. This concept had been explored by multiple researchers, such as Paul Werbos, throughout the 1970s [Wass89], although the work during that time was done independently and led to a duplication of effort. However, in 1986, David Rumelhart, Geoffrey Hinton, and Ronald Williams popularized the concept [RuHW86], and Yann LeCun proposed the modern form of backpropagation the following year [Lecu88].

The introduction of backpropagation led to a resurgence of interest in NN projects, creating the second wave of AI. Some projects of note during this time include:

- 1987 1995 The PROMETHEUS project was one of the first driverless cars. Created in Europe and funded by a multi-government cooperation entity called EUREKA, the car drove from Munich, Germany to Odense, Denmark and back. Although human intervention was required on average every 5.5 miles, there was a stretch 100 miles that required no human intervention.
- 1990s The idea of *statistical machine translation* (SMT) was introduced as a branch of *Natural Language Processing* (NLP) in which the translation algorithm learns from the data rather than using human-created rules. SMT algorithms are trained on large sets of content that has been translated by humans and learn to translate future input using statistical models based on the context of the text [Koeh10]. By the early twenty-

first century most major machine translation programs used this method, including Google Translate and Microsoft Translator, before it was surpassed by other methods in the mid-2010s [Wool20].

- 1994 The checkers-playing program CHINOOK drew against current World Checker Champion Marlon Tinsley in all six championship games. Shortly after, Tinsley withdrew from the competition and CHINOOK became the first program to be awarded the world championship title in checkers [ScLL96].
- 1997 The chess-playing computer Deep Blue, created by IBM, defeated World Chess Champion Garry Kasparov in a highly publicised match. [Kasp17]. The game of chess had been considered a benchmark problem in AI as it posed a more significant challenge for researchers compared to checkers due to far greater number of possible moves at each turn.
- 1998 An early CNN called LeNet is introduced by French computer scientist Yann LeCun, which is used to recognize handwritten characters [LeBB98]. CNNs would become an more prominent during the third wave of AI.

Second AI winter: By the late 1990s AI was considered a mature field, primarily because of DeepBlue and the development of SAT solvers. SAT solvers were named after the boolean satisfiability problem, which is the NP-complete problem "of checking whether simple logical expressions are consistent" [Wool20].

However, at this time, once again too many ambitious claims had been made of AI that could not be fulfilled, and it began to lose popularity. Although some areas of ML were still thriving, the attention was no longer on neural nets because computers were not yet powerful enough to implement the new ideas in that area. At this point AI was dominated by statistics, rather than philosophy, logic, or cognitive science as before. It remained this

way until the new millennium, when computer hardware caught up and opened new avenues of research in neural nets and deep learning.

2.2.4 Third wave of Artificial Intelligence

After a few years out of the spotlight, AI, and specifically neural networks, made a comeback in a third wave starting in 2006 that continues to this day. The third wave of AI is focused on contextual adaptation and was born out of developments with deep neural networks and deep learning.

Deep learning was championed by Canadian computer scientists Geoffrey Hinton and Yoshua Bengio, along with French computer scientist Yann LeCun [GoBC16]. Collectively they are known as the "Godfathers of Deep Learning" and even the "Godfathers of AI". Compared to previous generations of multi-layer neural networks, deep learning has more layers, more neurons, and more connections. The deepening of neural networks was made possible by advances in computational power and larger datasets, which in turn meant that algorithms that were previously impracticable to implement could now be used. Figure 2.1 shows the relationship of deep learning and other fields within the larger context of AI.

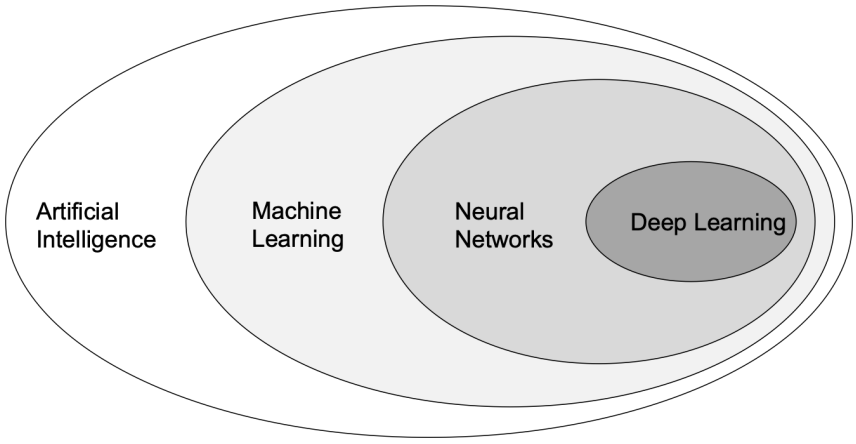

Fig. 2.1: Relationship between the major fields within artificial intelligence, including machine learning and deep learning. (Partially adapted from [GoBC16])

By 2016, it was a widely accepted rule of thumb that it would generally take about 5000 labelled examples per category for a supervised deep learning algorithm to achieve acceptable performance, and when trained with a dataset containing at least 10 million labelled examples it would match or exceed human performance [GoBC16].

It was also around this time that artificial intelligence left the research laboratories and began entered our homes. Google Translate was first made available in 2006 and improved over time by adopting a type of deep learning in 2016, that would be come to be used in many translation projects, called *neural machine translation* (NMT) [WuSC16]. Already by 2010 Google was testing deep learning for speech recognition, and within two years similar technology was commercialized and used by multiple companies in personal products such as Apple's *Siri* and Amazon's *Alexa* [MaWa19].

In addition to commercial products, several significant milestones in artificial intelligence research captured the public interest during the 2010s. These include:

- In 2011, IBMs Watson beat Ken Jennings at Jeopardy live on television [Chan14].
- In 2014, Google sparked major business and press interest in AI by acquiring the United Kingdom-based company DeepMind for a reported $650 million. DeepMind was first founded in 2010 with the mission statement "to solving intelligence, to advance science and benefit humanity".
- Also in 2014, a computer program called AlphaGo beat professional Go player Fan Hui. More complicated than checkers or chess, mastering the game of Go was seen as a benchmark problem in AI research [Natu16][Gibn16].
- On February 14th, 2016, a "social humanoid robot" named Sophia made by researchers in Hong Kong was first turned on. In October 2017 she was granted Saudi Arabian citizenship and became the first robot to be granted citizenship in any country. Although

this was largely a publicity stunt, with no claims that Sophia possessed human-level intelligence, the move indicated a growing societal acceptance for AI research [Gres18].

- 2017 Final ImageNet competition (winning program had a 98% accuracy). The ImageNet archive of images is classified into categories using WordNet.
- Also in 2017, Google created a new program to play Go called AlphaGo Zero (AGZ). This version was able to significantly beat its predecessor, AlphaGo [SiSS17]. The same year, the DeepMind team within Google announced AlphaZero (AZ) which was a more generalized version of AlphaGo Zero capable of playing multiple games at a superhuman level [SiHS17].

2.2.5 Artificial Intelligence Today

Today there is still strong interest in AI, in both research and commercial fields. Advancements in computing hardware have revitalized work in many fields of AI such as *artificial general intelligence* (AGI), knowledge-based systems, robotics, computer vision, and *Natural Language Processing* (NLP), among others [Wool20].

ML has received particular attention, and has been applied to an increasingly wide field of tasks as ML software libraries and processing power have become more commonly available. Within ML, *deep learning* (DL), and artificial neural networks (ANNs, or NNs for short) continue to be popular as part of the third wave of AI. Just as the ability to train NNs that are deeper and more connected has grown over time, newer NN models have also been designed that can learn from sequential data. These NNs can leverage underlying relationships that may exist within sequential data, such as in natural language data or time series data, rather than treating each piece of input independently.

One area of interest that has been made possible through the development of deeper

NNs is time series classification. **Classification** is a common ML task of recognizing and categorizing input data into groups or "classes". For example, a classifier can be used to assign an e-mail to one of the groups of "**spam**" or "not spam" [ReRa17]. Time series data are sequences of data points collected chronologically, such as speech, biomedical data, or stock market prices. The task of categorizing these sequences is known as **time series classification**, which is considerably more difficult than general classification due to the larger data sets involved that have higher dimensionality [Sado19].

Convolutional neural networks (also known as ConvNets or, more commonly, CNNs) have been used for time series classification tasks in a variety of fields [GoBC16]. These deep NNs were first introduced in 1998 and are named after their use of convolution instead of general matrix multiplication in one or more of the hidden layers [LeBB98]. Originally inspired by the biological visual cortex, CNNs were most commonly associated with image recognition tasks before being applied to other realms including time series classification [Masl21]. CNNs have shown to be effective at processing information that has a "grid-like topology" such as a time series, which can be thought of as a one-dimensional grid of data, or images consisting of two-dimensional grids of pixels [GoBC16].

The transformer is another type deep NN that is starting to be used for time series classification. Introduced in 2017, transformers were developed by a team at Google Brain to process sequential data [VaSP17]. Transformers were first used in NLP tasks where previously a *recurrent neural network* (RNN) was the most common choice. One of the main advantages of transformers is that they process sequential input data all at once, rather than in sequence as is done in RNNs. This reduces the relative time that it takes to train a transformer by processing the input in parallel, which is particularly useful for large datasets. However, transformers have quickly been applied to other areas with sequential input data including computer vision and tasks involving time series data [WeZZ22].

Fourth Wave of Artificial Intelligence: Time will tell if we are currently still in the middle of the third wave of AI, or entering a fourth wave already. If the latter, the fourth wave is largely an extension of the third, without the waves being delineated by a characteristic winter in between. The fourth wave is sometimes referred to as "Autonomous AI". During this wave, AI technology is expected to have more decision-making capabilities with less human oversight. Therefore, there will also be an additional push towards transparency, and more focus on the ethical and moral implications of AI than before.

As the field of AI has grown, so too has the need to create common measures and classification schemes to support a common framework for discourse. One example is the Winograd schema challenge by Hector Levesque, with Ernest Davis and Leora Morgenstern. First proposed in 2011, the Winograd schema challenge consists of questions to test whether a machine can think [Leve14][MoDO16]. As well, the Turing Test from 1950 continues to be an enduring, although no longer elusive, goal. In 2014 a chatbot named "Eugene Goostman", created by Ukrainian and Russian programmers, was publicly acknowledged to have passed the Turing Test in a competition held by the Royal Society of London [WaSh16].

As another example, by the 2010s autonomous cars were starting to become feasible, albeit with some level of driver oversight. To describe and track this progress, in 2014 SAE International produced a six-point scheme to classify levels of vehicle autonomy from no autonomy to full automation [HoSc21].

In recent years the moral and ethical implications of artificial intelligence research, and products derived from it, has become an area of research in its own right. In 2015 the Asilomar principles were first developed for ethical AI research and since then many companies have come up with their own AI ethics guidelines. In general, the guidelines are based on the following core tenets: accountability, responsibility, and transparency [Wool20].

2.3 Evolution of Learning Theories and their Relationship with AI

As humans we are constantly learning new concepts, new ideas, new facts, new connections, new people, new places, and more. Sometimes what we learn is simply new to us, and at other times the knowledge learned is new to everyone. Learning involves being exposed to some new knowledge, either on purpose or by happenstance, processing or integrating it with what we already know, and remembering it for the future. Learning can happen quickly or gradually, perhaps with repeated attempts. It can be tempting to say that we can recognize learning when we see it, but it is more difficult to explain why and how it happened.

There are multiple fields that study what learning it is, how it occurs, and how it can be supported. Branches of psychology are concerned with biological-based learning, while education research focuses on how humans learn both naturally and in structured environments. The field of AI, and the related subfield of ML, are concerned with algorithms and machines for non-biological-based learning. These fields influence each other to advance our overall understanding learning. The dominant theories of the time in psychology and education have led to the development of new ML techniques and can be seen reflected in the waves of AI. In return, the success of these techniques can provide insight into human learning.

Over time, several educational learning theories have been developed to describe, understand, and predict how humans integrate and retain new knowledge. The main educational learning theories can be grouped into three categories of behaviourism, cognitivism, and constructivism. Other prominent learning theories, such as humanism and connectionism, are largely considered to be variants of, or responses to, the main three theories. However, some researchers and educators prefer to distinguish some of these variants and propose a

count of five or more main learning theories.

Section 2.3 explores the main educational learning theories of behaviourism, cognitivism, and constructivism in more detail and their significance to AI.

2.3.1 Behaviourism

The earliest of the three primary learning theories is behaviourism. Behaviourism was first formally introduced in the early twentieth century and is credited to John B. Watson who wrote that the "theoretical goal [of behaviourism] is the prediction and control of behavior [sic]" [Wats13]. This school of thought focuses on behaviours that can be objectively observed, without attempting to speculate about what is going on inside the mind [SiAd13]. Through behaviourism, Watson approached psychology as an objective and experimental science. In doing so, behaviourism offered an advantage over earlier trends in psychology which focused on introspection that were more difficult to validate by way of experiment.

Some of the early work in AI during the twentieth century appears to be influenced by behaviourism. For example, the "imitation game" introduced by Alan Turing in 1950 was focused only on observable behaviours to facilitate experimentation [Turi50]. The concepts of behaviourism continue to underpin work in AI today. Intelligent systems are trained using data collected from human behaviours, such as purchasing habits. Indeed, one of the most noticeable examples is in the realm of advertising, where it is common to see phrases such as "items similar to your previous purchase" or "similar customers also enjoyed". Recommender systems are based on the idea that people will be internally consistent in their behaviours, and that people who behave in the similar ways will be interested in the same or similar recommendations. For example, in an educational environment, recommendations can be made to students based on their behaviour as it is observed through their time-

stamped interactions with course materials in an LMS, and these interactions can be studied as a proxy for student learning.

Behaviourism provides a lens to study learning from externally observable behaviours and actions, without any knowledge of the internal thoughts of the learners. As such, the collection of student behavioural data with specialised software (such as LMS timestamps, mouse-click records, or eye movement tracking) to be used for student outcome prediction, is rooted within the field of behaviourism. The underlying assumption in this doctoral research is that if observable student behaviours can be used to predict student learning outcomes (i.e. successfully passing a course), then those same behaviours can serve as a proxy for learning. Studies that use LMS information as a representation of student learning are explored further in Section 2.4.

Behaviourism has also influenced the development of ML techniques such as reinforcement learning, in which desirable outcomes are reinforced and undesirable outcomes are discouraged during training of ML models. This type of ML was inspired by biological learning and the theories of classical conditioning, learning laws, and operant conditioning.

Classical Conditioning: The work of Russian physiologist Ivan Pavlov in classical conditioning, also known as respondent conditioning, formed one of the fundamental building blocks of behaviourism [Pavl27]. First published in 1897, his experiments were initially directed towards non-human animal behaviour and learning. However, these findings were later incorporated into educational learning theories for humans as well. Indeed, Pavlov's experiments with dogs, food, and bells were so straightforward yet striking that his name is now synonymous with classical conditioning both in the field of psychology and in the public consciousness. In these experiments, Pavlov demonstrated that an involuntary response can be learned as a response to a repeated external stimulus.

Learning Laws: Around the same time as Pavlov's work in classical conditioning, an American psychologist named Edward Thorndike was working on concepts that would help lay the foundation for later work in educational psychology, and would heavily influence the field of behaviourism. During his work with non-human subjects in comparative psychology, Thorndike proposed the "Law of Effect" which posits that behaviour is influenced by consequences [Broc20]. In this paradigm, behaviour that produces a "satisfying" result will be "stamped in" to the animal, whereas behaviour that produces an "annoying" or "unsatisfying" result will be "stamped out" [WaTS07].

Operant Conditioning: In the mid-twentieth century B.F. Skinner experimented further with types of behaviourism and developed many advances in the field. He was influenced by Thorndike, and among his many credits is the introduction of operant conditioning. Skinner proposed that behaviour could be defined as the observable movements and actions of an organism as it interacts with the world [Skin38]. Through operant conditioning, an external stimulus is used to motivate a desired behaviour: either a positive stimulus to motivate a desired behaviour or a negative stimulus to discourage an undesired behaviour. However, unlike in classical conditioning, which is based on involuntary responses, operant conditioning assumes that the organism has a choice in how to respond.

Although both respondent behaviours from classical conditioning and operant behaviours from operant conditioning have their origins in the study of animal behaviour and are about how animals learn as a reaction to stimuli, they are related but not the same. Classical conditioning is primarily concerned with involuntary responses, whereas operant conditioning involves the choice to do something to get a reward or avoid a punishment [WaTS07].

Additional Learning Theories Related to Behaviourism: As the field of behaviourism evolved, several variations gained traction and became independent fields of study, some-

times overlapping with other learning theories as well. These include connectionism, social learning theory, and humanism learning theory, all of which have impacted AI in various ways.

Connectionism, credited to Edward Thorndike, is one such learning theory. Like behaviourism, the connectionism learning theory is based on the ideas of stimuli and responses. However, the focus in connectionism is on the associative relationship between pieces of knowledge or events [Thor54]. That is, if two ideas are associated with each other, then thinking abut one idea in the pair is likely to lead to thoughts of the associated idea. The association may be due to similarity, an alikeness between the ideas, or contiguity, meaning they are proximally or sequentially related. These ideas of association have been discussed by philosophers for centuries, tracing back to Aristotle, and were tested by Thorndike in his creation of the connectionism learning theory. Thorndike described the idea of a neural bond, "between one mental fact and another" [Thor13] and learning as the creation and strengthening of these bonds. As such, connectionism can be considered as a step between purely behaviourist models and later constructivist models [Thor54]. Connectionism is closely associated with *neural network* (NN) research, which is predicated on the weight, or strengths, of bonds between artificial neurons to produce intelligent behaviour.

Social Learning Theory is considered a subtype of behaviourism developed by Albert Bandura. It builds on the behaviourist theories of classical conditioning and operant conditioning to include the idea of observational learning [Band77]. In observational learning a student learns from watching the behaviours and actions of someone else. The student may then model the observed behaviour, or otherwise learn from it. Bandura described four conditions that impact observational learning: the attentiveness of the student in watching the behaviour; the ability of the student to remember what they have observed; the ability, physically and mentally, of the student to repeat the behaviour; and the motivation of the student to engage in the behaviour, possibly due to external stimuli. Tenets of social learning

theory can be found in ML, such as in the field of reinforcement learning [BoPM17].

The concept of motivation underlies the **Humanism learning theory**, which was developed as a response to behaviourism. Humanism is based on Maslow's Hierarchy of Needs, which consists of five levels of basic needs: physiological, safety, love, esteem, and self-actualization [Masl43]. Maslow postulated that human behaviours and actions are motivated by a desire to satisfy one or more of these needs at any given time. Humanism was developed by Carl Rogers and James F.T. Bugental, who built on Maslow's model to consider learning in the context of the individual as a whole and take into account the learner's individual preferences and motivations [Roge79]. Of all the learning theories, humanism is uniquely human-centric with its foundations in concepts such as self-actualization. It provides a contrast to other learning theories that could conceivably be applied to processes that look like learning in non-biological, or non-sentient, entities such as machines.

2.3.2 Cognitivism

Cognitivism emerged in the mid-twentieth century, as a response to behaviourism, and by the 1960s-1970s had surpassed behaviourism as the leading trend in psychology. While behaviourism focused on outwardly observable aspects of human behaviour, and dealt with thinking as a behaviour, cognitive theories recognize thought processes as a separate and important factor in learning. Cognitivism is interested in "how learning occurs in the brain", including how new information is received, processed, and stored in memory [SiAd13]. Therefore cognitivism is concerned with how external factors, including stimuli and observable behaviours, as well as internal mental states and internal processes impact learning.

The learning process is central to cognitivism. An example of this is Robert Gagne's model of learning that link nine events of learning with particular cognitive processes [Gagn85].

How information is organized when presented to the learner is considered to be important to facilitate learning. David Ausubel introduced the idea of the "advance organizer", which is a way of helping students to connect new information to what they already know [Ausu68]. During the presentation of new material, cognitivism posits that how that information is organized impacts the learning process. To improve retention, larger or complex topics should also be broken down into smaller parts. In contrast, concepts that are presented in a haphazard way or as a mixture of relevant and irrelevant information may be difficult for the student to process and store for later retrieval.

Memory also plays an important role in cognitivism, in which forgetting can be seen as failure to retrieve information or a mechanism to remove pieces of information that are no longer relevant. *Cognitive Load Theory* (CLT) describes knowledge as being transferred from short-term memory, or working memory, to long-term memory [PaSe21] [Kirs02]. The cognitive load is the amount of working memory that is required of the learner. Working memory has a limited capacity and retention time, and information must be encoded into long-term memory if it is to be remembered in the future. If there is too much information for the working memory to hold, some information will necessarily be dropped before it can be learned by being encoded into the long-term memory [PlMB10].

One way in which the mental processes that are central to cognitivism can be understood is through the theory of computationalism, also called the *computational theory of mind* (CTM) [Wats08]. In this theory, biological neural activity is a type of computation which in turn produces cognition. The idea was first proposed by Warren McCulloch and Walter Pitts, who also invented one of the earliest artificial neuron models [McPi43]. Cognitivism was also fundamental to the foundation of the interdisciplinary field of cognitive science, which studies intelligence drawing from multiple fields including AI to better understand mental processes including learning.

2.3.3 Constructivism

The most modern of the three major learning theories, constructivism was introduced in the 1980s and is linked to Jean Piaget's stages of cognitive development. In the constructivism approach, the learner builds on previous knowledge and experience to construct an understanding of new ideas and concepts [SiAd13]. An example of this is a spiral curriculum approach, attributed to Jerome Bruner, in which concepts are repeated with additional complexity each time [Brun60].

Learning theories that emphasize the importance of social and cultural context to the learning process are considered to be related to constructivism as well. Lev Vygotsky's sociocultural learning theory, which emphasizes the importance of social interaction in the learning process [Hass11] is an example of this. Vygotsky postulated that the cultural context in which the learning occurs impacts the tools and techniques that are made available to learner. He introduced the concept of the *Zone of Proximal Development* (ZPD), which is the difference between what a learner is able to do without assistance compared to what they can accomplish with the guidance of a mentor [Hass11]. This is also a type of situated learning, in which learning occurs within a community of practice [JaSh19].

The constructivist paradigm parallels the return to robotics and rise of agent-based AI in the 1980s and 1990s, with a focus on constructing knowledge and recognition of situated learning.

2.3.4 Connectivism and Other Learning Theories

Beyond the three major learning theories of behaviourism, cognitivism, constructivism, and their offshoots, there are other paradigms and frameworks with which to approach discussions on learning. For example, models such as Bloom's Taxonomy [Aira01], Fink's

theory of significant learning [Fink03], Kolb's experiential learning cycle [Kolb84], and Gardner's Theory of Multiple Intelligences [Gard83], among others [Pete69], can provide common language to discuss learning theories and inform organizations such as the *Canadian Engineering Accreditation Board* (CEAB) looking to create a taxonomy for understanding learning outcomes.

Of particular note is connectivism, which is a learning theory directly inspired by technology. Connectivism recognizes that the connections a learner is able to make are directly influenced by their personal learning networks consisting of their access to information via Internet technologies [Duna11]. Learning theories continue to evolve over time as previous theories are tested and new technology leads to new insights. In many ways, the fields of study of human learning and AI and ML have symbiotic relationships. As such, ML is used to study education. This is discussed in the next section.

2.4 Machine Learning for Studying Education

The fields of study of human learning and ML have a symbiotic relationship. Many ML algorithms and methods are, and continue to be, inspired by the learning theories that have emerged from research in biological learning. In turn, the study of learning processes in machines can provide insights into biological learning as well.

Machine learning algorithms can advance the study of education and our theoretical understanding of learning in two main ways. One way is that these tools can be used to model learning behaviours, such as simulating autonomous agents and observing how they interact and change. This allows for learning theories to be tested without the need for biological subjects. Agent-based models are used in many areas, including in education. An example is the NetLogo programming environment, which is able to simulate complex systems comprised of individual agents that interact over time with various parameters

and conditions [WiRa15]. NetLogo is also an educational tool to teach students about agent-based models. Second, ML tools can also be used to analyse real student data that is collected using specialized tracking education software, *learning management system* (LMS) databases, or other software that is integrated into the classroom.

In the following sub-sections the application of ML to educational data is explored. First, the proliferation of digital education data is examined as a result of advancements in computer data collection and storage capabilities, increased access to educational technology, and growing numbers of online or remote delivery courses. Second, applications of ML with education data are discussed, including student modelling and prediction, decision support systems, and adaptive learning environments. Third, the particular application of ML for student outcome prediction as seen in the literature is presented. Finally, research into student course outcome prediction using LMS interaction data is analysed to identify the gap in the research that this doctoral thesis explores.

2.4.1 Digital Education Data

Traditional research approaches in the field of education rely on labour and time intensive techniques such as interviews, classroom observations, and surveys [FiBR15] [RoVP11][SaFD18]. However, the adoption and expansion of technology, particularly technology in classrooms, offers new avenues for educational research through the collection, storage, and use of digital education data. The proliferation of data-collecting technology has created new data repositories that offer novel opportunities for ML to advance the field of education [Bake14]. There are multiple groups that have started to bring ML techniques to focus on a variety of educational research areas and goals [Bake14] [BaZE17] [Kins15] [Pena14].

Using computers in education is not, in and of itself, new. Early behaviourist re-

searcher B.F. Skinner experimented with a form of *computer-assisted learning* (CAL) in 1954, prior even to the Dartmouth Summer Research Project on Artificial Intelligence of 1956 [Bate19]. Skinner's teaching machines used behaviourist principles to guide learning through providing structured information, testing, and immediate feedback to students. As other educational learning theories gained favour in the following decades, CAL fell out of fashion as it was less effective for higher level learning. However, similar systems under the term *computer-based training* (CBT) are still developed and used today for situations where lower cognitive domain levels are deemed sufficient, such as training for a specific task in the workplace.

By the 1980s, educational environments primarily used computers for CBT or to facilitate communication either between students and educators, or students and their peers [Bate19]. Starr Roxanne Hiltz and Murray Turoff pioneered the field of *computer mediated communication* (CMC) and did early work with online discussion forums in the 1970s [HiTu93]. However, the scope and reach of this type of research was limited until there was greater connectivity between users through the invention and widespread adoption of the World Wide Web in the 1990s. As technology continues to evolve, quantity and detail of educational data that can be collected has increased along with the number of ML techniques that can be used to analyse that data. In 2005, the term *Educational Data Mining* (EDM) was introduced to describe using data mining and ML techniques on data collected from higher level education settings in computer science [BaZE17].

Today, research at the intersection of AI and ML with education is still considered an emerging field [HeHT19] [HeIP18] [KoDM15]. Due to the field's relative infancy, the collection of digital educational data used for research purposes is not standardized [RoVP11]. Some educational data exists in repositories such as the DataShop@CMU [Data21], IEEE DataPort, Google Research, while other studies are conducted on a case-by-case basis.

Much of the work being done uses sensitive personal information, such as course enrolment and past grades [JiPa20] or relies on specialized software [HuZZ19][MaMP20]. Of note, the *computer-aided personalized system of instruction* (CAPSI) program developed at the University of Manitoba is an example of an innovative personalized system of instruction, still in use in specific undergraduate courses [Kins88][Kins90][Pear88].

As well, multiple studies have been performed on data collected from courses offered as a massive open online course (*massive open online course* (MOOC)) [Anto09]. The student experience in a MOOC is different than that of a typical lecture-style class, in that these courses are often asynchronous or may be taken not-for-credit. Thus, the type of learner that chooses to enrol in a MOOC can have very different expectations and engagement compared to students who enrol in a for-credit course. While there is an increasing number of these alternate options available for continued learning, the traditional undergraduate degree from brick-and-mortar universities is still the first choice for many students, particularly those coming directly from a high school program.

Distance and online undergraduate courses at universities have also been available for many years, usually asynchronously, for students who self-select to enrol. These distance courses are typically an option offered concurrently with a traditional in-person, synchronous class section. The move to fully remote, online learning in 2020 due to the COVID-19 pandemic has presented a new opportunity to study how students learn in a synchronous class. In March 2020, partway through the winter academic term, most classes were abruptly moved completely online with little warning or preparation. Subsequent terms at many undergraduate institutions continued to deliver courses that were prepared and run in a remote learning environment. These remote learning courses were offered using an LMS that acted as a portal for most, if not all, student activities such as obtaining course content, submitting work to be graded, writing timed exams, and potentially interacting with classmates. An LMS is “a centralized web based information systems where the learn-

ing content is managed and learning activities are organized. LMS represents a more general term for a technology framework that supports all aspects of formal and informal learning processes, including learning management, content management, course management, etc." [Wang14] Since remote courses became the default rather than the exception during the COVID-19 pandemic, the data collected by the LMS during these courses presents an unprecedented opportunity to study how students interact with course materials on their personal learning journey.

2.4.2 Overview of ML Applications in Education

Just as the amount and availability of education-related data has increased with advancements in computing technology, so too has the breadth of topics that are studied. Today, there are multiple areas of research and groups interested in bringing together computers and education through digital education data and ML.

Across all computer-related education research there are four main stakeholder groups: students, educators, administrators, and researchers [BaZE17]. Applications of ML in education each target one or more of these groups. Traditionally work in this area has been cross-disciplinary, with input from researchers in education, computing sciences, engineering, and psychology [Stem91]. Publications such as IEEE Frontiers in Education (FIE) and ACM Transactions on Computing Education (TOCE) bring together researchers across multiple disciplines. Over time, new inter-disciplinary fields have also emerged as well, such as *Engineering Education Research* (EER), Computer Science Education, and EDM, with their own priorities, publications, and taxonomies [SFMB21].

The applications of computer-related education research can be classified based on their objectives into three broad categories: student modelling, decision support systems, and adaptive systems [BaZE17] [Pena14].

In the first objective, **student modelling**, student models are created for prediction or structure discovery. Prediction may come in the form of predicting overall performance, specific characteristics, or undesirable behaviours. The topic of ML prediction in education is explored in greater detail in Section 2.4.3. In structure discovery, ML techniques are used to create groups or profiles of students to find groups of similar students, for example. An alternate use of structure discovery is to form teams of students with complementary characteristics, which is also a form of social network analysis [BaZE17]. Applications of student modelling typically indirectly benefit students and educators by uncovering new patterns that can be applied to decision support systems.

The second objective of ML in education, **decision support systems**, directly benefits students and educators through the development of tools that can lead to more informed decision-making. These tools may be supported by research into how to present information in new ways or at particularly relevant times. For example, these tools may include reports, data visualization, alert systems, recommender systems, scheduling support, or software to develop course content. These tools can provide feedback to educators and help students to monitor their progress, and may rely on predictive results from student modelling research in order to provide that feedback.

Lastly, **adaptive systems** in education refer to creating a learning environment that is tailored to the needs of the individual learner. This is also known as the "personalization of learning". An example of this is the concept of *cognitive digital twins* (CDTs) [Kins19]. Digital twins have existed in engineering industries for years as "digital replicas of specific electromechanical systems such as cars and airplanes to analyse and predict their behaviours. Although there is much work to be done to reach a true digital replica of a student, advances in understanding how students learn could lead to the development of CDTs creating much more personalised learning environments in the future (for a discussion on CDTs, see Appendix A).

Additional applications of ML in education include work into developing and testing educational learning theories or methods of evaluation. To achieve these objectives a variety of ML algorithms are used, such as regression, clustering, association rule mining, social network analysis, and text mining [RoVe10].

2.4.3 Prediction as an Application of ML in Education

One of the main applications of applying machine learning to education data is in the realm of student modelling for the purposes of prediction. Three main sub-areas of research within this field were identified in a review of the literature: studies related to grade prediction, dropout prediction, and predictions of student outcomes in learning activities. (For details on how the literature review was conducted, see Appendix B.)

The first of the main sub-areas of research is the use of ML techniques for **grade prediction**. ML techniques are used to predict student outcomes within a single course, where student outcomes may be defined as a final grade percentage, a final letter grade, a final grade level (ex. 'high', 'middle', or 'low'), or as passing vs. failing the course [MBKK03]. Alternatively, there has been interest in predicting student outcomes over multiple courses, such as a *cumulative grade point average* (CGPA) [MuAM22] or *grade point average* (GPA) upon graduation [Ogor07].

The second main sub-area in the field of ML and education prediction relates to student **dropout prediction**. In this sub-area, ML techniques have been studied for their ability to predict the likelihood of a student dropping a course [MbMG22] [TeRN19], graduating on time [PaJO17], or completing a certification or program [SiSL20].

ML has also been used to predict **student outcomes in learning activities** that occur over a shorter span of time than a course or term. For example, studies have investigated the suitability of ML to predict a student's mastery of a singular skill [LoBe21] or ability to

independently complete a particular task such as a programming exercise [MoGB21].

At this time, research is still being conducted to determine both the type of data and the features derived from that data that may have predictive value. Overall, the field consists of heterogeneous studies with mixed results and no singular set of metrics to compare the accuracy or successfulness of the predictions. In the following sub-sections, attribute properties and considerations of data sets used in student outcome prediction are briefly described, considered in terms of the types of students studied, methods of course delivery, and types of data collected. Then studies that use LMS time intervals as a predictive feature are discussed. Finally, some considerations on the current limitations of educational data for student outcome prediction are presented.

2.4.3.1 Attribute of Education Data: Types of Students

Research in student outcome prediction has been conducted using data collected from different cohorts of students, such as elementary and high school, or undergraduate and graduate students at universities [Spit21] [LoBe21] [LiCh20] [TiLW20] [SoOk20] [JiNT22].

The most common cohort of students studied in experiments of ML-based prediction are students studying undergraduate-level material. This may be attributed to the wealth of available data at this level, particularly data from more recent advances in educational software to support course delivery.

2.4.3.2 Attribute of Education Data: Methods of Course Delivery

The use of ML for student outcome prediction has been investigated in LMS in a variety of course deliveries, including traditional face-to-face classrooms [HaBH17] [RGPO21], blended and hybrid instruction [VaBD20] [MaSG17] [OrVa20] [PaML19], and flipped

classrooms [Wang21], as well as completely online courses that are delivered synchronously or asynchronously [YuWu21] [YuPS19] [KuLB15]. The method of course delivery influences how much the students are intended to interface with the LMS, and the course delivery method is therefore typically identified and considered separately in the research. The *massive open online course* (MOOC) is a type of course delivery method that has received considerable research attention related to student outcome prediction due to the amount of open access data available for researchers [Anto09] [BoSK16] [SiCa15] [MbMG22] [QuLW19] [CrAA18] [MuAC21]. However, the relationship between student behaviours and outcomes in courses delivered through tuition-based, degree-granting campuses such as universities is not considered the same as the relationship between student behaviours and outcomes in a freely-available MOOC. This is because of the different incentives and levels of investment in each type of course. It has been posited that it is more valuable to be able to understand and predict the former relationship given the additional costs of tuition-based programs on stakeholders and the importance of successful outcomes in this environment [DeDS22].

2.4.3.3 Attribute of Education Data: Types of Data Collected

Data collected from an LMS in a tuition-based course or a MOOC generally involves timestamp information of students' interactions with the system. However, the level of granularity in terms of which activities within the LMS are recorded as timestamped interactions may differ based on the particular software implementation. Information collected by some LMS implementations that has been used for student outcome prediction includes lecture video interactions [ZhUD22] [KuLB15] [MuAC21], general or specific time-on-task measurements [OrVa20] [TeRN19] [KuGI11], and student-to-student or student-to-teacher interactions on discussion boards or messaging [TiLW20] [Bail20] [ThPA13] [Kim14].

Research into student outcome prediction typically uses these timestamped datasets of activity logs in novel ways to create new features, or the timestamped data sets are supplemented with additional data. Examples of using the timestamped activity data to create new features related to the intervals between LMS interactions are described in further detail in Section 2.4.3.4. Supplemental data from outside the LMS may also be included as features for student outcome prediction, such as demographic information (including one or more of age, gender, and ethnicity) [HeLL18] [ArPi12] [MaCC21] [KuGI11] [Kots12] [MaUW21]; past academic performance in the form of course enrolment or current GPA [ArPi12] [MaME18] [JiPa20]; or student self-assessments in the form of surveys [LiFJ19].

Alternatively, some prediction approaches use data collected from specialized software or devices. For example, research has been conducted into how biological indicators such as facial cues, heart rate, eye movement, or brain activity can predict student outcomes [PeON20] [LaNL22]. Less intrusively, specialized software to capture additional details such as navigation or mouse clicks has also been investigated [HuZZ19][MaMP20] [TeBP19]. These approaches are less common as they require significant resources that are not generally available, which give the results a limited scope of applicability.

2.4.3.4 Research focused on: Time Intervals as a Feature for Student Outcome Prediction

In studies of undergraduate-level university course outcome prediction that use LMS activity logs, with or without supplementary data, there are certain features that are typically created from the timestamp data. These common features include the number of logins, number or percentage of items accessed over a defined time period (day, week, month, or weekday vs. weekend), and time spent on an activity or on the course overall [OrVa20]

[MaDa10]. The time in between the interactions, however, was found to be rarely considered. In one study, the number of inactive days (as a continuous stretch, and overall) was calculated [BrPH22]. In another, the intervals between LMS interactions were calculated, in a course that was delivered through in-person lectures and labs with LMS support for content management [DeBr20]. In the literature, the use of intervals as a measure to explore undergraduate student behaviour with an LMS in an online, undergraduate synchronous course over time or as a feature to predict student outcomes using only standard LMS data has not been explored.

2.4.3.5 Data Limitations and Bias Considerations

While previous work has claimed prediction accuracy results of up to 99%, direct comparisons between studies are challenging due to differences in what is considered a 'successful' prediction [DeDS22] [Zach18]. Instead, rather than comparisons between prediction models, this relatively young field is still focused on finding data and features that offer any predictive value. Features such as past academic performance and demographic information have been shown to be correlated with future academic performance, but there has been less success so far in predicting course outcomes using features that rely solely on "low-level student actions" in the form of timestamped interaction data [YuPS19]. Furthermore, the additional data discussed above is not always available to instructors. Mouse-click and eye tracking require specialized software and devices for data collection. As well, not all LMS implementations collect nuanced discussion board data or time-on-task measurements, making features from this information less generalizable. Student demographic or academic history information may not be accessible to the instructor, be incomplete, or may include reporting biases. Even when available, demographic or academic history data may unnecessarily bias predictive models by basing predictions on information that is not representative of a student's current behaviour in a course. As such, there is value in continuing to seek

predictive features that do not rely on that type of personal information [DeDS22].

Due to these considerations, there is interest in predicting student outcomes predominately or exclusively using broadly available course behavioural data, such as data collected through an LMS. Section 2.4.4 provides further detail on studies in student course outcome prediction using LMS data.

2.4.4 Student Course Outcome Prediction Using ML with LMS Data

As described in Section 2.4.3, within the vast landscape of ML for prediction in educational contexts, one sub-area of interest is course outcome prediction. This sub-area uses ML techniques to predict how students will perform in a single course, as measured by their final grade, and has been gaining traction in recent years (see Appendix B for information on the literature search). As discussed, the predictive models may use student demographic data (such as gender or race), administrative data (such as program enrolment, past academic performance, or financial aid received), current course behaviour data (such as LMS interactions, interim course grades, discussion board activity, or mouse click activity), student self-assessments (such as surveys) or a combination of these features.

Of these data types, course behaviour data collected through an LMS provides the most opportunities to be studied in large quantities. This type of data does not require software beyond what is commonly available at many educational institutions, nor does it contain sensitive personal information like demographic or administrative data which may require additional permissions and introduce biases as discussed in Section 2.4.3.

In the following subsub-sections, six representative studies on ML for course outcome prediction using LMS data have been chosen for deeper analysis based on their high reported accuracies and types of LMS features used. The studies are described and evaluated for their strengths and limitations. Through these analyses, the novelty of this doctoral research

study will be established.

2.4.4.1 Review of course outcome prediction models (Arizmendi et al.)

In a 2022 paper, Arizmendi et al. reviewed 82 course outcome prediction models across 39 papers [ArBR22]. Of these, 29 prediction models used only current course behaviour data, which the authors refer to as "behavioural predictors." A variety of ML prediction algorithms were used and, of the 29 prediction models, twelve were decision trees [BaLL15] [BeCU20] [KoTF13] [RoEZ10] [Zach18], ten were a type of regression [BaLL15] [BeCU20] [Will19] [You16] [YuLF20] [Zach15], four were a Naive Bayes algorithm [BaLL15] [BeCU20], and the remaining three were a W-K algorithm [BaLL15], rule induction [RoEZ10], and genetic programming [XiGP15]. The size of the data sets (i.e. number of students) and method of course delivery varied across the studies, making it difficult to directly compare prediction accuracies. However, of interest is the high reported accuracies in some of these studies. For example, the highest prediction accuracy within this group of studies was 99% reported by Zacharis, using a decision tree prediction algorithm [Zach18]. Three studies reported prediction accuracies of 80-82% from decision trees [KoTF13], logistic regression [Zach15], and genetic programming [XiGP15]. The remaining 26 prediction models reported more modest course outcome prediction accuracies in the range of 55-65%. These results indicate that there is potential for LMS data to support student outcome predictions, and that there is still room to improve the predictive models.

2.4.4.2 High Prediction Accuracy with LMS Data (Zacharis)

As introduced above, Zacharis experimented with decision trees and LMS data to great effect, producing models with 82% and in 99% accuracy in 2015 and 2018, respec-

tively [Zach15] [Zach18]. Both studies were conducted using data sets from undergraduate-level blended courses, which consisted of both in-person lectures and labs as well as online interactions, offered at a large technological university. In the first study the data set contained LMS records from 134 students, while the second study consisted of records from 352 students. In both cases, the LMS data included the number of files viewed, quiz attempts, student contributions to class content such as a wiki page, and volume of communication through discussion boards and direct messages. Zacharis concluded that the amount that a student made use of the available communication options was the most predictive factor in a blended environment. In the 2013 study, the number of emails, chats, and messages read or posted to an online class forum accounted for 37.6% of the variation in student's final grades. Later, in the 2018 study, Zacharis codified this relationship into two predictive rules based on the number of text messages sent from a student to their instructor or classmates [Zach18]:

> "(a) IF messages exchanged $\leq$ 172 THEN student fails, and
> (b) IF messages exchanged $>$ 172 AND content creation contributions $\geq$ 13 THEN student succeeds."

Although Zacharis achieved significant success with these two rules, and decision trees in general, there is no indication that the results would be replicable in other courses. Blended courses offer a different experience than fully online courses, with required in-person components that provide structure and opportunities to meet and interact with instructors, teaching assistants, and peers in the real world. The heavy reliance on student communication for prediction in these studies suggests that the particular courses studied were both well-suited for and encouraged this type of behaviour. For example, not all LMS courses include a course-specific messaging function, discussion board or wiki, and those that do are not equally monitored by the teaching staff, which may impact student engagement. The potential power of ML was not harnessed, as only one ML was implemented and it

could be replaced by a set of conditional expressions. There is no indication in the Zacharis studies that 172 is an important number of messages that can predict student success in other courses, nor that there exists a specific number of messages that can be discovered for additional courses and used to predict student success therein. Zacharis postulates that the course design, which encouraged collaboration, may have influenced the relative predictive value of the features [Zach18]. Other features which are commonly found in other course outcome prediction studies, such as total time online, number of LMS interactions, and number of logins, were found to only weakly correlate with student outcomes in these studies. Therefore, there is room to develop new features from LMS data that may better correlate with student outcomes on individually or as combinations of features.

2.4.4.3 Prediction with Multiple Features (Macfadyen and Dawson)

In an earlier study, Macfadyen and Dawson also used LMS features including mail messages and discussion board posts, as well as time on task features. They aimed to predict students who failed and those who were at risk of failing (defined as a final grade of $< 60\%$), using a data set of 118 students enrolled in a fully online undergraduate university course in 2008. With regression modelling they were able to accurately predict 80.9% of the students who failed, and 70.3% of students in the "at risk" category. In this study, just as in the Zacharis studies, the most predictive feature was related to the social aspect of learning - in this case, student contributions to discussion boards. Therefore, Macfadyen and Dawson indicate that student communication data has predictive value, however the study also shows that simple conditional relationships using this data alone can not always account for course outcomes.

2.4.4.4 Prediction with Limited LMS Interaction Data (Orji and Vassileva)

Depending on the online learning environment, or how the course is structured, student communication data is not always available. For example, in another study, Orji and Vassileva explored the relationship between LMS data and student course outcomes in a blended, undergraduate level university courses [OrVa20]. The data set consisted of records from 490 students within an LMS (which is referred to as a *technology-enhanced learning system* (TELS) in the paper). Unlike in the previously mentioned studies, the data set used by Orji and Vassileva did not include student communication data. The features included in the model were total time spent on a task type (homework, assignments, quizzes, or readings), number of logins, percentage of activities accessed, and average assessment score. Using a random forest model with an 80%-20% train-test split, Orji and Vassileva predicted student course outcomes with 84.1% accuracy. However, the "AveAssessmentScore" feature was the important factor to the prediction, contributing 60% to the prediction outcome. This feature is the average score a student has achieved on all marked assessments in the LMS, thereby indicating that the unweighted average of course marks is highly correlated with a student's final grade, as is to be expected.

2.4.4.5 LMS Data for Early Prediction (Brdnik et al.)

Some studies attempt to predict student outcomes using only a subset of the LMS data for early prediction. For example, Brdnik et al. attempted to predict student outcomes at each month during an undergraduate-level, online course from the UK-based Open University which offers low-cost distance course options that are similar to a MOOC [BrPH22] [KuHZ17]. The course ran from October 2013 to May 2014, and students could obtain grades between 0 and 100 with 40 set as the threshold to pass. In order to facilitate early prediction, the LMS data collected for 777 students was augmented with demographic and

administrative data. The study had access to demographic data including gender and age, as well as past student performance such as whether the student had taken the course before, and their current education level. The data set also contained the number of clicks per day in the LMS (which is referred to as a *virtual learning environment* (VLE) in the paper). Using linear regression, Brdnik et al. were able to accurately identify 74% of students who would fail the course one month before the final exam. However, it is notable that in this study, the LMS data is augmented with demographic or administrative data that is not always available. As well, the data set was obtained from a MOOC-type course, rather than a degree-granting university, which may impact student motivation and behaviour. In fact, due to the lower levels of time and financial investment in these types of courses, it has been speculated that it is more valuable to study course outcome predictions from degree-granting universities because of the relatively higher costs of their tuition-based programs [DeDS22].

2.4.4.6 Comparison of Behavioural Data, Administrative Data, or Both for Prediction (Bird et al.)

Bird et al. studied the predictive strength of administrative data relative to LMS data [BiCS22]. Using a large data set, their study addressed criticisms that earlier work did not include sufficient numbers of students. The data sets used by Bird et al. included 226, 784 students across 2646 courses in 23 community college institutions in the Virginia Community College System (VCCS). Each student record consisted of data from 2000 onward, and LMS data from 2019 onward, omitting Spring 2020 to account for a different grading scheme used during the initial COVID-19 pandemic shutdown. The administrative records included "program of study, courses taken, grades earned, credits accumulated, financial aid received, and degrees or certificates awarded", and the LMS data included content page accesses, discussion board posts, messages, and assignment and quiz submissions. This study

compared the predictive strength of the administrative-only data, LMS-only data, and the combined administrative and LMS data when using a random forest prediction model. Bird et al. found that the administrative data had the greatest predictive value, for students not in their first term. The LMS data on its own was the least predictive (c-statistic of 0.779) and combining it with the administrative data resulted in only marginal improvements from the administrative data-only model. A drawback to this study was the use of data from a wide variety of courses across multiple disciplines and delivery modes. The authors indicate that there may be differences in prediction performance based on the course itself, citing the example that instructors in an English course may set up their course on an LMS differently compared to a Mathematics instructor. Therefore, there is value in focusing on specific courses or course types when performing course outcome prediction studies, even at the expense of larger data sets, in order to reduce confounding factors.

2.4.4.7 Summary of Previous Studies

Current course outcome prediction studies often make use of demographic or administrative data that is not available in all cases. Therefore, there is interest in exploring the predictive value of LMS-only data, as shown in the review paper by Arizmendi et al. which contains multiple LMS-only predictive models. Although Bird et al. indicate that LMS-only data may be less predictive than other student information, Zacharis shows that by narrowing the scope to homogenous course data it may be possible to achieve strong prediction results using behavioural data alone depending on the features derived from the LMS. As well, different ML models may be better suited to processing the type of data recorded by LMS software. As observed in Arizmendi et al., current course outcome prediction studies tend to employ only one ML predictive model, most commonly either regression models or decision trees. This work addresses the gap in the literature of LMS-based course outcome prediction through two main avenues. The first way is through the development

of novel LMS-based features that encapsulate a student's behaviour over time, including intervals and burstiness features. The second way is the application of additional types of ML models to generate predictions from LMS-only data.

2.4.5 Contributions to the Field

This work addresses the gap in the literature of LMS-based course outcome prediction through in three main ways. The first is through the development of novel LMS-based features that encapsulate a students behaviour over time, including intervals and burstiness features. The second is the application of additional types of ML models to generate predictions from LMS-only data. The third is the use of intervals as a measure to explore undergraduate student behaviour with an LMS in an online, undergraduate synchronous course over time or as a feature to predict student outcomes using only standard LMS data, a context that has not been explored in the literature. These are the contributions this research makes to the field.

2.5 Literature Review Summary

This chapter introduced the evolution of AI and ML over time. The three main learning theories of behaviourism, cognitivism, and constructivism were presented, with parallels drawn to advancements within AI. The current trends and some future directions for using AI and ML for research on education and personalised learning were discussed, and the proliferation of digital education data, increased access to educational technology, and growing numbers of online or remote delivery courses. Applications of ML using education data are discussed, including student modelling and prediction, decision support systems, and adaptive learning environments. In particular, the application of ML to education data

for the purpose of student outcome prediction was explored, including course or cumulative grade prediction and dropout prediction. Literature on the application of ML using different educational data sets for the purpose of student course outcome prediction was examined. Then research on student course outcome prediction using LMS interaction data was analysed and synthesized, supporting the contributions this doctoral thesis makes to the field.

Chapter 3

Methodology

In this chapter, the experiments that were designed to address the research questions in Section 1.2.3 are presented. The processes of acquiring the LMS data available from the UM Learn Data Hub, processing the raw data into course and student models, creating temporal features from the models to represent students' interactions with the LMS, and the design of time series classification experiments are described. The temporal features are used in the first three groups of research questions to perform an exploratory analysis of students' patterns of behaviours within the LMS; discover correlations between the temporal features and between the features and course outcomes; and evaluate the use of the features for early prediction of course outcomes. The time series classification experiments address the fourth group of research questions.

3.1 UM Learn Data Hub

This work focuses on a data-driven approach, wherein the data is analysed to create models and draw conclusions. This differs from a model-driven approach in which models

and theories are postulated first by assuming certain characteristics and statistical properties of the data in advance, and then tested for fit with the available data.

The data-driven approach is made possible by having sufficient data available to create features for machine learning. The following section describes the types of educational data collected, and the models created from that data.

This work uses the quantitative information that is already collected and stored in an LMS to model and create new features and predictions, without requiring any additional data collection.

The University of Manitoba uses an LMS called UM Learn that is an institution-specific implementation of the Brightspace product from Desire2Learn. The UM Learn platform records time-stamped user activity, some of which students and instructors are able to access through course-specific webpages on UM Learn.

A companion product called the Data Hub stores the full set of user activity data, and is managed by The Centre for the Advancement of Teaching and Learning ("The Centre") at the University of Manitoba. Although information in the Data Hub is linked to specific student profiles, any such linking information is removed by The Centre. No identifying information, such as personal or demographic data, is included in the dataset. That is, the user activity data was used to create a profile of an anonymous learner. This is different from some of the previous work in this area, which uses past course enrolment and grades [JiPa20]. In this work, only the anonymous data that is available through the Data Hub is used to create the student timelines and predict course outcomes in order to examine how student behaviour within the LMS relates to course outcomes without bias from previous course performance. As well, this dataset mimics the information that could be made available to an instructor teaching the course without requiring data collection beyond what is gathered through the LMS.

3.1.1 General UM Learn Data Sets

Conversations with The Centre have shown that research on information stored by the UM Learn Data Hub has rarely been conducted [Nikn17] [Ronc19].

Personal and demographic information is not collected in the Data Hub and identifying student information, such as student numbers, is removed from the data and replaced by an anonymous user ID automatically at The Centre. Both The Centre and the Research Ethics Board advised that Research Ethics Board approval was not required in order to access and use the data stored in the Data Hub.

For each course on UM Learn, the Data Hub records information about the course setup and student interactions with the course components. For each student enrolled in the course, the Data Hub datasets include:

- An anonymised user ID;
- Most recent access date of the UM Learn course page;
- Date of each assignment submission (if submitted);
- Date of when assignment feedback was most recently read (if at all);
- Most recent access date for each course content item (ex. course slides, assignment instructions, lab instructions, example code);
- Total number of times that the content item was visited;
- Date and time (to the second) that the student started each quiz;
- Date and time (to the second) that the student completed each quiz;
- Quiz score; and
- Final grade as a percentage.

Taken together, these features were used to create models of student learning behaviour within the course LMS.

The raw data from the LMS is stored in a series of tables for each course, each of which can be opened individually as an Excel spreadsheet. The Brightspace software allows for a number of different data sets [D2L22], although The Centre declined to disclose the total number of data sets implemented in the UM Learn Data Hub. The four data sets provided by The Centre and used in this work are: Content User Progress (i.e. content page accesses), Grade Data, Quiz Data, and Assignment Data. All four data set tables were acquired for each of the courses described in Section 3.1.2, for a total of twelve raw data set tables to be processed as described in Section 3.2. Each data set table was stored in a separate Microsoft Excel file by The Centre.

Tables 3.1-3.4 show examples of the raw data in each of the four types of data tables.

Table 3.1: Example of the header and two rows of the Grade data table. The UserId field has been modified with dummy values for clarity.

Section	Calculated Final Grade
COMP-1010-A01 - Introductory Computer Science 1	38.952074152
COMP-1010-A01 - Introductory Computer Science 1	85.462796942

Calculated Final Grade Denominator	Calculated Final Grade %	Users.UserId
100	38.952074152	111111
100	85.462796942	123123

Table 3.2: Example of the header and two rows of the Content User Progress data table. The UserId and Title fields have been modified with dummy values for clarity.

ContentObjectId	UserId	CompletedDate	LastVisited	IsRead
1951766	111111	None	2020-07-22 22:02:42	True
1958861	111111	None	2020-07-30 23:07:59	True

NumRealVisits	NumFakeVisits	TotalTime	IsVisited
1	0	1	1
1	0	1	1

IsCurrentBookmark	IsSelfAssessComplete	LastModified
1	0	2020-07-22 22:04:12
1	0	2020-07-30 23:09:08

contentobject4courses.OrgUnitId	contentobject4courses.Title
375277	Lecture1
375277	Lecture2

Table 3.3: Example of the header and two rows of the Quiz data table. The UserId field has been modified with dummy values for clarity.

AttemptId	QuizId	UserId	OrgUnitId	AttemptNumber
2096020	135883	610116	111111	1
2095456	135883	592109	123123	1

TimeStarted	TimeCompleted	Score	IsGraded
2020-07-31 20:59:55.250000	2020-07-31 21:14:22.566000	11	True
2020-07-31 17:21:04.956000	2020-07-31 17:31:11.506000	11	True

OldAttemptNumber	IsDeleted	PossibleScore
	False	12
	False	12

IsRetakeIncorrectOnly	QuizObjects.QuizName
False	Quiz 9
False	Quiz 9

Table 3.4: Example of the header and two rows of the Assignment data table. The UserId field has been modified with dummy values for clarity.

DropboxId	OrgUnitId	SubmitterId	SubmitterType
99040	375277	111111	User
98421	375277	123123	User

FileSubmissionCount	TotalFileSize	FeedbackUserId	FeedbackIsRead
4	129607	441042	False
2	23305	357728	False

Score	IsGraded	LastSubmissionDate	Feedback
27.500000000	False	2020-08-17 08:31:55.653000	
26.500000000	False	2020-08-02 05:04:15.113000	

FeedbackLastModified	FeedbackReadDate	CompletionDate
2020-08-20 21:45:33.970000		2020-08-17 08:31:55.653000
2020-08-10 01:56:52.156000		2020-08-02 05:04:15.113000

AssignmentSummary.Name	AssignmentSummary.Category
Assignment 5	Assignments
Assignment 4	Assignments

AssignmentSummary.Type	AssignmentSummary.StartDate
Individual	2020-08-03 09:00:00
Individual	2020-07-20 09:00:00

AssignmentSummary.EndDate	AssignmentSummary.DueDate
2020-08-17 09:00:00	2020-08-15 00:59:00
2020-08-05 01:59:00	2020-08-03 00:59:00

As shown in the examples of Tables 3.1-3.4, each row represents information about a student interaction in the course and each column contains a field related to the type of data in the table. Each student interaction of the type specified in a table is in a separate row. That is, a student's information is not grouped into a single row but spread across many rows. For example, if a student accessed 100 content pages, these interactions will be represented as 100 distinct rows in the Content User Progress table for that course. Similarly, in the Assignment Data and Quiz Data tables each student has a separate row for each assignment and quiz submitted. In the Grade Data table, each student appears in only one row storing their final grade. In addition, not all of the fields are relevant to a particular course. For example, in the Content User Progress tables the "IsSelfAssessComplete" field is populated with all zeroes because self-assessments were not used in the course. Furthermore, not all fields in each table are populated. For example, in the same Content User Progress tables the "CompletedDate" field is empty in all rows.

3.1.2 COMP 1010 UM Learn Data Set

The UM Learn Data Hub datasets presented here for this work are for three iterations of COMP 1010: Introductory Computer Science 1 held over three separate terms. Although students can come into the course with a range of programming experiences, COMP 1010 is considered the first in a series of Computer Science courses. Prior knowledge of programming is not required, however Grade 12 mathematics is a pre-requisite. An overview of the three courses is shown in Table 3.5. The number of weeks refers to the length of time that the course LMS page was available to students, and the enrolment is the number of students who received a final grade in the course.

Table 3.5: COMP 1010 Data Set

	Term	Number of Weeks	Enrolment
Course A	Summer 2020	13	101
Course B	Fall 2020	18	124
Course C	Summer 2021	18	101

All of the courses in Table 3.5 were offered as remote, synchronous courses. These courses were chosen due to their consistency, repeatability, and availability. Sections of COMP 1010 typically have a large enrolment cap (over 100 students), and are offered multiple times per year. Due the number of sections per year of the course, there is a standardized structure to the course that has stabilized over time to facilitate a common presentation of the course material regardless of the current instructor. As well, there are many opportunities for students to interact with the LMS, through the content pages, quizzes, and assignments. The students who enrol in COMP 1010 are not a homogenous group. Since there are minimal pre-requisite requirements, the students can register with a wide range of both topic-specific programming knowledge and experience with a university LMS. These three particular sections of COMP 1010 were also chosen because they were delivered fully online during the COVID-19 lock downs, during which no alternate in-person course offerings were available. Therefore, any student who wanted to take the course during this time, needed to do so through with the remote delivery option through the LMS.

3.2 Creating Models from LMS Data

The LMS data was extracted from the UM Learn Data Hub into spreadsheets organized by course and data type. Each student is represented by unique user ID number that is unrelated to any personal information, such as an official student number. The LMS student **interaction** entries in the spreadsheets are associated with a user ID to preserve

anonymity. Two models were created to encapsulate the information about the courses and the students. The first was a model of a generic course structure, which was designed to provide a framework and context for the student LMS activity data relative to milestones in each course such as start and end of term, *Voluntary Withdrawal deadline* (VW deadline), and due dates. The second was a student model, which was designed to represent an individual learner's activity patterns and performance over the term.

3.2.1 Course Model

To describe the models, the term "course" will be used generically to refer to one scheduled section of one single undergraduate course that occurs over one term (Winter, Summer, or Fall term). There may be more than one section offered in a given term, although students typically do not have significant interactions with other sections. All students in the course have access to the same lecture materials, and have the same assessments and deadlines, other than exemptions or extensions for medical or compassionate reasons. For the purposes of analysing the data for patterns of behaviour, all students in a course were assumed to have the same deadlines. For example, handing in an assignment after the deadline was considered a late submission in the course model, even if the student was allowed to do so with an extension. This was done to reflect that a late submission, even with a instructor approval, is different behaviour than submitting the work by the original deadline.

Even though the set-up, lecture style, and assessment types can differ between courses, there are some commonalities underlying the general course offerings that were used to create a course model in Python. Courses each have a start and close date on UM Learn, which is the time period that the course is available to students, often starting before the first lecture and closing after the final exam at the end of the term. Each course has a

VW deadline, which is the last day that a student can withdraw from the course without penalty, and may have a mid-term break. These dates may impact a student's activity levels within the course. By creating the Python course model out of information that is common to many undergraduate courses, the tool can also be extended to model courses beyond COMP 1010 that follow a similar structure. A simplified depiction of the course model created in Python is shown in Figure 3.1.

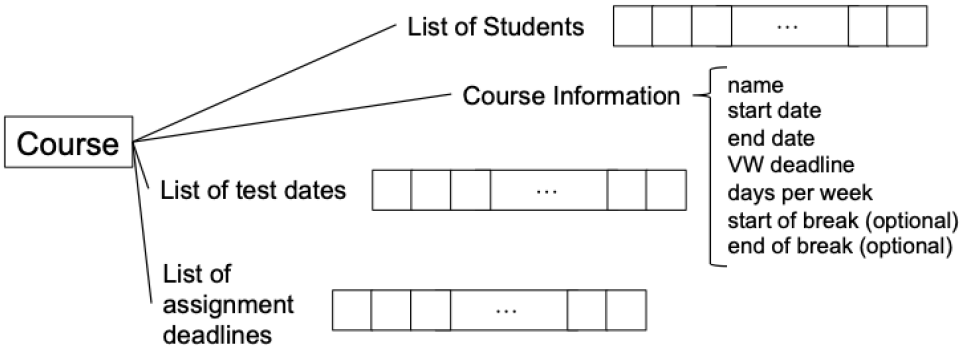

Fig. 3.1: Simplified visualization of the course model

In general, courses administered through an LMS have a series of assessments throughout the term. The types of assessments captured in the course model are assignments, labs, quizzes, midterm tests, and a final exam. The course model includes the number of each assessment type, and the deadline and weight of the assessment as a part of the overall course grade. Midterm tests and final exams are considered separately from the quiz assessment type, in order to distinguish between these longer tests from from shorter, less-weighty quizzes that may be administered more frequently. For each midterm test and exam, the model also includes timing information about an Honesty Declaration (if it was required), allowed writing time, and starting window. The model can be expanded to include additional assessment types that can be submitted through the LMS such as delineating different types of assignments, as required. Some courses may also have a component, such as class participation, that is measured outside of UM Learn. Course components for which the

grade is not available on the LMS are currently not included in the course model.

Finally, the course model includes the number of lecture days per week and the anonymised cohort of students associated with that course. Note that in this course model, the list of students is a list of student objects. Each student object is an instance of the student model, described in Section 3.2.2, storing information and features for one student.

3.2.2 Student Model

Students can interact with the course through UM Learn in a number of ways. The types of interactions between a student and the LMS that are used to create the student model are:

- UM Learn Assignments
- UM Learn Quizzes
- UM Learn Content page accesses

The student model pre-processes the raw LMS data for feature extraction by encapsulating the LMS interaction data and course grades into a Student object. Each student is a modelled in the same way, as an instance of the Student object, and a list of all Student objects associated with a particular course is included in the course model object described in Section 3.2.1.

As shown in Figure 3.2, every interaction that each student has with the LMS has data associated with it. The data associated with each type of interaction was encapsulated into custom Assignment, ContentPage, and Quiz objects. Any work completed independently and then submitted as one or more files by the student is considered a UM Learn Assignment. The type of work may fall under different assessment categories in the Course Syllabus. A UM Learn Assignment submitted by the student will have a name, a range of dates that the

assignment was allowed to be submitted, an actual submission date, a due date, a number of submissions, a submitted file size, and a grade associated with it. If the student's assignment is marked, it will also have a date that the feedback was read by the student.

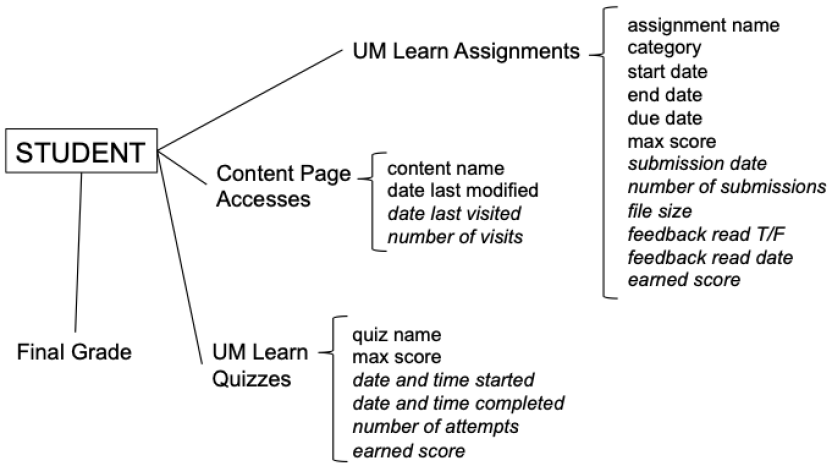

Fig. 3.2: LMS information associated with a student

Similarly, each quiz submitted by the student has a name, maximum score (or number of possible marks), a date and time that it was started, a date and time that it was completed, a number of attempts, and an earned score (the student's grade on the quiz).

Each content page accessed by the student has a name, a date that the page was last modified, a date that the page was last visited by the student, and a number of times the student visited the page.

In Figure 3.2 the data fields for each type of interaction are shown in either plaintext or italics. The data fields in plaintext are values that will be common across all students in a course, whereas those data fields in italics are specific to the individual student. For example, an assignment will have a common maximum score for all students, but each student will have an individual earned score on that assignment.

The student model includes all of the LMS interactions described in Figure 3.2 organized

into a timeline, as shown in Figure 3.3. A list of intervals is created from the amount of time in between successive interactions in the timeline. The timeline and intervals are discussed in further detail in Section 3.2.3. In addition, the student model also includes an anonymous user ID and a final grade associated with the student, which is a cumulative weighted grade of all coursework from the student for the term.

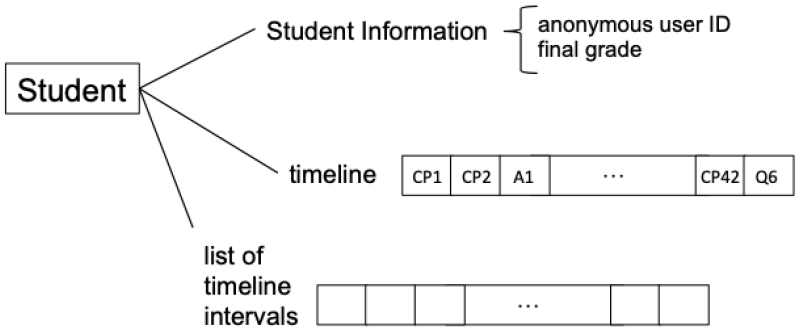

Fig. 3.3: Simplified visualization of the student model. The timeline consists of student interactions organized chronologically. The timeline includes content page accesses (CP), assignment submissions (A), and quiz submissions (Q). Each entry in the list of intervals is the length of time between successive timeline interactions, in chronological order.

3.2.3 Student Timelines

Within the student model, the LMS interactions are arranged into a chronological timeline. That is, for every student, each of their assignment submissions, quiz starts, and most recent content page accesses were assembled in order by date and time to create a timeline of activity within the structure of the course. Figure 3.4 illustrates this concept. The timeline for each student consists of a list of Assignment, ContentPage, and Quiz objects as described in Section 3.2.2, ordered chronologically by the date associated with each object.

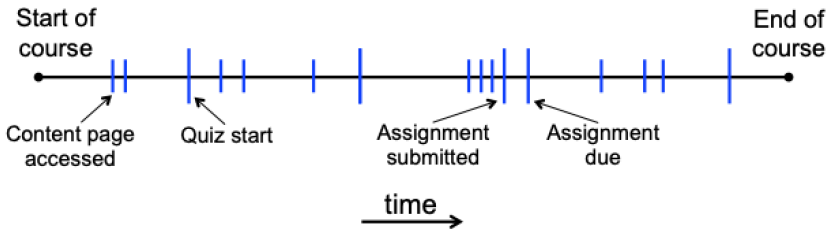

Fig. 3.4: Visualize a student timeline

An **interval**, also known as an **inter-event time**, is defined as the time between two successive timeline items for a student. For example, if a student accesses a content page containing lecture slides followed by a content page containing assignment instructions five minutes later, the interval between those interactions is five minutes. Intervals in LMS interactions have been studied previously for a synchronous, in-person undergraduate class, in which students still had face-to-face to contact with the instructor and offline activities [DeBr20]. The datasets obtained from Courses A to C in Table 3.5 were from fully remote, online, synchronous course offerings.

Each student can have a different number of items on their timeline, depending on how many of the course pages, assignments, and quizzes they interacted with on the LMS during the term. Different patterns of behaviour throughout the course are also possible, characterised by different patterns of intervals. For example, student timeline items may be evenly spaced, which would mean intervals of a consistent length. Student interactions may also occur in bursts, characterized by multiple short intervals followed by a longer interval, or in other patterns. Bursty patterns of activity have been shown to exist in some human interactions with technology, such as sending emails [Bara05]. Intervals are used to describe student interactions with the system temporally (i.e., via time events), beyond simply counting the number of interactions.

3.2.4 Student Model Features

The following numerical **features** were created based on the student timelines. Features 1-10 were calculated over the full timeline, as well as up to designated points in the timeline, whereas features 11-12 were calculated once. All of these features are discussed in further detail in sections 3.3 to 3.5.

1. Total number of student interactions (sum of the number of content page accesses, number of assignment submissions, and number of quiz submissions)
2. Total number of timeline items
3. Fraction of days, out of all course days, in which at least one LMS course interaction occurred (i.e. non-zero days)
4. Total number of intervals
5. Mean interval lengths in seconds
6. Variance of interval lengths
7. Skewness of interval lengths
8. Kurtosis of interval lengths
9. Burstiness of intervals
10. Time between assignment submission and assignment deadline, on average
11. Fraction of Midterm time used
12. Final Grade (out of 100)

3.2.5 ML Learned Features

The features introduced in Section 3.2.4 were hand-crafted using the Student Models and timelines. In addition, features were automatically generated using ML. In particular, the Featuretools framework [Feat22] was used in Python to find relationships in the raw LMS data.

To create these automatically learned features, the raw data was first pre-processed. As described in Section 3.1.1, the original datasets were spread over multiple separate files. Each file contained data about a type of LMS interaction, and therefore information about a particular student was spread across the files. All of the data across the Content User Progress (i.e. content page accesses), Quiz Data, and Assignment Data files was organized into a single table, represented as a DataFrame in Python, called 'interactions'. Each row in the 'interaction' table represented a single student interaction with the LMS and had three columns: the anonymous student ID, an interaction type (Content Page, Quiz, or Assignment), a date (time, day, month, and year).

The data from multiple courses was used, as described in Section 3.1.2. Since each of the courses was held in a different term, they each had a different start date and were not directly comparable. To account for this, all of the dates were normalized to the course start date. This means that an LMS] interaction on the first day of Course A would appear in the table as the same day, month, and year as an interaction on the first day of Course B or Course C, and so on.

In addition to the 'interactions' table, a second table with just one column was created which contained all of the anonymous student IDs in the data set. This 'student' table was also represented as a DataFrame in Python. The Grade data was not used to create the ML learned features, as the final grade was target variable (or information to be predicted) so it could not be incorporated into the features that would be used to make the predictions. A

trial was conducted including the Grade data, to see if it influenced the output features. The result was that the automated feature engineering algorithm simply identified the grades as an additional feature.

Once the 'interactions' and 'student' tables were created, an entity set was created from them. Using *Deep Feature Synthesis* (DFS)[1] in Featuretools, a relationship between the 'interactions' table and 'student' table was machine-generated using the anonymous student ID to link the information in both tables.

The DFS algorithm produced eleven automatically generated features, which are listed below:

1. COUNT(interactions)
2. MODE(interactions.type)
3. NUM_UNIQUE(interactions.type)
4. MODE(interactions.DAY(time))
5. MODE(interactions.MONTH(time))
6. MODE(interactions.WEEKDAY(time))
7. MODE(interactions.YEAR(time))
8. NUM_UNIQUE(interactions.DAY(time))
9. NUM_UNIQUE(interactions.MONTH(time))
10. NUM_UNIQUE(interactions.WEEKDAY(time))
11. NUM_UNIQUE(interactions.YEAR(time))

[1]In this case, DFS refers to Deep Feature Synthesis as a type of automated feature engineering that has been developed for temporal and relational data, and not Depth-First Search which is the more common acronym.

Of these eleven features, three were removed from the set. The feature MODE(interactions.type) was removed because it contained non-numeric data. This feature calculated the most common LMS interaction type for each student and had one of three values: 'ContentPage', 'Quiz', or 'Assignment'. Rather than using one-hot encoding to turn this non-numeric data into numeric data[2], the feature was simply removed because all but one student had the same value ('ContentPage') and therefore the predictive value of the feature in this data set was limited.

The two features related to the year, MODE(interactions.YEAR(time)) and NUM_UNIQUE(interactions.YEAR(time)), were also removed from the set. These features were removed because the courses were all less than one year in length meaning that all students had the same values for these features.

After removing the three features as described above, there were eight remaining features generated from the ML algorithm to be used in the predictive experiments described in more detail later on in this chapter.

3.3 Group 1 RQs: Patterns of Behaviours

In this section the first set of research questions and the experiments performed to address them are introduced. The following two sections similarly introduce the experiments for the other two groups of research questions.

The first group of research questions centres around student's patterns of behaviours. The overarching question is, what are students patterns of behaviours (as measured by time

[2]One-hot encoding is a method of converting non-numeric data into numeric data, particularly when the non-numeric data has a limited number of values. In this case, one-hot encoding would transform the feature of MODE(interactions.type) into three separate features, one for each interaction type of Content Page, Assignment, or Quiz. If that interaction type was the most common for a student, then its feature value would be 1, otherwise the feature value would be 0.

and interaction)? To address this question, an exploratory data analysis was conducted by creating features from the raw LMS data.

RQ 1.1: What are students' patterns of behaviour on an LMS over time (over time periods, within a single interaction/time, between interactions, time spent on assignments/quizzes, in relation to course events such as assignment deadlines and VW dates)

To begin to explore patterns in the data, it was necessary to create features from the raw LMS data. The first of these features was the number of interactions that each student had with the course page (see Section 3.2.4 #1). This was defined as the total number of assignment submissions, quiz submissions, and content page accesses. The second feature was the number of timeline items, where the timeline was defined as in 3.2.3 (see Section 3.2.4 #2).

Additionally, a feature was created out of students' interactions with the assignments (see Section 3.2.4 #10). In this feature, the amount of time between the assignment deadline and student's actual submission date was calculated. If there were multiple assignments, then the average of this difference was calculated to account for changes over the term while still maintaining a history of behaviour around assignments. The calculation of this feature is shown in equation Equation 3.3.1, in which the number of assignment is $n \geq 1$, the deadline of the i^{th} assignment is A_i, and the date of the student's submission of the i^{th} assignment is s_i.

$$\frac{1}{n}\sum_{i=1}^{n} A_i - s_i \qquad \text{(Equation 3.3.1)}$$

Each of the above three features were computed over the full timeline, as well as up to designated points in the timeline. These designated points were: one quarter of the way

through the term, half of the way through the term, three-quarters of the way through the term, the day after the VW deadline, the day after the midterm or first term test if there were multiple term tests, and the day after the final exam. The term was divided into quarters in order to observe the features at equally spaced points over time, with sufficient time in between the chosen points for students to have multiple interactions with the LMS. The additional designated points of the VW deadline, midterm (first term test), and final exam were chosen to examine their potential as trigger points for changes in student behaviour.

As well, an additional feature for each student was created out the amount of time that the student used before submitting their midterm (see Section 3.2.4 #11). This feature was calculated as the fraction of time the student used out of the total allowed writing time for the midterm (first term test). For example, if the length of time between when a student started and submitted their midterm was 45 minutes out of a maximum allowed writing time of 60 minutes, their value for this feature would be 0.75. The fraction of writing time was used as a feature, rather than the length of the writing time, in order to make the feature more flexible and be able to account for courses with midterms of any length.

RQ 1.2: What are students' patterns of behaviour on an LMS with regard to interactions and intervals?

Additional features were introduced to quantify the regularity with which a student interacts with the LMS, and whether those student interactions were predominately at consistent intervals or can be characterized as bursts.

One measure is through the fraction of days in which the student had at least one interaction with the course LMS (see Section 3.2.4 #3). This idea was encapsulated in the fraction of non-zero days feature, which was created by counting the number of days in

which the student had a non-zero number of interactions with the LMS within a given time period, and dividing that by the total number of days within that time period. If only a small fraction of the days contain LMS interactions, then the student is either interacting with the LMS minimally or grouping their interactions within a small number of active days. Alternatively, if the fraction is high then the student must be interacting with the LMS at least once per day fairly consistently throughout the term.

A second feature that was created to examine the spacing between LMS interactions was the Goh-Barabasi **burstiness** measure, defined in equation Equation 3.3.2 [GoBa08] (see Section 3.2.4 #9). Introduced by Goh and Barabasi, this measure is a way of quantifying activity patterns within a system as a value within the range $-1 \leq B \leq 1$. Bursts are characterized as short periods of time with high levels of activity, followed by longer stretches of time with decreased activity. Bursty patterns will have a B value close to 1, whereas steadier activity patterns, with more consistent inter-event times (also known as intervals), will have a B value close to 0. Bursty activity patterns have been observed in natural phenomena like earthquakes, as well as human activity such as email patterns [Bara05] [GoBa08]. The burstiness measure is calculated from the standard deviation $\sigma_{intervals}$ and mean $\mu_{intervals}$, of the timeline intervals.

$$B = \frac{\sigma_{intervals} - \mu_{intervals}}{\sigma_{intervals} + \mu_{intervals}} \qquad \text{(Equation 3.3.2)}$$

Each of the above features were computed over the full timeline, as well as up to designated points in the timeline. As with the previous features that were discussed, these designated points were: one quarter of the way through the term, half of the way through the term, three-quarters of the way through the term, the day after the VW deadline, the day after the midterm or first term test if there were multiple term tests, and the day after the final exam.

RQ 1.3: What are the predominate patterns of behaviour?

Student behaviour within the LMS can be characterized not only by the number of their interactions with the course content, but also by the amount of time in between those interactions. A student may interact with the course content at regularly spaced intervals, in quick succession with longer periods between bursts, or in varying patterns throughout the term.

The student timelines introduced in Section 3.2.3 that form part of the student models discussed in Section 3.2.2 were designed to study the periods of time between student interactions with the LMS. Individual timelines were created for each student, three of which were randomly selected from Summer 2021 and shown in Figure 3.5 as an example in order to visually illustrate the concept. The length of each interval is shown on the vertical axis, and the number of interactions each student had with the LMS is on the horizontal axis. For example, it can be seen that each student has one or more longer intervals of at least a week. This means that the student did not interact with any assignments, quizzes, or content pages during that time for the duration of a week or more.

As well, students may not interact with every item in a course. Indeed, this is seen in Figure 3.5 as the number of LMS interactions along the horizontal access is different for each student. In this group, Student 1 has 107 interactions, Student 2 has 241 interactions, and Student 3 has 207 interactions.

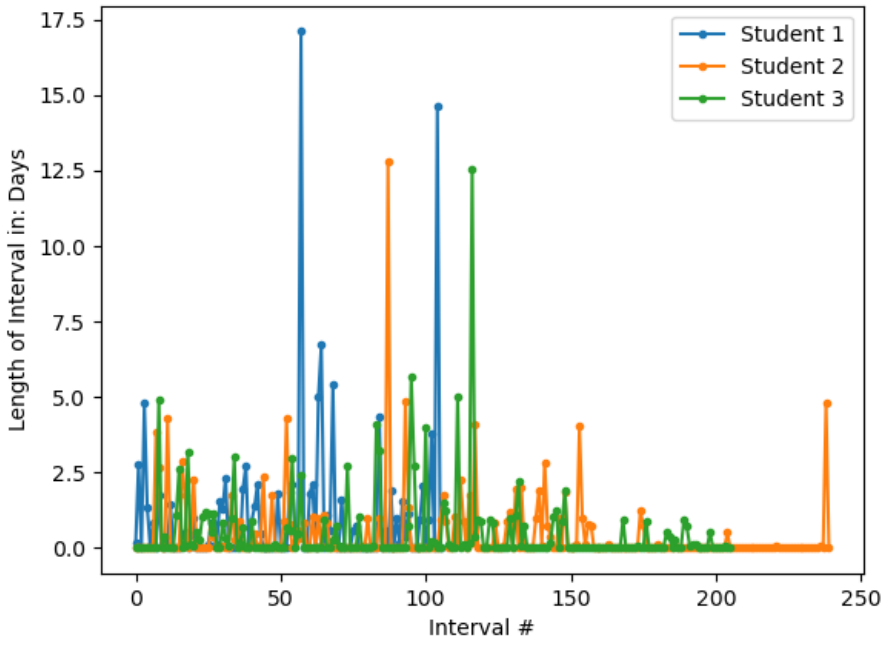

Fig. 3.5: Example student timelines from Summer 2021, selected at random.

The intervals are a variable that can be used to characterize the student timelines. The **probability mass function** (PMF) of the intervals of all students was calculated in order to explore the properties of this variable.

3.4 Group 2 RQs: Correlation of Features and Outcomes

What are the correlations between students' patterns of behaviour over the entire course and student final grades? Six sub-questions address how the features created from the raw student LMS data are correlated with each other, as well as with course outcomes.

RQ 2.1: Are time and pattern correlated? If so, how?

The intervals, or inter-event times, obtained from the student timelines have a probability mass function as addressed in research question 1.3. To evaluate how the intervals may change over the term, the statistical properties of the intervals were calculated at multiple points in time.

The first four moments were calculated: mean, variance, skewness and kurtosis (see Section 3.2.4 #5−8). The skewness is a measure to quantify the symmetry of a distribution. A distribution that is nearly symmetrical will have a skewness close to zero, whereas a distribution with a long tail is considered positively or negatively skewed depending on the direction of the tail. A distribution with a tail extending to the right will have a positive skewness, and if the tail extends to the left the distribution will have a negative skewness. The Fisher-Pearson sample skewness, used in this work, is a common skewness measure that is calculated as shown in Equation Equation 3.4.3. The skewness of the interval lengths indicates whether shorter or longer intervals are more likely (positive or negative skewness, respectively), or if they are equally likely (with near zero skewness). The skewness of the distribution of interval lengths serves is a numerical measure that can provide insight into behavioural patterns. If the distribution is largely symmetrical, then the interactions are occurring at consistent intervals. Otherwise, a positively skewed distribution indicates that the interactions are occurring in bursts (many short intervals separated by fewer long intervals) and a negatively skewed distribution indicates a lack of engagement with the LMS (that is, behaviour characterized by long spans of time between successive interactions).

$$g_1 = \frac{m_3}{m_2^{3/2}} \qquad \text{(Equation 3.4.3)}$$

The kurtosis was similarly calculated as the Pearson kurtosis, defined in Equation Equa-

tion 3.4.4. Kurtosis is a measure to quantify outliers in a distribution, as shown through the tail of the distribution. A distribution with more outliers, or more extreme outliers, will have a higher kurtosis. A high kurtosis of the interval lengths indicates that there are many long intervals or that some intervals are much longer than the average. Like skewness, the kurtosis is a numerical measure that can be used to characterize behavioural patterns with the interval lengths. A higher kurtosis indicates the presence of more extreme outliers within the interval lengths, either more very short or very long intervals depending on the skewness. Therefore, a higher kurtosis implies a that interactions with the LMS occur in longer intervals punctuated by bursts of activity, and a lower kurtosis implies that the lenght of time between intervals is consistent over time.

$$g_2 = \frac{m_4}{m_2^2} \qquad \text{(Equation 3.4.4)}$$

In the calculation of both skewness and kurtosis, m_i is the ith central moment calculated from N samples with mean $\bar{x}$ as shown in Equation Equation 3.4.5.

$$m_i = \frac{1}{N}\sum_{n=1}^{N}(x[n] - \bar{x})^i \qquad \text{(Equation 3.4.5)}$$

RQ 2.2: Are time, pattern and events/content (e.g., quizzes, assignments, exams) correlated? How?

Six features have been introduced that are related to the timeline intervals: the number of intervals, interval mean, variance, skewness, kurtosis, and the burstiness measure (B) described in RQ 1.2 [GoBa08]. As well, five additional features were introduced related to the midterm writing time, assignment hand in time, number of days with at least one interaction, the number of interactions, and the number of timeline items.

The number of intervals is directly related to the number of timeline items, so only one of those features needs to be used to capture the same information. In order to see how the other features are related, the correlation between each of the features with each of the other features was calculated at each quarter point in the term, as well as at the first term test (midterm) and VW deadline.

RQ 2.3: Are time, pattern, and student grades correlated? How?

To explore the relationship between the timeline intervals and final grades, the correlation between the final grade and each of the features of the interval mean, variance, skewness, kurtosis, and burstiness was calculated.

RQ 2.4: Are time, pattern, events/content, and student grades correlated? How?

The relationship between the final grade and each of the five additional independent features was also explored by calculating the correlation between the final grade and each of these features related to the midterm writing time, assignment hand in time, number of days with at least one interaction, the number of interactions, and the number of timeline items. If some features are highly correlated, they may contain similar predictive information and can be removed from predictive models to reduce the dimensionality of the input vectors.

RQ 2.5: Can these patterns be described in terms of high or low engagement?

To assess the level of engagement with the course material beyond the number of interactions, the number of repeated interactions was calculated as the average number of

interactions per student divided by the average number of timeline items per student.

RQ 2.6: Are there patterns of behaviour that are related with student outcomes in the course assuming students are grouped as pass/fail?

Although student outcomes in a course can be measured as a percentage final grade, it is often more useful to consider the outcome as whether or not a student has passed the course. In the COMP 1010 dataset, the minimum passing grade was 50%.

The percentage final grade can also be used to define student outcomes as one of five categories, based on the *Canadian Engineering Accreditation Board* (CEAB) taxonomy, which engineering educators must report on for each student in each course as part of the engineering program accreditation requirements. These categories, with the associated final grade percentages for this dataset, are:

- Strong: a final grade of A+ or A, defined as $\geq 80\%$.
- Competent: a final grade of B+ or B, defined as $\geq 70\%$.
- Developing: a final grade of C+ or C, defined as $\geq 60\%$.
- Needs Work: a final grade of D, defined as $\geq 50\%$.
- Failed: a final grade of F, defined as $< 50\%$.

The first four categories above (Strong, Competent, Developing, Needs Work) can also be grouped together as a single 'Passed' category, for the purpose of evaluating students on a binary pass/fail scale.

3.5 Group 3 RQs: Feature-Based Prediction and Early Prediction

Which variables and features of behaviours (time, pattern, events) have the greatest predictive capabilities for student outcomes? How early within the term can these patterns predict final course outcomes in terms of final grades, or passing or failing the course? Put another way, what is the horizon of predictability? The following five sub-questions further explore how the student features can be used for prediction.

RQ 3.1: What factors predict student success in an online environment?

The features introduced in Sections 3.2.4 and 3.2.5 were used as input to multiple supervised ML prediction models.

The first ML model used was linear regression, which accepts quantitative independent variables as input and predicts a continuous variable. The logistic regression model also accepts quantitative independent variables as input, however it is used to predict membership in a discrete class such 'Passed' or 'Failed'. Similarly, the k-Nearest Neighbours model is also used to predict membership in a class, rather than a final grade percentage.

For each model, the features to be used as input are chosen. Then, any students with non-numeric feature values are removed from the dataset. For example, if a student has only one item on their timeline then it is not possible to calculate interval statistics, and that student is removed from the dataset. The k-Nearest Neighbour model is sensitive to the feature values, so the features are scaled before use.

Next, the remaining students are randomly split into a training group and a testing group. The training group is used to train the ML model being used, while the testing group is used to test the predictive capability of the trained model. To ensure that the

experiments are repeatable, the random number generator is seeded with a known value before the random split occurs.

RQ 3.2: Can we define a set of archetypes (student behaviour + course outcome)?

Dimension reduction using *Principal Component Analysis* (PCA) was applied to create a set of two new features, called principal components, that are hybrids of the information contained in the features introduced in Section 3.2.4. This was done so that the students could be grouped and visualized on a two-dimensional scatterplot. Unsupervised clustering was used to group the students according to the two principal components, and membership in each cluster was compared to student outcomes. This experiment was repeated using the ML learned features introduced in Section 3.2.5 to observe how students were clustered based on the features that were hand-crafted from the Student Models compared to the features that were created from learned relationships in the raw data by the ML algorithm.

RQ 3.3: How early within the term can these factors predict final grades?

Put another way, is there an horizon of predictability for student outcomes? If so, what is the length of the horizon of predictability? To address this research question, the features calculated at partial points in the timeline were used as input to the linear regression prediction model described in RQ 3.1.

RQ 3.4: Can these variables and features be used to predict course outcomes of pass or fail before the end of the term, to indicate when intervention may be required?

It is not always necessary to predict final letter or percentage grades. To determine

whether intervention is required, being able to predict an outcome of passing or failing the course may be sufficient. To explore this question, the features calculated at partial points in the timeline were considered for the ability to predict a passing or failing outcome rather than final letter grade.

3.6 Group 4 RQs: Time Series Classification

With the advancement of ML techniques it can be valuable to see how well newer prediction algorithms perform on data, particularly complex data such as human interactions within an LMS. Techniques such as linear regression, logistic regression, k-Nearest Neighbours, and unsupervised clustering require features to be extracted from the raw timestamp data produced by an LMS. This feature extraction, also known as feature engineering, requires domain-specific knowledge and human oversight to develop pertinent features from the data at hand. In contrast, there is interest in ML techniques that can operate on the raw data itself, thereby removing the majority of the human cognition and perhaps potential human bias from the process.

Neural network models offer an opportunity to perform predictive analysis of time series data, without human-defined features, through a branch of research called time series classification. Although the raw data still needs to be pre-processed to ensure that it is in a compatible format, the resulting input data is maintained as a time series rather than a set of features. Although neural network-based models can make predictions of which classification category an input sample belongs to, these models do indicate the basis for that prediction. Therefore, these models are less interpretable than feature-based ML models.

3.6.1 Time series Classification Methodology Overview

A number of experiments were conducted to predict student course outcomes with the LMS timestamp data, using multiple types of neural networks and methods of pre-processing the raw data. In general, each experiment consisted of the following steps:

1. Pre-process the data
2. Build a Neural-network ML model
3. Evaluate the ML model

To address research questions 4.1 and 4.2, the full timeline data was processed and used. For research question 4.3 on early prediction, only timeline data up to the VW deadline was included. Although these three steps are described in more detail in Subsections 3.6.2 to 3.6.4, a brief overview of these steps is provided below:

Step 1: The data was preprocessed in the following ways:

1. intervals as input data
2. time stamps as input data
3. labelled time stamps as input data (multivariable data)

Step 2: For each of the three sets of preprocessed data, the following ML models were built:

1. CNN binary classifier
2. CNN ternary classifier with categories based on letter grades: {A+, A, B+, B}; {C+, C, D}; and {F}
3. CNN ternary classifier with categories of: {students with the top half of passing grades}; {students with the bottom half of passing grades}; {students who failed}

4. Transformer binary classifier
5. Transformer ternary classifier with categories based on letter grades: {A+, A, B+, B}; {C+, C, D}; and {F}
6. Transformer ternary classifier with categories of: {students with the top half of passing grades}; {students with the bottom half of passing grades}; {students who failed}

Step 3: Each of the three sets of preprocessed data was used as input in each of the six ML models, resulting in **eighteen experiments**. The performance of the ML models were evaluated individually and in relation to the other experiments, as described below.

3.6.2 Step 1: Data Preprocessing

The student timestamp data was preprocessed using four steps to create data sets that were standardized and rectangular for input into the ML models. The neural-based ML models required the data to be standardized as they can be sensitive to outliers [Theo19]. Further, the ML models required a rectangular data set, meaning that the length of each student data vector must be the same.

The first preprocessing step was to standardize the data over time. The raw student timestamp data was standardized over time in two ways: using intervals and by standardizing the start date. The first data set was created using the timeline intervals that were developed for the earlier experiments (as described in 3). These intervals are naturally date-insensitive. Intervals are the lengths of time between successive LMS interactions, and are not anchored to a calendar date. For example, the interval between June 1, 2020 at 8:00am and June 1, 2020 at 9:00am is the same as the interval between September 1, 2020 at 10:00am and September 1, 2020 at 11:00am. Therefore, intervals provide a method of standardizing the time dimension for courses held across multiple terms. The second data

set was created by standardizing all raw timestamp data relative to an initial start date. The initial standardized start date was arbitrarily chosen to be June 1, 2000. This date was substituted as the start date for each of Course A, B, and C described in 3.5, and the timestamps in each course were shifted to be relative to the new start date. The end date of each course was not standardized in order to retain the relational information between the timestamps, under the assumption that humans work in units of minutes, hours, and days, rather than in units of percentage of a course term.

The second preprocessing step was to normalize the data values. After standardizing the raw data over time, using either method, the data was then normalized using the z-score function to reduce the impact of outliers on the neural-based ML models. The z-score[3] is defined as shown in equation Equation 3.6.6, and normalizes the data to have a mean of zero and standard deviation of one.

$$z = \frac{(x - \mu)}{\sigma} \qquad \text{(Equation 3.6.6)}$$

The third preprocessing step was to standardize the lengths of each individual student data array. As shown in 3.5, students did not each interact with the LMS the same number of times. This means that the number of timestamps, and consequently the number of intervals, recorded for each student was different. In order to standardize the number of interactions across all students, the student with the greatest number of interactions in the raw data set was found (the student array of maximum length). Then, all students arrays with fewer interactions were extended to be the maximum length and filled with the mean value of zero.

Finally, the optional fourth preprocessing step was to create **multivariable** data by labelling the timestamps with interaction type. This step was only applied to the standard-

[3]The z-score was calculated in Python using the scipy.stats.zscore library.

ized timestamp data (from the first preprocessing step) not the intervals. This was for two reasons: the intervals between certain interaction types, such as quizzes, were dictated by the instructor rather than the students; and the number of intervals between certain interaction types, such as assignments, would be prohibitively small. The standardized timestamp data was organized either as a 1-dimensional array per student with all interactions, or as multivariable data in a 4-dimensional array per student with additional rows indicating the type of interaction using one-hot encoding. Examples of the timestamp and multivariable organization are shown in Table 3.6.

Table 3.6: Example of standardized timestamp data organized as a 1-dimensional array (timestamps only) and as a multivariable array with one-hot encoding for interaction type.

Timestamp	00:00:00:01:06:2000	00:04:08:01:06:2000	21:30:10:05:06:2000

Timestamp	00:00:00:01:06:2000	00:04:08:01:06:2000	21:30:10:05:06:2000
Assignment	0	0	0
Quiz	0	0	1
Content	1	1	0

3.6.3 Step 2: Building the models

The specific ML models used were CNN and transformers for classification. CNN classifiers are a more established type of neural network that have shown promise with time series classification tasks [Fawa20] [RoTB21] [WaYO17], and transformers are a newer type of neural network that have grown out of the field of *Natural Language Processing* (NLP) and have more recently been applied to time series classification tasks as well [Ntak21].

The binary classifiers were trained to predict whether students would pass or fail the course, based on a passing grade of $\geq 50\%$. The ternary classifier was trained to predict

whether students would pass the course with a high grade, pass the course with a low grade, or fail the course. A high and low passing grade was defined in two ways. In the first way, the classification groups were defined based on the students' final letter grades, with a high passing grade was defined as $\geq 70\%$ (a grade of B or higher), a low passing grade was defined as $\geq 50\%$ and $< 70\%$ (a grade of C+, C, or D), while $< 50\%$ was considered a failing grade (a grade of F). In the second way, the classification groups consisted of students in the top half of the passing group, students in the bottom half of the passing group, and students who failed. That is, the classification groups were defined based on the median passing grade. The second type of classification groups for ternary classification were chosen to create more balanced group sizes. In both cases, students who passed the course with a low passing grade were considered the "warning" group, to indicate that instructors (or the students themselves) could be warned that these students are not firmly on track to pass.

In order to train and test all of the models, the processed input data sets were split into randomized sets in the following way:

- 70% of the students were put into a training set,
- 20% of the students were put into a testing set, and
- 10% of the students were put into a validation set.

Therefore, 70% of the students were used for training the models, and a total of 30% of students were used for testing [PeVG11]. For each model, the training set was used to train the model to produce either binary or ternary predictions. In all cases the maximum number of training epochs was set to 500, and the validation set was used to allow for an early training exit in order to minimize overfitting. Each trained model was then evaluated based on its accuracy at predicting students in the associated testing set.

3.6.4 Step 3: Evaluating the models

In each of the eighteen experiments, the ML models were evaluated for their performance on the given data set in the following ways:

1. Testing accuracy, defined as the overall percentage of students in the testing set classified correctly.

2. Number of true positive, false positives, true negatives, and false negatives. The number of false positives (i.e. students who were predicted to pass but did not) was of particular interest, to identify students most in need of support who might be missed using the model.

Each type of model was built multiple times, using different random seeds, and both the average performance and best performance of the model in each type of experiment was reported. The number of random seeds used for each of the CNN classifiers was 100, and the number of random seeds used for each of the transformer classifiers was 20, due to the relatively long running time of the transformer classifiers. It took approximately 30 seconds on average to train and test one CNN model, and 15 minutes to 2 hours to train and test one transformer model, using the available hardware.

3.7 Methodology Summary

This chapter introduced the type of data collected by UM Learn, the University of Manitoba implementation of the Brightspace LMS from Desire2Learn. The design of a code-based tool was presented to model courses and students based on LMS data. Numeric, temporal features were created from the models to represent students' interactions with the LMS over the term. Experiments for the first three groups of research questions were proposed to explore students' behaviours in an online and synchronous course, examine the

relationship between students' behaviours as their course outcomes, and test the suitability of the features for student outcome prediction in ML algorithms. Experiments for the fourth group of research questions related to time series classification without features were also proposed and described.

Chapter 4

Results

The following chapter presents the results of the experiments outlined in Chapter 3. There are four groups of experiments related to the four groups of research questions in Section 1.2.3. In the first set, an exploratory analysis of students' patterns of behaviours within the LMS is performed using intervals and other temporal features created from the timelines in the Student Model of chronologically ordered interaction timestamp data. In the second set, the correlations between these hand-crafted temporal features with each other and with course outcomes are calculated. In the third set, the temporal features created from the Student Model timelines are used in ML algorithms to predict student outcomes, and the results are compared to student outcome predictions from features created by a ML algorithm from the raw timestamp data. In the fourth set, time series classification is performed with CNN and transformer neural networks using the LMS timestamp data to predict student course outcomes.

4.1 Group 1 RQs: Patterns of Behaviours

In this set of research questions, students' patterns of behaviour throughout a term for a class delivered fully online were explored by creating quantitative features to represent the students' temporal interactions with the LMS. These features were created from a dataset created out of LMS interactions in three iterations of an undergraduate computer science class.

RQ 1.1: What are students' patterns of behaviour on an LMS over time (over time periods, within a single interaction/time, between interactions, time spent on assignments/quizzes, in relation to course events such as assignment deadlines and VW dates)

The first sub-question regarding patterns of behaviours looked at the statistical properties of four features: the number of interactions with the LMS per student, the number of items on each student's LMS timeline, the average difference in time between assignment due dates and when they were submitted, and the fraction of the allotted midterm (first term test) time that was used.

To explore the LMS interactions over time, the features were calculated at various points throughout the term. The average values of these features were calculated at the quarter-points in the term, at the VW deadline and at the midterm (first term test), and exam. The LMS features, shown in Table 4.1, were the total number interactions with the LMS course page, the total number of timeline items, and the amount of time between assignment submissions and their deadlines.

Table 4.1: RQ 1.1: Statistical Properties of Features

Num. Interactions	1st Quarter	Midterm	2nd Quarter	VW	3rd Quarter	Exam	All
Average	91.41	121.13	183.38	221.98	277.10	345.17	353.24

Num. Timeline Items	1st Quarter	Midterm	2nd Quarter	VW	3rd Quarter	Exam	All
Average	44	58.16	83.74	100.66	121.85	142.70	144.38

Assign. Hand-in (hrs)	1st Quarter	Midterm	2nd Quarter	VW	3rd Quarter	Exam	All
Average	42.64	41.66	32.06	30.46	27.54	27.18	27.18

As shown in Table 4.1, the average number of interactions students had with the LMS increased throughout the term at a constant rate. During each of the first quarter of the term the average number of interactions per student was 91. During the second quarter of the term, the average number of interactions was 92 (calculated as the average number of interactions up to the 2nd quarter minus the average number of interactions up to the 1st quarter, which was $183.384 - 91.4097 = 91.9743$, rounded to the nearest whole number). Similarly, the average number of interactions per student was calculated as 94 and 76 during the third and last quarters, respectively. Also shown in Table 4.1, the average number of timeline items students increased throughout the term at a nearly constant rate as well. During each of the first, second, and third quarters of the term the average number of timeline items per student was 44, 40, and 38, respectively. The average number of new timeline items per student in the last quarter was lower, only 23. The lower number of

interactions and timeline items during the last quarter could be due to the period of time after the final exam when the LMS course page remained open to students.

Since there is only one first term test (midterm), the fraction of allowed time spent writing the test does not change throughout the term. The properties of this feature were calculated and are shown in Table 4.2. On average, students used 93.4% of the allowed writing time for the midterm (first term test), with low variance. This means that the majority of students used their allotted time, and that most midterm-writing times were close to the average.

Table 4.2: RQ 1.1: Average fraction of time spent writing the midterm

Fraction of Midterm Time	
Average	0.934
Variance	0.072

RQ 1.2: What are students' patterns of behaviour on an LMS with regard to interactions and intervals?

For the second sub-question regarding patterns of behaviours, two measures were used to quantify whether a student interacts with the LMS in a consistent or bursty way: the Goh-Barabasi burstiness measure and the fraction of days with at least one LMS interaction. The average values of these features throughout the term is shown in Table 4.3.

Table 4.3: RQ 1.2: Statistical properties of features related to patterns of behaviour

Fraction of Non-Zero Days	1st Quarter	Midterm	2nd Quarter	VW	3rd Quarter	Exam	All
Average	0.47	0.48	0.44	0.44	0.43	0.43	0.36

Burstiness	1st Quarter	Midterm	2nd Quarter	VW	3rd Quarter	Exam	All
Average	0.95	0.95	0.96	0.96	0.96	0.96	0.96

The Goh-Barabasi burstiness measure, or just 'burstiness' measure, is calculated between 0 and 1. Values close to 0 indicate consistent patterns of activity, and values closer to 1 indicate bursty behaviours. At each point that it was calculated during the term, the burstiness measure was nearly 1, implying that students are predominately accessing the material in a bursty way, rather than in consistently spread out throughout the term. One explanation for the high degree of burstiness could be if students are downloading multiple files when they interact with the course LMS.

The fraction of non-zero days is the fraction of days that student's interacted with the course LMS at least once. On average, this value decreased over the term from approximately 0.47 at the one-quarter mark to 0.43 by the final exam, corresponding to a minimum of three days with interactions every week (three out of seven days is a fraction of 0.43 non-zero days). By the end of the term, the fraction of non-zero days dropped to 0.36, which may be attributed to decreased activity during the period of time after the final exam when the LMS course page remained open to students but the course was effectively over and therefore, student interactions with the LMS are not expected.

RQ 1.3: What are the predominate patterns of behaviour?

From Figure 4.1, it can be seen that the intervals were not all the same length and that the different interval lengths did not occur with equal frequency. For example, the majority of intervals were less than 48 hours long. The following two figures show probability mass functions to examine trends in interval lengths over all students. In both Figure 4.1 and 4.2, the vertical axis specifies the likelihood of each interval length in hours, when looking at intervals from all student data.

The majority of intervals were less than one hour in length. However, the distribution of intervals has a long tail in which the maximum interval length was 1029.1 hours (or over 42 days), and the minimum interval length was under one second.

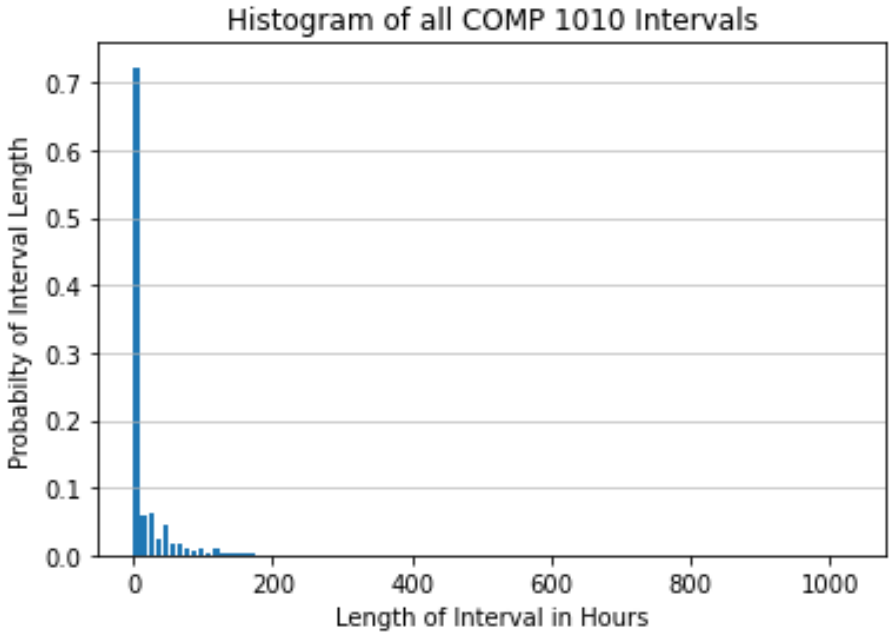

Fig. 4.1: Probability Mass Function (PMF) of interval lengths over all COMP 1010 students.

Due to the long tail in Figure 4.1, it is difficult to see any patterns in the distribution. Visually, the large peak on the left side of the distribution dominates the vertical axis and

the long tail to the right condenses the horizontal axis. In order to better visualize the middle of the distribution, intervals that were less than one hour or more than 168 hours (one week) were removed from the dataset to produce Figure 4.2. Intervals less than one hour were removed so that, visually, the y-axis of the distribution would not be dominated by the large peak near zero. Intervals of more than one week were removed so that, again visually, the x-axis of the distribution would not be dominated by the long tail to the right.

In Figure 4.2, it can be seen that not all interval lengths in the middle of the distribution are equally common. For intervals greater than one hour long, the distribution follows a decaying sinusoidal pattern with peaks corresponding to roughly 24-hour periods. This means that intervals lengths that were multiples of 24 hours were more common than other interval lengths, and that shorter intervals were more common than longer intervals. For example, intervals of roughly 24 hours were more common than intervals of roughly 48 hours.

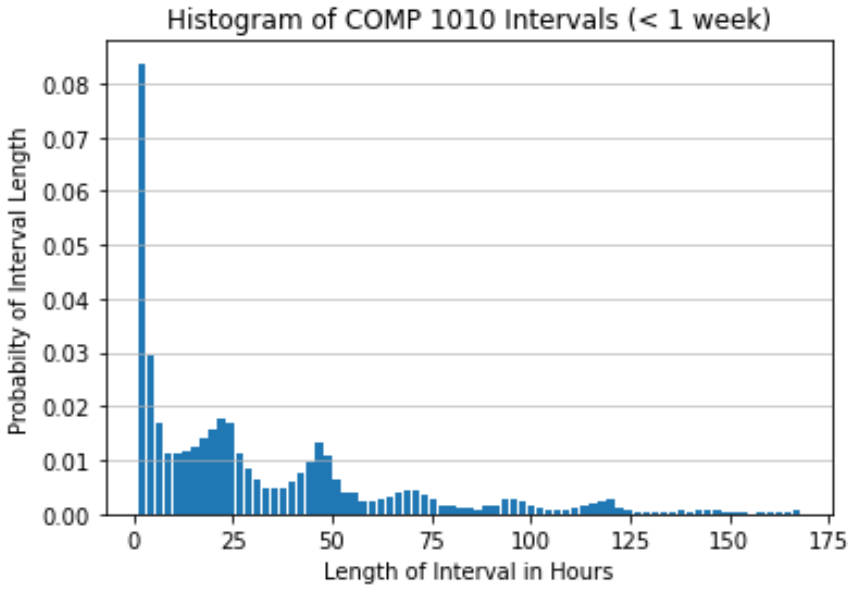

Fig. 4.2: Probability Mass Function (PMF) of interval lengths over all COMP 1010 students, with intervals of less than one hour or more than one week removed.

4.2 Group 2 RQs: Correlation of Features and Outcomes

Through the following research questions, the correlations between students' patterns of behaviour over the entire course and students' outcomes are explored. The correlations are calculated at each quarter point through the term. Students' patterns of behaviour are quantified through the interval characteristics including mean, variance, higher order moments, and the burstiness measure. Three event points are denoted: the first test (midterm), VW deadline, and final exam. As well, three additional features to characterize students' interaction with the LMS content are also calculated. These features are the amount of time spent writing the first test (midterm), the average time between submission of assignments and their deadlines, and the fraction of days with LMS interactions.

The correlation between features, as well as between the features and course outcomes, was explored in this research question. Course outcomes were quantified as the final grade as a percentage, as a binary pass or fail, and as a competency level based on the final grade. For the binary pass or fail metric, a passing grade of $\geq 50\%$ was used, keeping in line with what was communicated to students in the course. The competency levels were based on the *Canadian Engineering Accreditation Board* (CEAB) categories of strong (A+ or A), competent (B+ or B), developing (C+ or C), and needs work (D), with the additional category of failed (F) included to encompass the full spectrum of grades.

RQ 2.1: Are time and pattern correlated? If so, how?

To observe student interaction behaviour with the LMS over time, the statistical properties of the interval lengths were calculated as shown in Table 4.4.

Table 4.4: Statistical Properties of Interval Lengths Across All Students (units in hours)

	1st Quarter	Midterm	2nd Quarter	VW	3rd Quarter	Exam	All
Mean	19.71	17.98	20.17	20.45	20.33	18.90	18.77
Variance	2027.78	1830.94	2481.44	2516.29	2482.41	2353.40	2356.38
Skewness	2.19	2.44	3.13	3.14	3.15	3.54	3.60
Kurtosis	5.04	6.87	13.47	14.24	14.62	19.30	20.27

As shown in Table 4.4, the average interval length stayed around 20 hours at each point during the term, increasing from the midterm until the VW deadline and then decreasing again. The variance of the interval lengths similarly increased towards the middle of the term, and then decreases slightly. Both the skewness and kurtosis monotonically increased over the term.

RQ 2.2: Are time, pattern and events/content (e.g., quizzes, assignments, exams) correlated? How?

The correlation between the independent features in the bulleted list below was calculated at multiple points in the term.

- Interval properties:
  - mean of interval length
  - variance of interval length
  - skewness of interval length
  - kurtosis of interval length
  - burstiness

- Other independent features:
  - midterm writing time
  - difference between assignment submission and due dates
  - fraction of days with at least one interaction (non-zero days)
  - number of interactions
  - number of timeline items

The correlation between the features is shown in Figures 4.3 to 4.8. As observed in these figures, the number of interactions is highly correlated with the number of timeline items at all points during the term. The number of interactions, and number of timeline items, is also highly correlated with the fraction of non-zero days.

As expected, the number of interactions and timeline items is negatively correlated with the mean interval length at all points throughout the term: as the number of interactions increases, the amount of time between those interactions must decrease on average. However, the number of interactions and timeline items is positively correlated with the higher order moments of the interval lengths, skewness and kurtosis. As the number of interactions increases, the distribution of the intervals becomes less normal.

Both the burstiness measure and average assignment submission time exhibit a low correlation with each other and other features. The fraction of midterm writing time is most correlated with the fraction of non-zero days.

The correlations between features shown in Figures 4.3 to 4.8 are colour-coded to based on the strength of the correlation. Values closer to 1 are coloured in red, while values closer to -1 are coloured in blue. The colour scale is shown on the right side of each figure. To quantify the significance of these correlations, the p values are provided in

Appendix C.

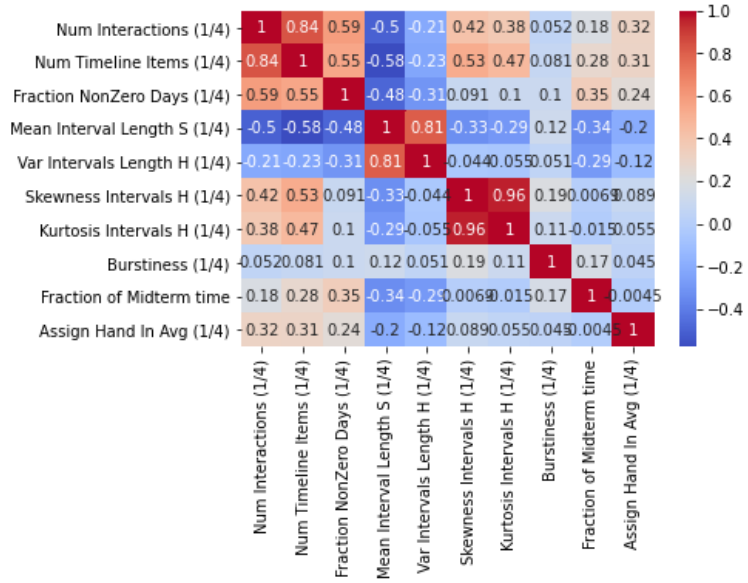

Fig. 4.3: Correlations between the temporal features with each other, at the first quarter point in the term.

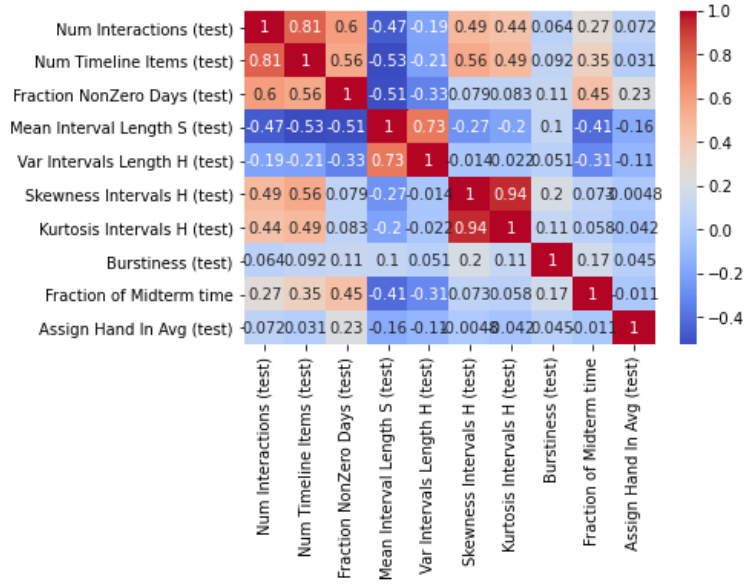

Fig. 4.4: Correlations between the temporal features with each other, at the midterm (first term test).

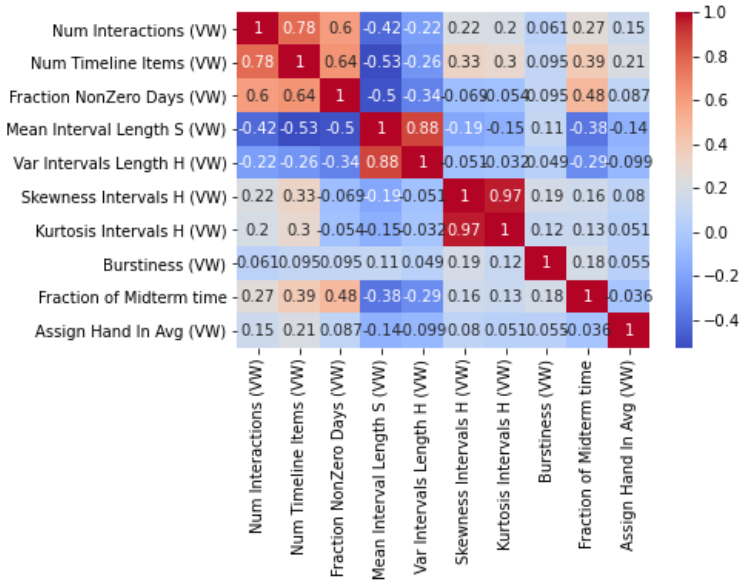

Fig. 4.5: Correlations between the temporal features with each other, at the VW deadline.

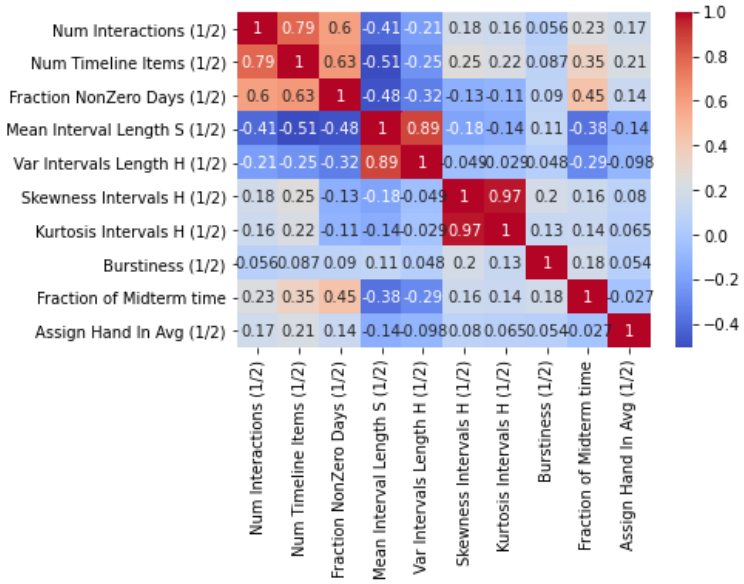

Fig. 4.6: Correlations between the temporal features with each other, at the halfway point in the term.

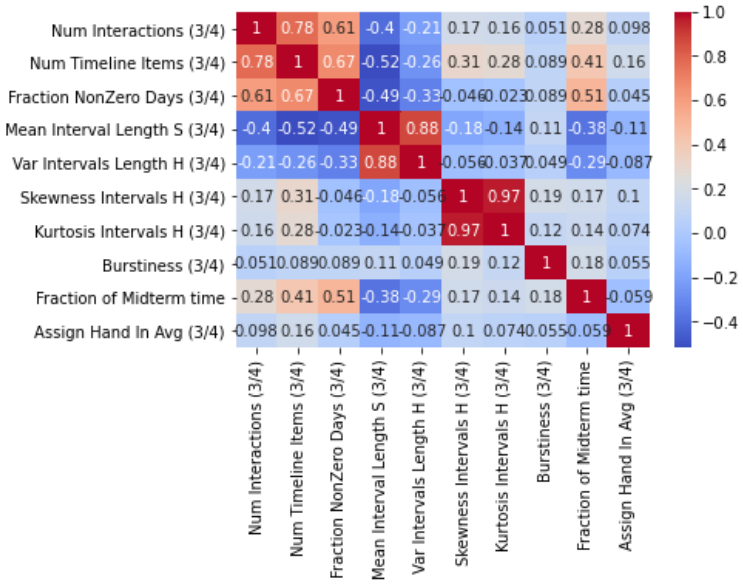

Fig. 4.7: Correlations between the temporal features with each other, at the three-quarters point in the term.

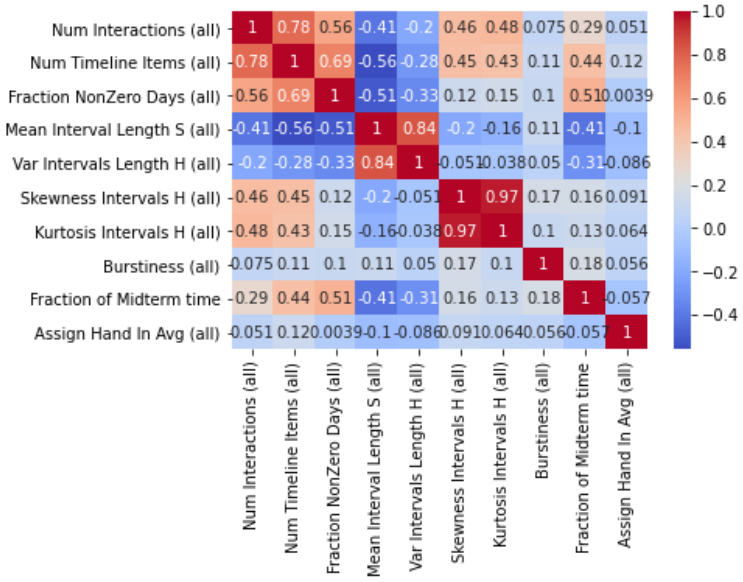

Fig. 4.8: Correlations between the temporal features with each other, over the whole term.

RQ 2.3: Are time, pattern, and student grades correlated? How?

The correlation between the interval features, listed below, and both the midterm and final grade was calculated and are as shown in Figures 4.9 to 4.14.

Interval properties:

- mean of interval length
- variance of interval length
- skewness of interval length
- kurtosis of interval length
- burstiness

As shown in Figures 4.9 to 4.14, the mean of the interval lengths has a negative correlation with the final grade. The variance of the interval lengths is also negatively correlated with the final grade, albeit less strongly. The higher order moments of the interval lengths, as well as the burstiness measure, do not exhibit a strong positive or negative relationship with the final grade. These relationships are consistent at each point of time examined during the term. Finally, the midterm grade strongly correlates with final grade. However, this relationship does not provide insight on student behaviour throughout the term over time.

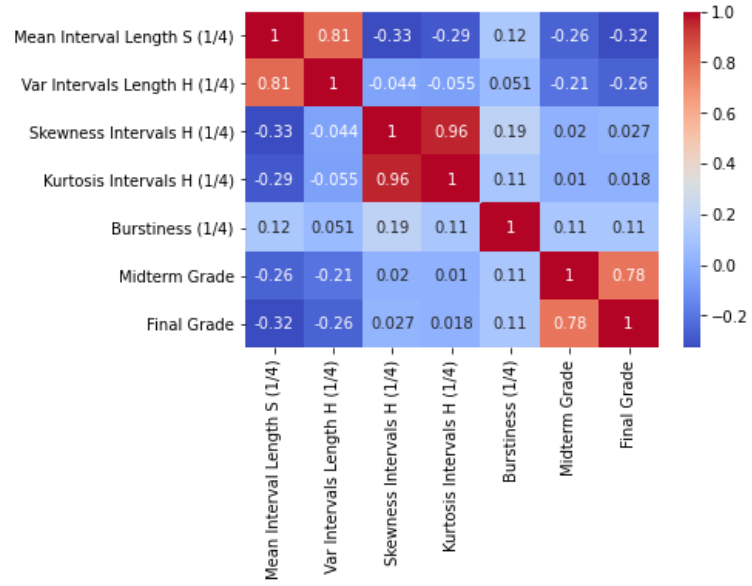

Fig. 4.9: Correlations between the interval features with each other, at the first quarter point in the term.

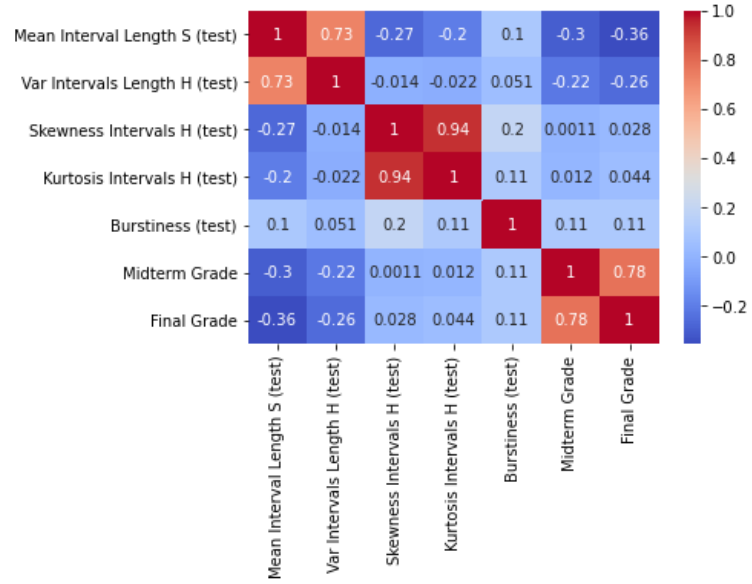

Fig. 4.10: Correlations between the interval features with each other, at the midterm (first term test).

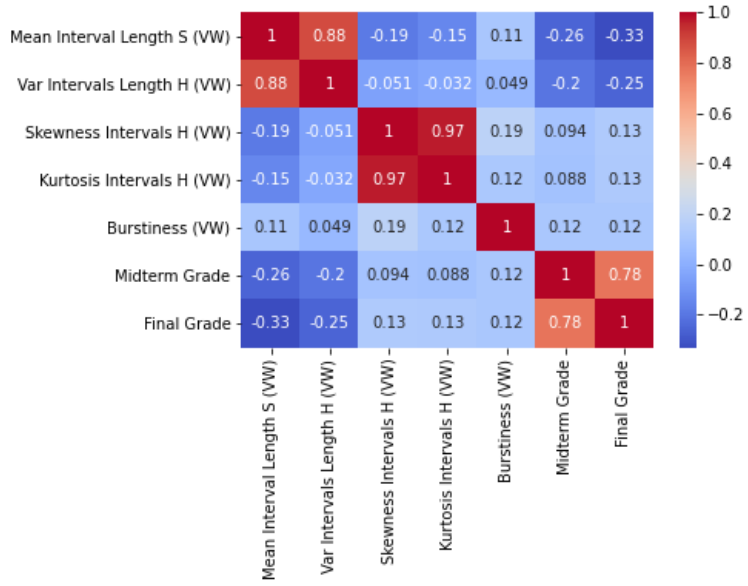

Fig. 4.11: Correlations between the interval features with each other, at the VW deadline.

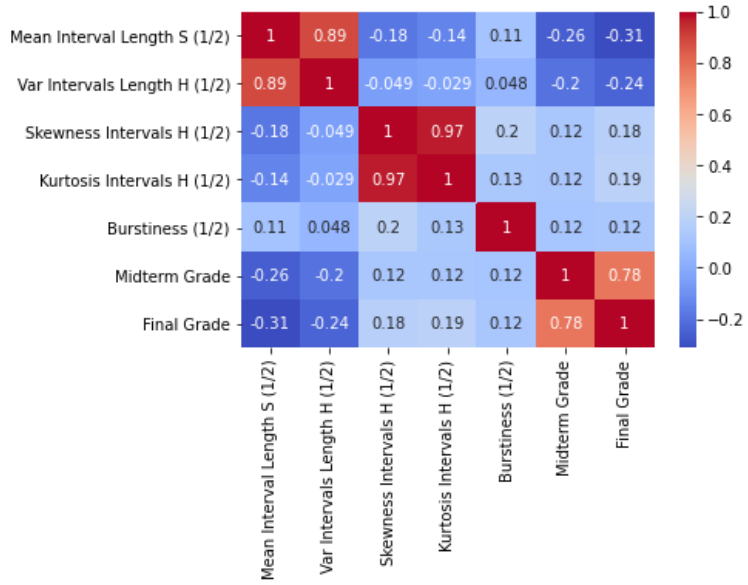

Fig. 4.12: Correlations between the interval features with each other, at the halfway point in the term.

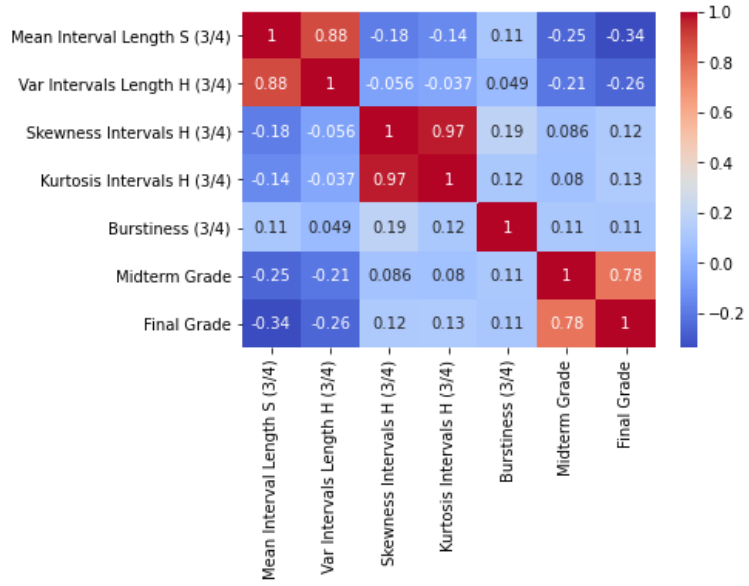

Fig. 4.13: Correlations between the interval features with each other, at the three-quarters point in the term.

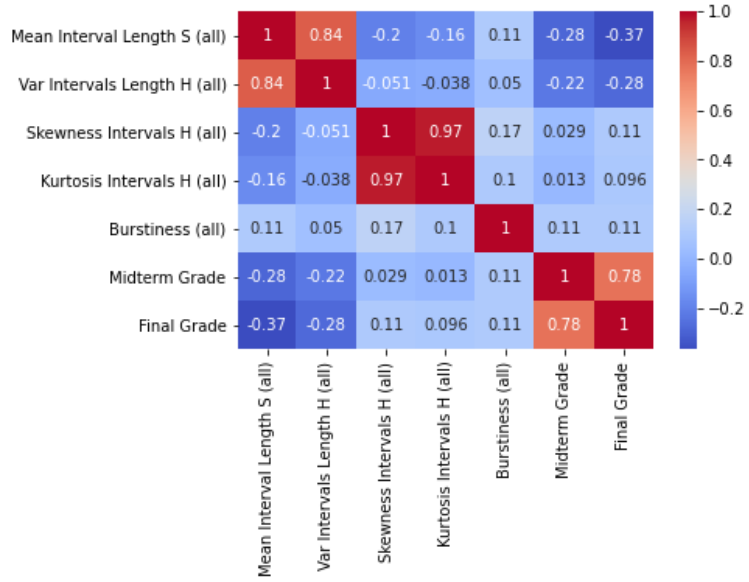

Fig. 4.14: Correlations between the interval features with each other, over the whole term.

RQ 2.4: Are time, pattern, events/content, and student grades correlated? How?

The correlation between the independent features listed below with each of the midterm and final grade was calculated:

- midterm writing time
- difference between assignment submission and due dates
- fraction of days with at least one interaction (non-zero days)
- number of interactions
- number of timeline items

In Figures 4.15 to 4.20, the correlation between the final grade and the features not related to interval lengths are shown. The number of interactions positively correlate with the final grade, and this correlation increases over the term from about 0.2 to 0.35. The correlation of the number of timeline items to the final grade is higher, increasing from 0.25 to 0.47 over the term. Overall, the correlation between the grades and the fraction of non-zero days increases over the term as well, from 0.33 to to 0.58.

The average assignment submission time relative to the due date showed very little relationship with the final grade at any point in the term. Whether students submitted their work well in advance of the due date, just in time, or late within the grace period, was not indicative of how they would finish the course. However, the amount of time a student spent writing the midterm was positively correlated with both their midterm grade (correlation of 0.58) and final grade (correlation of 0.55).

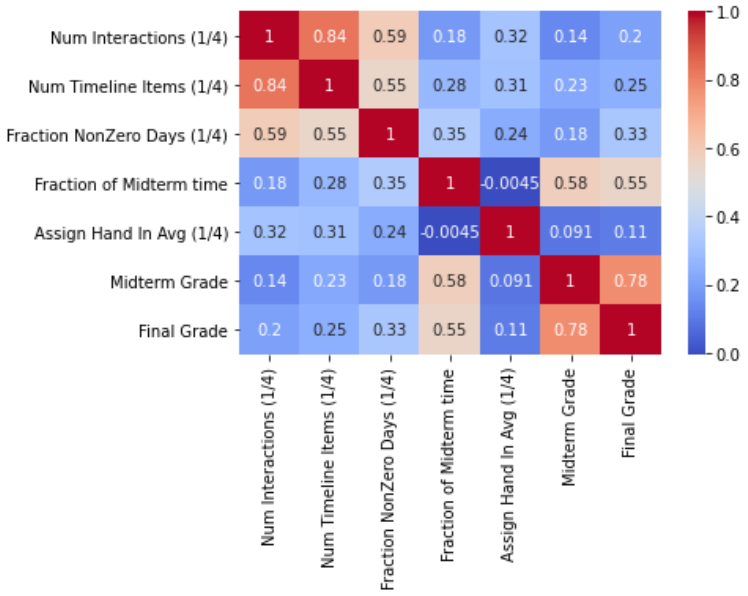

Fig. 4.15: Correlations between the temporal features with the grades, at the first quarter point in the term.

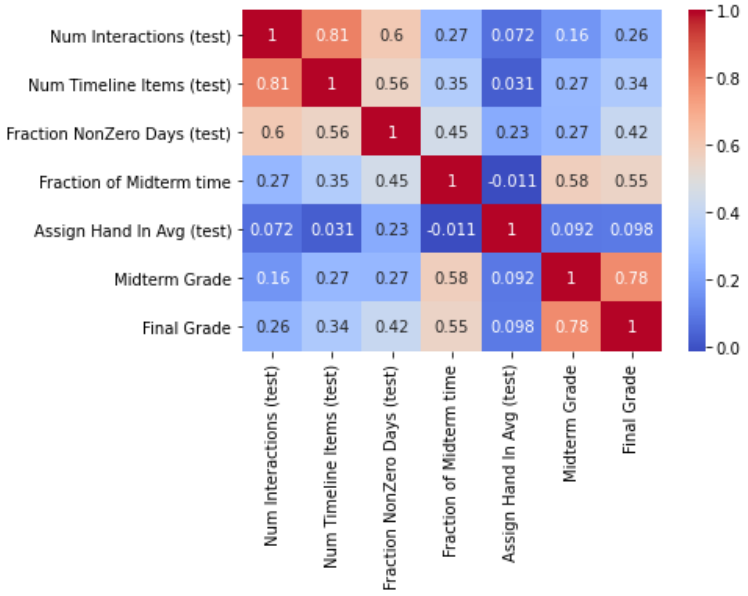

Fig. 4.16: Correlations between the temporal features with the grades, at the midterm (first term test).

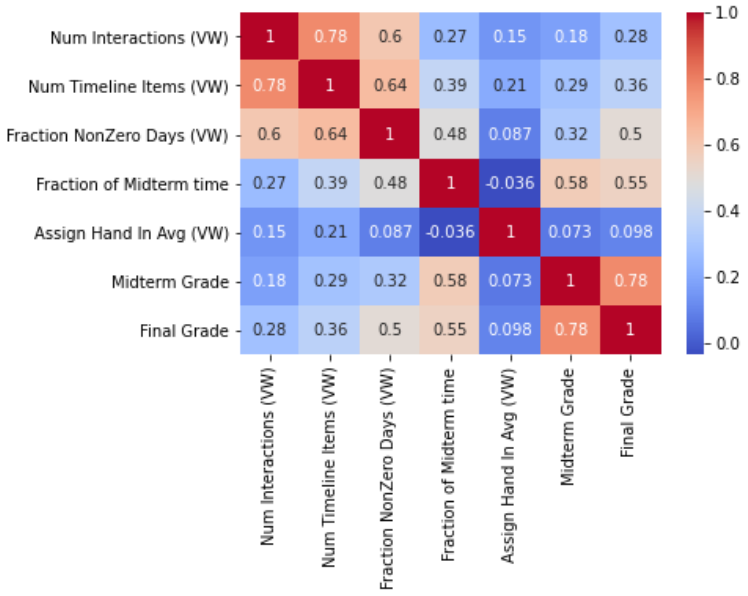

Fig. 4.17: Correlations between the temporal features with the grades, at the VW Deadline.

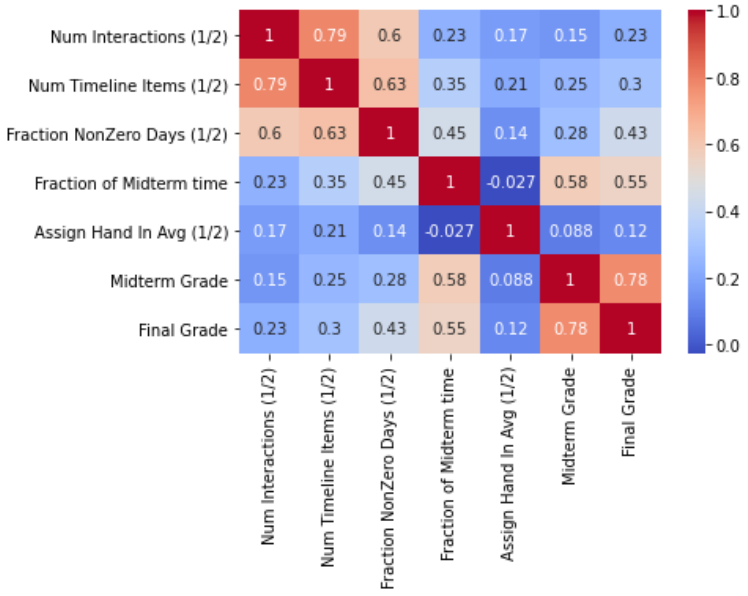

Fig. 4.18: Correlations between the temporal features with the grades, at the halfway point in the term.

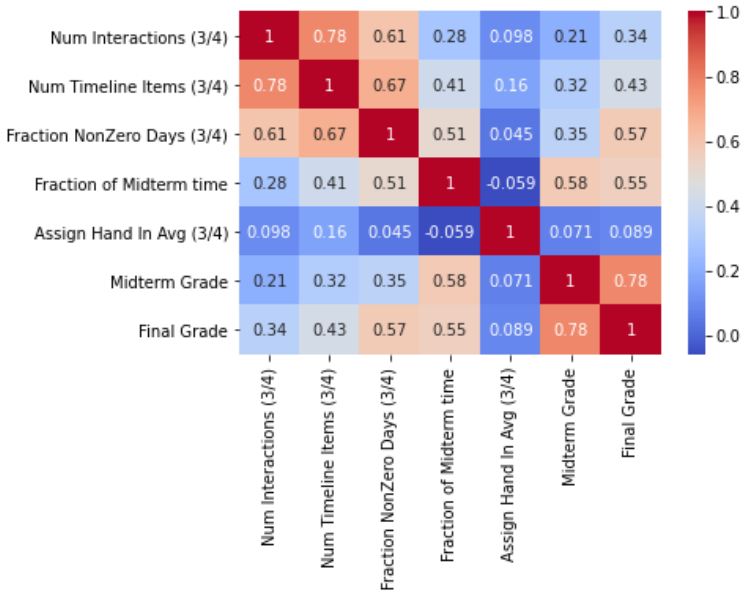

Fig. 4.19: Correlations between the temporal features with the grades, at the three-quarters point in the term.

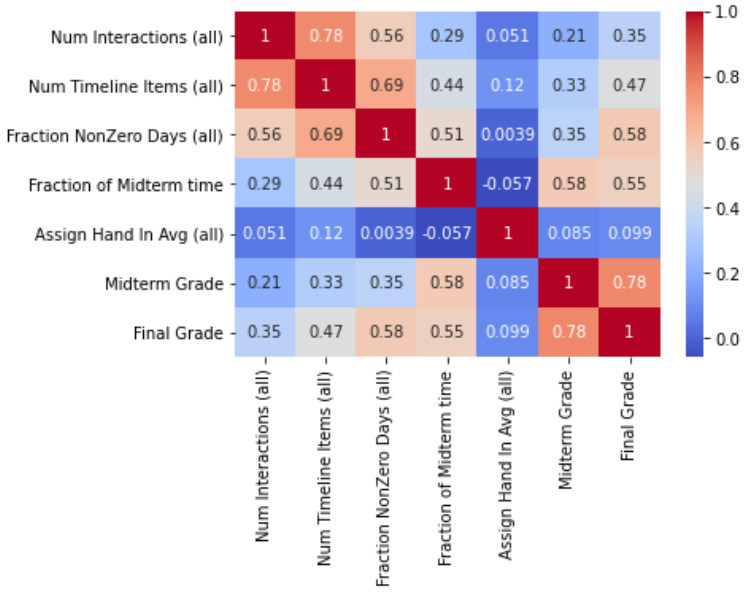

Fig. 4.20: Correlations between the temporal features with the grades, over the whole term.

RQ 2.5: Can these patterns be described in terms of high or low engagement?

The number of repeated interactions was calculated as the average number of interactions per student divided by the average number of timeline items per student. As shown in Table 4.5 the number of repeated interactions per student increased by less than one repeated interaction, from 2.08 to 2.45 over the term, indicating that the content was not frequently re-visited.

Table 4.5: The number of average repeated interactions per student was calculated as the average number of interactions per student divided by the average number of timeline items per student.

Point in term	1st Quarter	Midterm	2nd Quarter	VW	3rd Quarter	Exam	All
Repeated Interactions	2.08	2.08	2.19	2.21	2.27	2.42	2.45

RQ 2.6: Are there patterns of behaviour that are related with student outcomes in the course assuming students are grouped as pass/fail?

In order to examine the relationship between student behaviours and course outcomes, the features were compared based on whether students had passed or failed the course as shown in Table 4.6.

Table 4.6: Comparison of feature values based on passing and failing groups of students. All times are in hours.

Average Feature Value	Passing Group	Failing Group
Number of Interactions	386.70	232.67
Number of Timeline Items	156.22	99.41
Interval Length Mean	16.23	24.68
Interval Length Variance	1265.77	5031.69
Interval Length Skewness	3.64	3.43
Interval Length Kurtosis	20.87	18.25
Burstiness	0.97	0.93
Assign. Hand In Avg.	27.54	27.22
Fraction of Non-Zero Days	0.39	0.24
Fraction of Midterm Time	1.00	0.66

Most of the average feature values were different between the group of students who passed compared to the group of students who failed. However, the average difference between assignment submission and due date was similar across both groups.

Since more students passed the course compared to those who failed, the group of students who passed was further divided into four categories: strong, competent, developing, and needs work. These categories are based on the CEAB taxonomy and were introduced in Section 3.4. The average feature values for each group are shown in Table 4.7.

Table 4.7: Comparison of feature values based on groups of students designated by final grade categories. All times are in hours.

Average Feature Value	Strong	Competent	Developing	Needs Work
Number of Interactions	386.37	354.19	402.69	329.21
Number of Timeline Items	155.53	150.88	162.52	143.88
Interval Length Mean	16.99	17.81	15.31	16.87
Interval Length Variance	1546.01	1595.69	1141.71	1357.81
Interval Length Skewness	3.72	3.61	3.55	3.38
Interval Length Kurtosis	21.72	19.63	19.22	17.94
Burstiness	0.969402	0.97	0.97	0.97
Assign. Hand In Avg.	35.09	32.71	26.31	20.00
Fraction of Non-Zero Days	0.38	0.38	0.40	0.38
Fraction of Midterm Time	1.01	0.98	1.01	1.02

Within the four categories of students who passed, a larger number of LMS interactions and timeline items was generally associated with a higher final grade. The exception to this was in the developing category. These students, on average, had more interactions with the LMS than any other group. The students in the developing category also had the smallest interval length variance and highest fraction of days with at least one interaction. As well, within the four categories of students who passed the course, students who earned a higher final grade on average submitted assignments earlier.

4.3 Group 3 RQs: Feature-Based Prediction and Early Prediction

Predictive ML models were used to explore which variables and features of behaviours have the greatest predictive capabilities for student course outcomes as measured by final

grade or as a binary pass/fail metric.

For early prediction, the feature values at earlier points during the term were used as input to the predictive models.

RQ 3.1: What factors predict student success in an online environment?

The features introduced in the previous research questions were used as input into ML prediction models, in order to see which features were most predictive of student success at the end of the term. The ML models used were linear regression, logistic regression, and k-Nearest Neighbours. In each case, results are shown from trials when the full set of students was randomly divided such that 70% of the students were used to train the model and the remaining 30% were used to test the model. Reserving 25%-30% of the data to test a ML model is common practice [Theo19] [PeVG11]. Using a larger percentage of the dataset to train the model resulted in relatively fewer false predictions, at the expense of having less testing data to verify the model.

Linear Regression with the Student Model Features

In the linear regression trials, the final grade was used as the dependent variable. To predict the final grade with linear regression, the full set of students was randomly divided such that 70% of the students were used to train the model, and the remaining 30% were used to test the model. For each combination of features, the linear regression algorithm was run one hundred times and the mean absolute error in final grade prediction was averaged over those trials.

The maximum possible grade was 100, and the minimum possible grade was 0. In the dataset, the actual range of grade values was 98.7864.

The features were calculated over the full timelines. The average *mean average error* (MAE) in the final grade prediction when using only one feature at a time is shown in Table 4.8.

Table 4.8: The average MAE in the final grade prediction when using only one feature

Feature	Error (MAE)
Number of Interactions	18.54
Number of Timeline Items	17.83
Interval length mean	18.24
Interval length variance	17.81
Interval length skewness	19.14
Interval length: kurtosis	19.11
Burstiness	18.96
Assignment Hand-In Time	19.21
Fraction of Midterm Time Used	16.76
Fraction of Non-Zero Days	16.85

Of the 10 features, defined above, there are 1023 ways to choose subsets of 1 to 10 (inclusive) of those features to use in linear prediction. However, not all of the features are independent variables. Since the timeline intervals have a non-Gaussian distribution, the moments of the interval lengths can not be considered independent of each other. Similarly, since the Burstiness measure is created out of the mean and variance of the interval lengths, it also can not be considered to be independent of those features. Therefore, when choosing a subset of features for linear regression, at most only six independent features were chosen at a time, and each subset included at most one of the interval length mean, interval length variance, or burstiness measure. The higher order moments of the interval lengths, skewness and kurtosis, were excluded as they correlated less strongly with the final grade.

The average mean absolute error over 100 trials was computed for each possible subset. In each of the 100 trials, the students were divided into new randomized testing and training groups. Each trial was guaranteed to be unique by seeding the random number generator that facilitated splitting the students into the two groups[1]. The smallest mean absolute error out of these trials for each subset size is shown in Tables 4.9 - 4.11, along with the subset of features that were used in that trial.

For the subset containing the interval length mean:

[1]For all experiments in this dissertation that required splitting the students into testing and training groups, the train_test_split() method from the sklearn.model_selection library was used with shuffling enabled [PeVG11]. In the linear regression experiments, to create the 100 trials this method was seeded with values 1 to 100 (inclusive).

Table 4.9: Linear Regression with interval length mean

Error (MAE)	Num Interactions	Num Timeline Items	Interval Mean	Assignment	Midterm	Non-Zero Days
15.4098					X	X
15.3578				X	X	X
15.3937	X			X	X	X
15.4671	X		X	X	X	X
15.5694	X	X	X	X	X	X

For the subset containing the interval length variance:

Table 4.10: Linear Regression with interval length variance

Error (MAE)	Num Interactions	Num Timeline Items	Interval Variance	Assignment	Midterm	Non-Zero Days
15.4098					X	X
15.3578				X	X	X
15.3937	X			X	X	X
15.4671	X		X	X	X	X
15.5467	X	X	X	X	X	X

For the subset containing the burstiness measure:

Table 4.11: Linear Regression with burstiness measure

Error (MAE)	Num Interactions	Num Timeline Items	Burstiness	Assignment	Midterm	Non-Zero Days
15.4098					X	X
15.3578				X	X	X
15.3937	X			X	X	X
15.4608		X	X	X	X	X
15.5304	X	X	X	X	X	X

Linear Regression with the ML Learned Features

The linear regression experiments were repeated using the ML Learned Features that were introduced in Section 3.2.5. The dependent variable was the final grade, and all eight learned features were used as independent variables. The full set of students was again randomly divided into two groups: one group of students was used to train the model, and the other was used to test the model. The average MAE was calculated over 100 trials, in which each trial had a different set of students in the training and testing groups[2]

The ML learned features created from the full set of LMS data were able to predict students' final grades using linear regression with an average MAE of 15.54 over 100 trials. When the LMS data after the VW deadline was removed, and the features were re-calculated, the average MAE of the linear regression model predicting students' final grades after 100 trials was 16.85 grade points.

Linear Regression: Comparison

As a comparison, the linear regression algorithm was run 100 times on generated random values with a normal distribution and again with a uniform distribution. For both types of random values, the mean absolute error was in the range of 19.3 to 19.5.

Using linear regression with just one Student Model feature at a time to predict the final grade as a percentage resulted in a MAE of approximately 19 grade points for the features of number of interactions, interval length skewness, interval length kurtosis, burstiness, and average difference between assignment submission and due dates, as shown in Table 4.8. Therefore, using just one of these features for linear regression prediction was only slightly better than using random feature values as inputs. The MAE decreased to about 18 grade

[2] As before, the train_test_split() method from the sklearn.model_selection library was used with shuffling enabled to split the 70% of the students into a training group and the remaining 30% of students into a testing group [PeVG11]. In the 100 trials, this method was seeded with values 1 to 100 (inclusive).

points for the features of number of timeline items, interval length mean, and interval length variance. The MAE decreased further to about 17 grade points for features of fraction of midterm writing time, and fraction of non-zero days. A decrease in the MAE indicates that the predictions are more accurate.

By using multiple Student Model features as independent variables, the accuracy of the linear regression model increased to a MAE of approximately 15.4 grade points. The MAE was 15.54 when using the ML learned features created from the full set of rawe LMS data, making the results of linear prediction using either sets of features comparable.

Logistic Regression with the Student Model Features

In many instances, being able to the predict the specific value of a student's final grade is not necessary. It may be sufficient to predict the likelihood that they will pass or fail the course overall, and identify students who are at an elevated risk of failure. Logistic regression can be used to predict binary outcomes, such as passing or failing a course. In the trials using logistic regression, the dependent variable was whether a student had achieved a final grade above or below 50%, denoted as 'Passed' or not. To predict the final grade with logistic regression, the full set of students was again randomly divided such that 70% of the students were used to train the model, and the remaining 30% were used to test the model. Each of the below tables is from one run of the logistic regression model seeded the same way. That is, the students were split into the same test and training groups in each case in order to compare the effect of using different features on the predictive capabilities of the model.

Error matrices are used to present the results from the logistic regression trials. The error matrices show the total number of students whose outcomes were correctly by the model, as well as the number of false positives (students who were predicted to pass, but in

reality did not) and the number of false negative (students who were predicted to fail, but in fact passed). These values are arranged in the error matrices as follows:

$$\begin{bmatrix} \text{true negatives} & \text{false positives} \\ \text{false negatives} & \text{true positives} \end{bmatrix} \qquad \text{(Equation 4.3.1)}$$

In the first run, the independent features were the total number of interactions, total number of timeline items, fraction of non-zero days, fraction of midterm time used, and mean interval length. The classification report and accuracy score are shown in Table 4.12, and the error matrix was as follows: $\begin{bmatrix} 4 & 15 \\ 0 & 73 \end{bmatrix}$

As can be seen from the error matrix and Table 4.12, the model was able to accurately predict all 73 students in the test group who passed the course. Of the students in the test group who failed the course, four were accurately predicted while 15 were incorrectly predicted by the model to pass.

Table 4.12: Logistic Regression Feature Set 1 with Hand-crafted Features

	precision	recall	f1-score	support
0	1.00	0.21	0.35	19
1	0.83	1.00	0.91	73
accuracy			0.84	92
macro average	0.91	0.61	0.63	92
weighted average	0.86	0.84	0.79	92

In the second run, the independent features were the total number of interactions, total number of timeline items, fraction of non-zero days, fraction of midterm time used, and mean interval variance. The classification report and accuracy score are shown in Table 4.13, and

the error matrix was as follows: $\begin{bmatrix} 7 & 12 \\ 1 & 72 \end{bmatrix}$

During this run, the model was able to accurately predict 72 of 73 students in the test group who passed the course. One student was incorrectly predicted to fail, when they belonged to the passing group. Of the 19 students in the test group who failed the course, seven were accurately predicted to fail and the remaining 12 were incorrectly predicted to pass. A total of 13 students were predicted incorrectly during this run, making it more accurate overall compared to the first run which used the mean interval lengths. Even though there was one false negative, the number of true negatives was higher.

Table 4.13: Logistic Regression Feature Set 2 with Hand-crafted Features

	precision	recall	f1-score	support
0	0.88	0.37	0.52	19
1	0.86	0.99	0.92	73
accuracy			0.86	92
macro average	0.87	0.68	0.72	92
weighted average	0.86	0.86	0.83	92

In the third run, the independent features were the total number of interactions, total number of timeline items, fraction of non-zero days, fraction of midterm time used, and burstiness measure. The classification report and accuracy score are shown in Table 4.14, and the error matrix was as follows: $\begin{bmatrix} 6 & 13 \\ 1 & 72 \end{bmatrix}$

This run of the logistic regression model with the burstiness measure accurately predicted 72 of the 73 students who passed and 6 of the 19 students who failed. This means that the model made a total of 14 errors, which is one more than the previous run.

Table 4.14: Logistic Regression Feature Set 3 with Hand-crafted Features

	precision	recall	f1-score	support
0	0.86	0.32	0.46	19
1	0.85	0.99	0.91	73
accuracy			0.85	92
macro average	0.85	0.65	0.69	92
weighted average	0.85	0.85	0.82	92

In the fourth run, all interval data was removed and the independent features used were the total number of interactions, total number of timeline items, fraction of non-zero days, and fraction of midterm time used. The classification report and accuracy score are shown in Table 4.15, and the error matrix was as follows: $\begin{bmatrix} 6 & 13 \\ 1 & 72 \end{bmatrix}$

The predictions of this run had the same performance as when the burstiness measure was included. There was one false negative, and 13 false positives for a total of 14 prediction errors.

Table 4.15: Logistic Regression Feature Set 4 with Hand-crafted Features

	precision	recall	f1-score	support
0	0.86	0.32	0.46	19
1	0.85	0.99	0.91	73
accuracy			0.85	92
macro average	0.85	0.65	0.69	92
weighted average	0.85	0.85	0.82	92

In the fifth run, only the timeline features were used. Running the logistic regression model with the total number of timeline items and mean interval length produced the

classification report and accuracy score shown in Table 4.16, and the error matrix was as follows: $\begin{bmatrix} 4 & 15 \\ 0 & 73 \end{bmatrix}$

Using just the timeline features, the model accurately predicted all 73 students in the test group who passed the course. However, it was only able to correctly predict four of the 19 students who failed the course, leading to a higher number of false positives.

Table 4.16: Logistic Regression Feature Set 5 with Hand-crafted Features

	precision	recall	f1-score	support
0	1.00	0.21	0.35	19
1	0.83	1.00	0.91	73
accuracy			0.84	92
macro average	0.91	0.61	0.63	92
weighted average	0.86	0.84	0.79	92

Logistic Regression with the ML Learned Features

In the logistic regression experiments were repeated with the eight ML learned features as the independent variables. As previously, the dependent variable was whether a student had achieved a final grade above or below 50%, denoted as 'Passed' or not. The students were again split into the same training and testing groups.

The logistic regression model was first trained and tested using the eight ML learned features created from the full set of LMS data. Upon testing, the model failed to converge before it reached the limit on the number of iterations. However, the classification report and accuracy score that were produced at the maximum number of iterations are shown in

Table 4.17, and the error matrix was: $\begin{bmatrix} 14 & 14 \\ 1 & 66 \end{bmatrix}$

Table 4.17: Logistic Regression Feature Set 1 with ML Features

	precision	recall	f1-score	support
0	0.93	0.50	0.65	28
1	0.82	0.99	0.90	67
accuracy			0.84	95
macro average	0.88	0.74	0.77	95
weighted average	0.86	0.84	0.83	95

The logistic regression model was then trained and tested using the eight ML learned features created from the set LMS data up to the VW deadline. Upon testing, the model also failed to converge before it reached the limit on the number of iterations. Once again, the classification report and accuracy score that were produced at the maximum number of iterations are shown in Table 4.18, and the error matrix was: $\begin{bmatrix} 9 & 20 \\ 3 & 62 \end{bmatrix}$

Table 4.18: Logistic Regression Feature Set 2 with ML Features

	precision	recall	f1-score	support
0	0.75	0.31	0.44	29
1	0.76	0.95	0.84	65
accuracy			0.76	94
macro average	0.63	0.61	0.64	94
weighted average	0.76	0.84	0.72	94

Logistic Regression: Comparison

In each of the logistic regression trials the number of false positives was higher than then number of false negatives. The logistic regression models were able to successfully identify most, or all, of the passing students but struggled to identify the students who failed the course. When logistic regression was run using the features created from the Student models, the algorithm was able to converge, even if it did not predict all of the students accurately. However, when the ML learned features were used the algorithm was not able to converge but just stopped at the maximum number of iterations.

k-Nearest Neighbour with the Student Model Features

In the k-Nearest Neighbour trials, the dependent variable was whether a student had achieved a final grade above or below 50%, denoted as 'Passed' or not. The independent variables used were the mean interval length, number of Interactions, number of timeline items, fraction of midterm time used, burstiness, and fraction of days with at least one interaction.

The full set of students was again randomly divided such that 70% of the students were used to train the model, and the remaining 30% were used to test the model. A total of 10 students were removed from the data set due to having less than two timeline items, leaving 218 students in the training set and 94 students in the test set. Again, using a larger percentage of the dataset to train the model resulted in relatively fewer false predictions, at the expense of having less testing data to verify the model. In this context, false positives are students who were predicted to pass but failed in reality, and false negatives are students who were predicted to fail but actually passed.

The model was run using different values of k, which is the number of data points closest to the one being predicted as 'Passed' or 'Failed' that are to be used in the prediction. The

number of prediction errors for values of k between 2-10 is shown in Table 4.19. Predictions for values of k larger than 10 were similar, and predictions for values of k larger than 20 stayed the same (with 19 total errors).

Table 4.19: k-Nearest Neighbour Prediction Errors with Hand-crafted Features

k	false positives	false negatives	Total Errors
2	13	20	33
3	16	2	18
4	13	6	19
5	16	1	17
6	15	2	17
7	16	1	17
8	15	2	17
9	16	1	17
10	13	2	15

k-Nearest Neighbour with the ML Learned Features

In the k-Nearest Neighbour experiments were repeated using the ML learned features as independent variables and whether a student had achieved a final grade above or below 50%, denoted as 'Passed' or not as the dependent variable. The full set of students was again randomly divided such that 70% of the students were used to train the model, and the remaining 30% were used to test the model.

First the algorithm was run using features created from the full set of LMS data. There were nine rows in this feature set with non-numeric values that were dropped[3], leaving a

[3]If it was not possible to calculate a feature for a student, the value of that ML learned feature for the

total of 314 rows (one row per student) as input to the algorithm. The number of prediction errors for values of k between 2-10 is shown in Table 4.20.

Table 4.20: k-Nearest Neighbour Prediction Errors with ML Features

k	false positives	false negatives	Total Errors
2	9	31	40
3	16	9	25
4	15	14	29
5	17	5	22
6	16	7	23
7	18	3	21
8	17	4	21
9	18	1	19
10	18	2	20

The k-Nearest Neighbour algorithm was then run using features created from the LMS data up to the VW deadline. After dropping eleven rows in this feature set there were 312 rows as input to the algorithm. The number of prediction errors for values of k between 2-10 is shown in Table 4.21.

student was 'nan', meaning 'not a number', to indicate that the feature did not have a valid value. Non-numeric values can not be used in the prediction algorithms, so any rows containing one or more of them was dropped from the input set.

Table 4.21: k-Nearest Neighbour Early Prediction Errors with ML Features

k	false positives	false negatives	Total Errors
2	10	29	39
3	18	9	27
4	15	12	27
5	18	7	25
6	17	10	27
7	18	3	21
8	16	6	22
9	19	2	21
10	18	4	22

k-Nearest Neighbour: Comparison

In all of the k-Nearest Neighbour trials, the results when $k = 2$ produced more false negatives than false positives, as well as the highest number of overall prediction errors. For higher values of k, there were fewer overall prediction errors, but the number of false positives was greater than the number of false negatives. For each value of k, the features created from the Student Model resulted in fewer prediction errors compared to the ML learned features.

RQ 3.2: Can we define a set of archetypes (student behaviour + course outcome)?

Using the features identified in the earlier research questions, an unsupervised clustering algorithm was used to group students by their behaviour.

Clustering with the Student Model Features

Six of the Student Model features were included: number of interactions, number of timeline items, fraction of midterm time, mean interval length, burstiness, and average time to hand in assignments. A total of 16 students were excluded due to having not enough timeline items to calculated the interval mean, or no submitted assignments. This left 307 students in the data set to be clustered. After scaling the features, the dimensionality was reduced to two dimensions using PCA so that the clusters could be plotted on a 2D scatter plot for visual inspection. Although PCA can be used to determine the optimal number of meaningful reduced dimensions in a data set, it is also commonly used in ML research with clustering algorithms to reduce the number of features so that the clusters can be visualized in 2D or 3D plots [Lind20].

Each of the two principal components created by PCA are a weighted combination of the original features. The explained variance ratio for the first principal component was:

- NumInteractions (all): 0.421379
- Num Timeline Items (all): 0.489524
- Fraction of Midterm time: 0.301098
- Mean Interval Length S (all): 0.441346
- Fraction NonZero Days (all): 0.475323

The explained variance ratio for the second principal component was:

- Burstiness (all): 0.504109
- Assign Hand In Avg (all): 0.734704

After performing PCA to reduce the number of feature dimensions to two, the silhouette

coefficient was calculated to evaluate the optimal number of clusters for the two principal components. The silhouette coefficient was calculated starting at two clusters and going up to ten clusters, as shown in Figure 4.21. This coefficient is a measure of evaluating a given number of clusters based on the distance between those clusters [Scik22]. The silhouette coefficient can range from -1 to 1, with higher values indicating better clusters due to more separation between the clusters. Using the optimal number of two derived from the silhouette coefficient in Figure 4.21, the student data was then clustered into two groups as shown in Figure 4.22. Clustering is an unsupervised ML algorithm, meaning that the data is not split into training and testing groups as was the case in supervised algorithms such as Regression and k-Nearest Neighbours. Instead, the algorithm attempts to assign each data point membership into the cluster to which it is most similar using the number of clusters and features provided by the user. In this case, each data point represents one student in the data set.

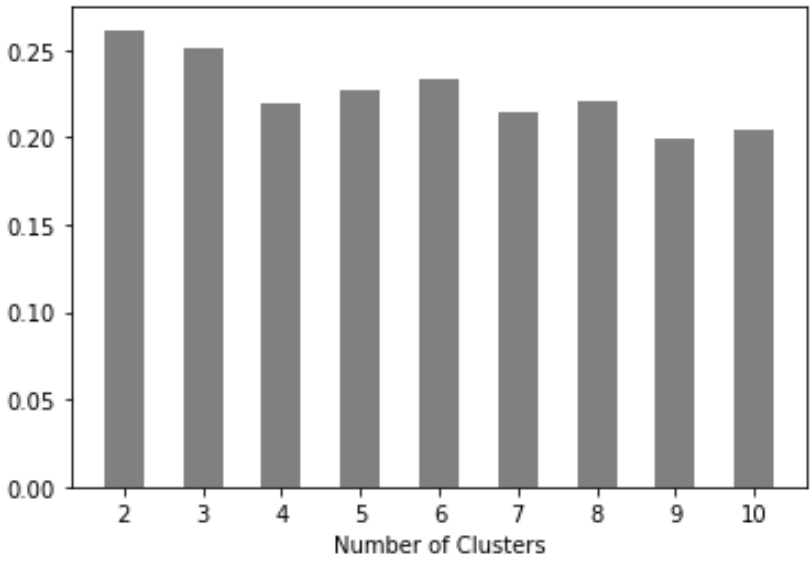

Fig. 4.21: Silhouette coefficient to determine the optimal number of clusters between 2 and 10 with Hand-crafted Features

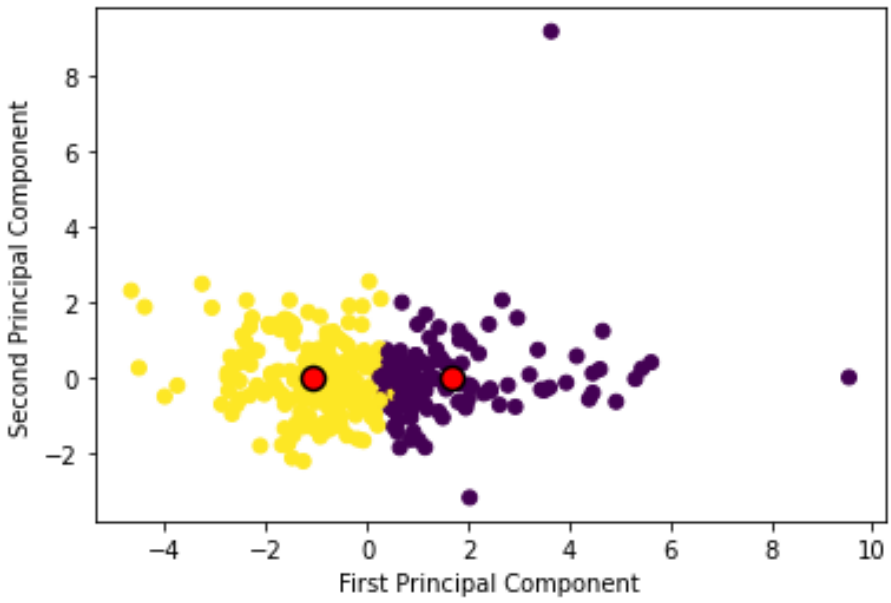

Fig. 4.22: Student clusters (unsupervised) using Hand-crafted Features

Student course outcomes, in terms of passing or failing, for each cluster is shown in Table 4.22. The yellow cluster contains the majority of the students with a course outcome 'Passed'. Of the 224 students who passed, 160 were included in the yellow cluster. Of the 83 students who failed, 56 were included in the purple cluster. If the yellow cluster is categorized as the group of students predicted to pass, and the purple cluster is considered the group of students predicted to fail, 91 students will be misclassified. However, of these, only 27 would be students who were incorrectly predicted to pass. The other 64 students were incorrectly predicted to fail.

Table 4.22: The number of students who passed or failed the course per cluster in Figure 4.22

	# Passed	# Failed	Total Students in Group
Yellow Cluster	160	27	187
Purple Cluster	64	56	120
All Students	224	83	307

Clustering with the ML Learned Features

Unsupervised clustering was repeated using the eight ML learned features created from the full set of LMS timeline data. Nine students were removed from the dataset for having non-numeric feature values. As before, PCA was used to reduce the number of dimensions to two, to allow the clusters to be meaningfully represented on the page for visual inspection. Using the two principal components created from PCA, a silhouette analysis was again performed to determine the optimal number of clusters. From Figure 4.23, it can be seen that the optimal number of clusters was two. The clusters are depicted in yellow and purple in Figure 4.24, with the centre of each cluster shown as a red circle.

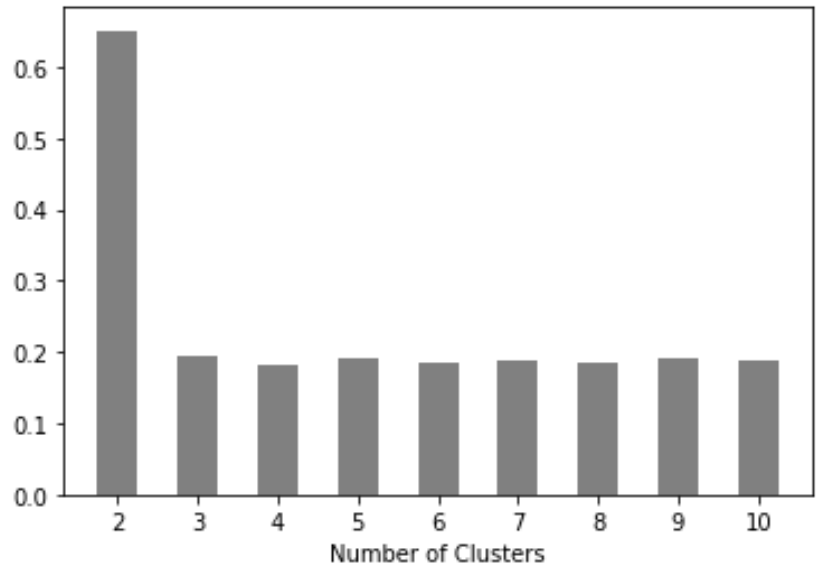

Fig. 4.23: Silhouette coefficient to determine the optimal number of clusters between 2 and 10 with ML Features

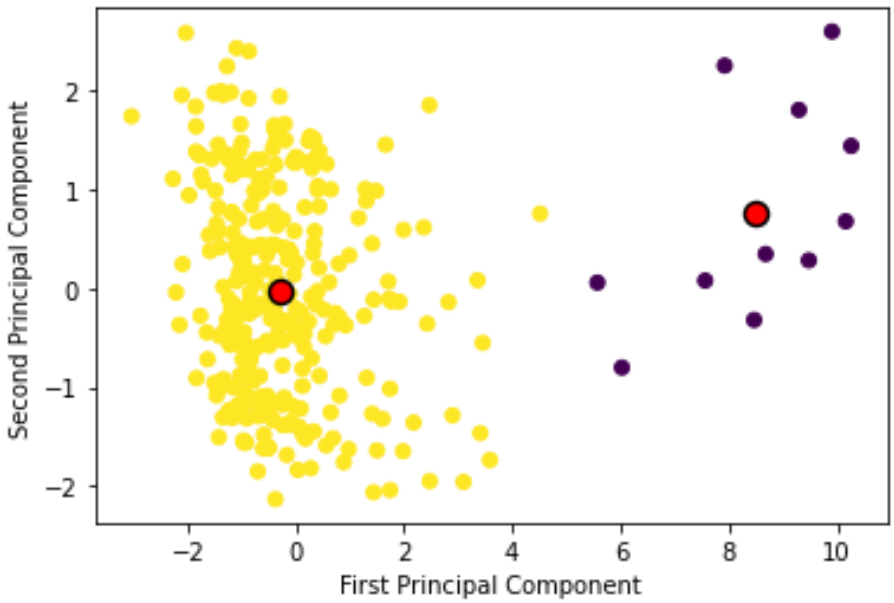

Fig. 4.24: Student clusters (unsupervised) with ML Features

Student course outcomes, in terms of passing or failing, for each cluster is shown in

Table 4.23. The yellow cluster contains the majority of all students. All students who passed the courses, as well as 79 of 90 students who failed the course, are grouped into the yellow cluster. The purple cluster contains only 11 students, all of whom failed the course. Therefore, if the yellow cluster is defined as the group of students expected to pass and the purple cluster is the group of students expected to fail, then there would be 79 false positives and zero false negatives.

Table 4.23: The number of students who passed or failed the course per cluster in Figure 4.24

	# Passed	# Failed	Total Students in Group
Yellow Cluster	224	79	303
Purple Cluster	0	11	11
All Students	224	90	314

The unsupervised clustering was repeated using the eight ML learned features created from the LMS timeline data up to the VW deadline. After removing eleven students for having non-numeric feature values, the features were then reduced using PCA to two dimensions. A silhouette analysis was performed to determine the optimal number of clusters using these two principal components, which was found to again be two as shown in Figure 4.25. The clusters are depicted in yellow and purple in Figure 4.26, with the centre of each cluster denoted by a red circle.

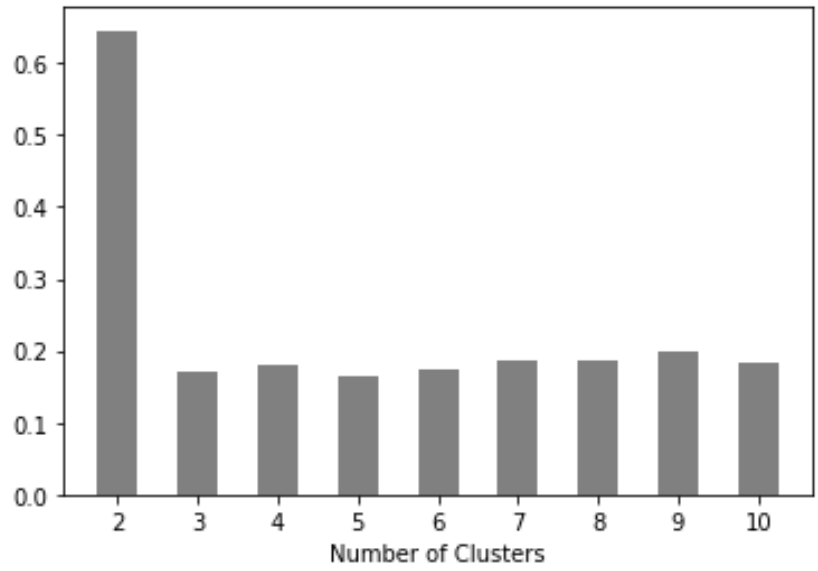

Fig. 4.25: Silhouette coefficient to determine the optimal number of clusters between 2 and 10 with ML Features up to the VW Deadline

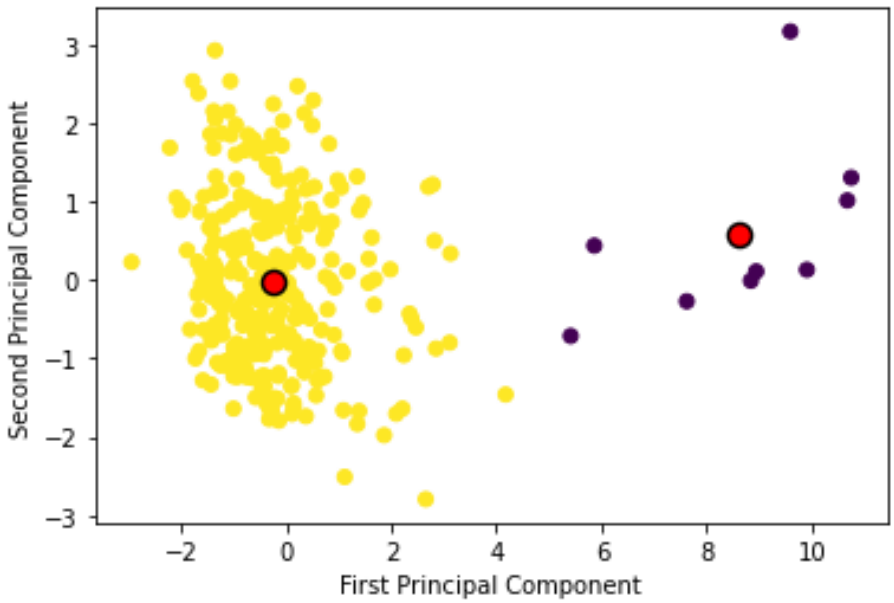

Fig. 4.26: Student clusters (unsupervised) with ML Features up to the VW Deadline

The correspondence between student course outcomes and each of the two clusters is

shown in Table 4.24. As with the clusters created from the ML learned features over the full set of LMSdata, the yellow cluster contain the majority of all students. The yellow cluster contains all students who passed the courses and 79 of the 88 students who failed the course. The purple cluster consists only of students failed the course, but only 9 of them. Therefore, if the yellow cluster is defined as the group of students expected to pass and the purple cluster is the group of students expected to fail, then there would be 79 false positives and zero false negatives.

Table 4.24: The number of students who passed or failed the course per cluster with ML Features up to the VW Deadline

	# Passed	# Failed	Total Students in Group
Yellow Cluster	224	79	303
Purple Cluster	0	9	9
All Students	224	88	312

Clustering: Comparison

The clusters created from both the Student Model features and the ML learned features had some errors when mapping the clusters to passing and failing groups. However, the Student Model features created two more equally sized clusters, compared to the ML learned features which created one large and one small cluster. The number of misclassified students overall was higher in the clusters created from the Student Model features, with a total of 91 students given the wrong group membership. Of these, 27 were students grouped into the passing cluster who had actually failed (false positives). In comparison, while the clusters created from the ML learned features mislabelled fewer students overall, all 79 of the errors were false positives.

RQ 3.3: How early within the term can these factors predict final grades?

The ML prediction methods employed in RQ 3.1 were used again. However, this time they were provided with only a subset of the timeline data.

Linear Regression In the linear regression trials, the full set of students was randomly divided such that 70% of the students were used to train the model, and the remaining 30% were used to test the model. As before, for each combination of independent features, the linear regression algorithm was run one hundred times and the mean absolute error in final grade prediction was averaged over those trials. The dependent feature was the final grade.

When the features were calculated at each quarter of the timelines, the average MAE in the final grade prediction when using only one feature at a time is shown in Table 4.25. The feature of the fraction of midterm writing time used was omitted, as this feature does not change over the term.

Table 4.25: MAE of each feature calculated at each quarter of the timeline

Feature	1st Quarter	2nd Quarter	3rd Quarter
Number of Interactions	18.9327	18.8767	18.5517
Number of Timeline Items	18.7414	18.6308	17.9725
Interval length mean	18.5506	18.4449	18.2003
Interval length variance	19.1915	18.4953	17.8150
Interval length skewness	19.2487	18.9059	19.0729
Interval length kurtosis	19.2636	18.7403	18.9797
Burstiness	19.0283	19.2092	18.9995
Assignment Hand-In Time	19.1950	19.2169	19.2625
Fraction of Non-Zero Days	18.6482	18.3132	16.9079

As shown in Table 4.25, the MAE using only one feature at a time was close to 19 grade points, making the predictions similar to random guesses. The predictions improved slightly by the three-quarter point in the term for the features of the number of timeline items and interval length variance.

The average mean absolute error over 100 trials was computed for each possible subset of the features above, at each quarter point during the term. The smallest mean absolute error out of these trials for each subset size is shown in Tables 4.26 - 4.28, along with the subset of features that were used in that trial. For trials conducted using only the first quarter of the timeline, the feature of the fraction of midterm writing time used was excluded as this feature would not be available at that point in the term.

In the subset containing the interval length mean, using multiple features improved the predictive ability of the model, as shown in Table 4.26. Using two or more features over three-quarters of the timeline, the predictions improved to an MAE of roughly 15.5 grade points.

Table 4.26: RQ 3.3: Linear Regression with interval length mean

Number of Features	**Error at 1st Quarter**	**Error at 2nd Quarter**	**Error at 3rd Quarter**
2	18.4711 (Interval length mean; Non-Zero days)	16.3409 (Fraction of Midterm time; Non-Zero days)	15.5042 (Fraction of Midterm; Non-Zero days)
3	18.5172 (Interval length mean; Non-Zero days; Num Interactions)	16.3645 (Assign Hand In; Non-Zero days; Fraction of Midterm)	15.5235 (Assign Hand In; Non-Zero days; Fraction of Midterm)
4	18.5983 (Interval length mean; Non-Zero days; Num Interactions; Num Timeline Items)	16.4328 (Interval length mean; Non-zero days; Assign Hand In; Fraction of Midterm)	15.5801 (Num Interactions; Assign Hand In; Fraction of Midterm; Non-Zero days)
5	18.7157 (Interval length mean; Non-Zero days; Num Interactions; Num Timeline Items; Assign Hand In)	16.5234 (Non-Zero days; Num Interactions; Num Timeline Items; Assign Hand In; Fraction of Midterm)	15.6430 (Num Interactions; Num Timeline Items; Assign Hand In; Fraction of Midterm; Non-Zero days)
6	n\a	16.6563 (all features)	15.7474 (all features)

In the subset containing the interval length variance, using multiple features also im-

proved the predictive ability of the model, as shown in Table 4.27. Using two or more features over three-quarters of the timeline, the predictions improved to an MAE of roughly 15.5 grade points which was similar to the predictions in which interval length variance was included as a feature.

Table 4.27: RQ 3.3: Linear Regression with interval length variance

Number of Features	**Error at 1st Quarter**	**Error at 2nd Quarter**	**Error at 3rd Quarter**
2	18.6837 (Num Timeline Items; Non-Zero days)	16.3409 (Fraction of Midterm; Non-Zero days)	15.5042 (Fraction of Midterm; Non-Zero days)
3	18.7076 (Num Timeline Items; Non-Zero days; Num Interactions)	16.3645 (Assign Hand In; Non-Zero days; Fraction of Midterm)	15.5235 (Assign Hand In; Non-Zero days; Fraction of Midterm)
4	18.8256 (Assign Hand In; Non-Zero days; Num Interactions; Num Timeline Items)	16.4494 (Num Interactions; Non-zero days; Assign Hand In; Fraction of Midterm)	15.5801 (Num Interactions; Assign Hand In; Fraction of Midterm; Non-Zero days)

Continued on next page

Table 4.27 – continued from previous page

Number of Features	**Error at 1st Quarter**	**Error at 2nd Quarter**	**Error at 3rd Quarter**
5	18.9567 (Interval length variance; Non-Zero days; Num Interactions; Num Timeline Items; Assign Hand In)	16.5234 (Non-Zero days; Num Interactions; Num Timeline Items; Assign Hand In; Fraction of Midterm)	15.6430 (Num Interactions; Num Timeline Items; Assign Hand In; Fraction of Midterm; Non-Zero days)
6	n\a	16.6473 (all features)	15.7187 (all features)

In the subset containing the burstiness measure, using multiple features improved the predictive ability of the model, as shown in Table 4.28. Using two or more features over three-quarters of the timeline, the predictions improved to an MAE of roughly 15.3 grade points which was slightly better than when either the interval length mean or variance was included.

Table 4.28: RQ 3.3: Linear Regression with burstiness

Number of Features	**Error at 1st Quarter**	**Error at 2nd Quarter**	**Error at 3rd Quarter**
2	18.6837 (Num Timeline Items; Non-Zero days)	16.3409 (Fraction of Midterm; Non-Zero days)	15.5042 (Fraction of Midterm; Non-Zero days)
3	18.7076 (Num Timeline Items; Non-Zero days; Num Interactions)	16.2146 (Burstiness; Non-Zero days; Fraction of Midterm)	15.3078 (Burstiness; Non-Zero days; Fraction of Midterm)
4	18.8177 (Burstiness; Non-Zero days; Num Interactions; Num Timeline Items)	16.2812 (Burstiness; Non-zero days; Assign Hand In; Fraction of Midterm)	15.3555 (Burstiness; Num Interactions; Fraction of Midterm; Non-Zero days)
5	18.9364 (Burstiness; Non-Zero days; Num Interactions; Num Timeline Items; Assign Hand In)	16.3717 (Burstiness; Non-Zero days; Num Interactions; Num Timeline Items; Fraction of Midterm)	15.4162 (Burstiness; Interactions; Num Timeline Items; Fraction of Midterm; Non-Zero days)
6	n\a	16.4504 (all features)	15.5198 (all features)

RQ 3.4: Can these variables and features be used to predict course outcomes of pass or fail before the end of the term, to indicate when intervention may be required?

As shown in Tables 4.26 - 4.28, the predictions at the quarter-point during the term using linear regression had a MAE of approximately 18.5 to 19 grade points, making these predictions little better than random. At the half way point in the term the predictions improve to a minimum MAE of 16.3 grade points, depending on the features used. By three-quarters of the way through the term the predictions again improve to a minimum MAE of 15.3 grade points, which is similar to the prediction error over the whole timeline. Therefore, it is possible to predict final grades before the end of the term with greater than random accuracy.

4.4 Group 4 RQs: Time Series Classification

The results of student course outcome prediction with binary and ternary CNN and transformer models trained on the full timelines are now provided. These experiments address research questions 4.1 and 4.2. Subsections 4.4.1 to 4.4.3 report on the accuracy of the CNN models, while subsections 4.4.4 to 4.4.6 present the results from the transformer models.

4.4.1 CNN Binary Classifiers

CNN binary classifiers were used to predict whether students passed or failed the course. For each type of input data, Table 4.29 shows the average training and and testing accuracies as percentages after running the model 100 times with different random starting seeds. The average number of epochs over the 100 trials is also provided in the table. When the CNN

binary classifier was provided with interval data as input, the average prediction accuracy of the test set was 74.51% and the average number of epochs was 171. Using the timestamp data (the one-dimensional timestamp data as shown in Table 3.6) as input instead, the average prediction accuracy of the test set increased to 82.10% and the average number of epochs also increased to 286. However, of the three data types, the multivariable data produced the most accurate results in the CNN binary classifier.

Table 4.29: Average values for CNN binary classifier (entire term)

Type of Data	Train (%)	Test (%)	Epochs
Intervals	74.10	74.51	171
Timestamp	81.88	82.10	286
Multivariable	89.53	86.35	310

Tables 4.30 to 4.32 show, for each input type of data provided to the CNN binary classifier, the average number of students who: actually failed and were predicted to fail (true negatives), actually failed but were predicted to pass (false positives), actually passed but were predicted to fail (false negatives), and actually passed and were predicted to pass (true positives). The interval, timestamp, and multivariable data all produced approximately the same average number of true positives and false negatives. That is, the CNN binary classifier was able to correctly identify students who passed the course with roughly the same accuracy regardless of the format of the input data.

However, the format of the input data impacted the ability of the CNN binary classifier to identify students who failed the course - arguably the group of students that are most important to identify. As shown in Table 4.30, when using the interval data the CNN binary classifier correctly identified only roughly 17% of the failing students in the test group (true negatives) on average. The CNN binary classifier was better able to identify

failing students using the timestamp data. On average the CNN binary classifier correctly identified approximately 47% and 67% of the students who failed using the timestamp data and multivariable data, respectively.

Table 4.30: Average error matrix for CNN binary classifier with Intervals data (entire term)

	Predicted Fail	Predicted Pass
Actual Fail	3	15
Actual Pass	2	43

Table 4.31: Average error matrix for CNN binary classifier with Timestamp data (entire term)

	Predicted Fail	Predicted Pass
Actual Fail	8	9
Actual Pass	4	42

Table 4.32: Average error matrix for CNN binary classifier with Multivariable data (entire term)

	Predicted Fail	Predicted Pass
Actual Fail	12	6
Actual Pass	3	42

In order to better compare how the format of the input data impacts the predictions of the CNN binary classifier, the statistics fo the best predictive model using each type of input data is shown in Table 4.33. The best model was defined as the model with the highest testing accuracy.For each type of input data, the best model performed approximately 10-15% better than the average shown in Table 4.29. The best predictive model had a test accuracy of 88.89% using intervals as input, 90.48% using timestamp data, and 98.41% using multivariable data.

Table 4.33: Best values for CNN binary classifier (entire term)

Type of Data	Train (%)	Test (%)	Epochs
Intervals	78.75	88.89	167
Timestamp	82.30	90.48	467
Multivariable	86.08	98.41	157

Tables 4.34 to 4.36 show the error matrices that correspond to the best CNN binary classifier model produced by each of the interval, timestamp, and multivariable input data. In each case the model was able to accurately predict the majority or all of the passing student correctly. The models were also able to predict the majority of failing students correctly, although the number of false positives was relatively higher than the number of false negatives in each case.

Table 4.34: Best error matrix for CNN binary classifier with Intervals data (entire term)

	Predicted Fail	Predicted Pass
Actual Fail	11	4
Actual Pass	3	45

Table 4.35: Best error matrix for CNN binary classifier with Timestamp data (entire term)

	Predicted Fail	Predicted Pass
Actual Fail	10	6
Actual Pass	0	47

Table 4.36: Best error matrix for CNN binary classifier with Multivariable data (entire term)

	Predicted Fail	Predicted Pass
Actual Fail	11	1
Actual Pass	0	51

4.4.2 CNN ternary classifiers: Letter Grades

In addition to predicting students who will fail a course, it would also be useful to be able to predict students who are at risk of failing the course. Ternary CNN classifiers were trained to predict whether students were members of one of three groups, based on final letter grades. The first group consisted of students who achieved a high passing grade defined as $\geq 70\%$ (a grade of B or higher); the second group consisted of students at risk of failing (or "warning" group), defined as achieving a low passing grade of $\geq 50\%$ & $< 70\%$ (a grade of C+, C, or D); and the third group consisted of students who failed the course with a final grade of $< 50\%$ (a grade of F). Table 4.37 shows the average training and testing accuracies, as well as average number of epochs, over 100 trials of CNN ternary classifiers with different random seeds.

As with the CNN binary classifiers, the format of the input data impacted the accuracy of the results. Using interval data as input, the CNN ternary classifiers achieved a 46.52% accuracy on average. This increased to an average of 57.33% when using the timestamp data as input, and increased further to an average of 64.57% when the models were trained on the multivariable data.

Table 4.37: Average values for CNN ternary classifier based on letter grades (entire term)

Type of Data	Train (%)	Test (%)	Epochs
Intervals	47.74	46.52	169
Timestamp	58.99	57.33	300
Multivariable	73.36	64.57	303

The average numbers of students who were correctly and incorrectly classified into each of the three classification groups, using each of the three types of input data, are shown in Tables 4.38 to 4.40. In these tables, each row contains the students who were actually in

each of the three groups of failing, warning, or passing. The columns indicate the average predictions of the CNN ternary classifiers. For example, in 4.38, the first row shows that of the students who actually failed, 5 were predicted to fail, 1 was predicted to be in the "warning" group, and 12 were predicted to pass, on average.

For all three types of input data, the CNN ternary classifier was able to correctly predict the majority of the passing students. However, the CNN ternary classifiers struggled to identify students in the "warning" group, predicting that many of those students would pass (on average: using the intervals, timestamp, and multivariable data, the models placed 78%, 61%, and 53%, respectively, of the students who should be in the "warning" group in the passing group). Given that the students in the "warning" group did indeed pass, this is not an altogether surprising result. Though it does mean that these students would not be identified by the models for additional course support. The CNN ternary classifiers on average identified 33% of the students who actually failed as being in either the failing or "warning" groups when using the interval data. Using the timestamp data, the average number of students who actually failed and were predicted to be in either of those two groups improved at 77%. With the multivariable data, the CNN ternary classifiers labelled 82% of the students who actually failed as being part of the failing or "warning" groups.

Table 4.38: Average error matrix for CNN ternary classifier based on letter grades with Intervals data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	5	1	12
Actual Warning	3	1	14
Actual Pass	4	1	22

Table 4.39: Average error matrix for CNN ternary classifier based on letter grades with Timestamp data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	10	2	5
Actual Warning	4	4	11
Actual Pass	2	4	20

Table 4.40: Average error matrix for CNN ternary classifier based on letter grades with Multivariable data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	12	2	3
Actual Warning	3	6	10
Actual Pass	1	4	22

The CNN ternary classifiers, with the group divisions as described above, that produced the best results are shown in Table 4.41. The model, for each input type of data, with the highest testing accuracy was considered the best. As shown in the table, the best model produced by both the intervals and timestamp data had a testing accuracy of 68.25%, although the number of epochs was different. By including the additional elements in the multivariable data, the best CNN ternary classifier model achieved an accuracy of 82.54%.

Table 4.41: Best values for CNN ternary classifier based on letter grades (entire term)

Type of Data	**Train (%)**	**Test (%)**	**Epochs**
Intervals	61.06	68.25	254
Timestamp	56.64	68.25	419
Multivariable	71.34	82.54	294

The error matrices for each of the best CNN ternary classifiers using each input type

of data are shown in Tables 4.42 to 4.44. In all cases, the majority of passing students were predicted correctly. As well, in all cases the majority of students in the failing group were correctly predicted to fail.

However, the type of input data influenced the ability of the CNN ternary classifier to predict students in the "warning" group. While neither of the best CNN ternary classifiers trained on the interval or timestamp data correctly identified many of the students in the "warning" group (only 15.4% and 37.5% of those students were classified correctly, respectively), the best model trained on the multivariable data correctly identified 62.5% of the students in the "warning" group. Alternatively, we may only be interested in whether the students who are actually in the "warning" group would be identified as requiring help (i.e. predicted to be in either of the "warning" or failing groups), rather than requiring that they be accurately classified in the "warning" group itself. In that case, then the best CNN ternary classifiers achieved that goal with 38.5%, 43.8%, and 75% accuracy with the interval, timestamp, and multivariable data, respectively.

Table 4.42: Best error matrix for CNN ternary classifier based on letter grades with Intervals data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	11	1	8
Actual Warning	3	2	8
Actual Pass	0	0	30

Table 4.43: Best error matrix for CNN ternary classifier based on letter grades with Timestamp data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	12	4	3
Actual Warning	1	6	9
Actual Pass	0	3	25

Table 4.44: Best error matrix for CNN ternary classifier based on letter grades with Multivariable data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	17	1	1
Actual Warning	2	10	4
Actual Pass	0	3	25

4.4.3 CNN ternary classifiers: Median Grade

The CNN ternary classifier experiments were repeated with a different set of classification groups. In these experiments, the group of students who passed the course was divided in half. The students with the top half of the passing marks were labelled as the "passing" group, and the other half of the students were labelled as the "warning" group. The "failing" group consisted of all students who failed the course. Table 4.45 shows the average training and testing accuracies, as well as the average number of epochs, over 100 trials of CNN ternary classifiers with different random seeds under this paradigm.

The type of input data affected the testing accuracy of these models. The interval data, on average, performed the worst with a testing accuracy of 40.38%. The timestamp data produced better results, with average testing accuracies of 52.30% and 61.95% for the timestamp and the multivariable data respectively.

Table 4.45: Average values for CNN ternary classifier based on median passing grade (entire term)

Type of Data	**Train (%)**	**Test (%)**	**Epochs**
Intervals	43.65	40.38	169
Timestamp	55.56	52.30	274
Multivariable	73.26	61.95	310

The average error matrices for the CNN ternary classifiers with the second type of classification groups are shown in Tables 4.46 to 4.48. As shown in these tables, the interval data resulted in many of the students being misclassified across all three groups. When using either of the timestamp data sets as input, the CNN classifiers were able to accurately classify the majority of the failing students. However, these models still had difficulty distinguishing between students in the "warning" and passing groups.

Table 4.46: Average error matrix for CNN ternary classifier based on median passing grade with Intervals data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	4	7	6
Actual Warning	3	10	11
Actual Pass	3	9	10

Table 4.47: Average error matrix for CNN ternary classifier based on median passing grade with Timestamp data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	10	5	3
Actual Warning	4	10	9
Actual Pass	2	9	11

Table 4.48: Average error matrix for CNN ternary classifier based on median passing grade with Multivariable data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	12	3	2
Actual Warning	3	12	8
Actual Pass	1	7	14

The CNN ternary classifiers, with the group divisions as described above, that produced

the highest testing accuracy are shown in Table 4.49. As shown in the table, the best model produced by intervals, timestamp data, and multivariable data, had a testing accuracy of 60.32%, 68.25%, and 76.19%, respectively.

Table 4.49: Best values for CNN ternary classifier based on median passing grade (entire term)

Type of Data	**Train (%)**	**Test (%)**	**Epochs**
Intervals	59.74	60.32	164
Timestamp	57.30	68.25	500
Multivariable	75.81	76.19	423

Tables 4.50 to 4.52 show the error matrices that correspond to each of the best CNN ternary classifier models in Table 4.49. For each type of input data, the best model correctly predicted half or more of the students in each group, as indicated along the diagonal of each error matrix.

Table 4.50: Best error matrix for CNN ternary classifier based on median passing grade with Intervals data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	8	4	4
Actual Warning	4	10	6
Actual Pass	0	7	20

Table 4.51: Best error matrix for CNN ternary classifier based on median passing grade with Timestamp data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	12	5	1
Actual Warning	0	14	7
Actual Pass	1	6	17

Table 4.52: Best error matrix for CNN ternary classifier based on median passing grade with Multivariable data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	9	6	3
Actual Warning	1	23	1
Actual Pass	0	5	15

4.4.4 Transformer Binary Classifiers

The experiments described above with CNN classifiers were repeated with transformers, which are a relatively newer type of ML. Like CNNs and other neural network types of ML, transformers produce prediction results without an explanation of why or how the model arrived at those predictions. The transformer models were more computationally intensive than the CNN models, and therefore fewer iterations with different random seeds were run.

Table 4.53 shows the average training and testing accuracy, as well as average number of epochs, for the transformer binary classifier over 20 iterations. As shown in the table, on average the model was less accurate using the interval data compared to the timestamp or multivariable data (70.00% compared to 78.89% and 79.37%, respectively), but required more epochs.

Table 4.53: Average values for transformer binary classifier (entire term)

Type of Data	**Train (%)**	**Test (%)**	**Epochs**
Intervals	81.52	70.00	247
Timestamp	82.68	78.89	80
Multivariable	83.54	79.37	101

Tables 4.54 to 4.56 show the average number of students who were correctly predicted to pass or fail, for each type of input data provided to the transformer binary classifier. The tables include the average number of students who: actually failed and were predicted to fail (true negatives), actually failed but were predicted to pass (false positives), actually passed but were predicted to fail (false negatives), and actually passed and were predicted to pass (true positives).

The transformer binary classifiers were able to correctly predict the majority of students who passed the course, regardless of the type of input data. However, the classifiers were less successful at classifying students who had failed the course. On average, only 17.6% of students who failed the course were predicted to fail (true negatives) when the interval data was used. With timestamp and multivariable data 35% and 44% of students who failed were correctly classified, respectively.

Table 4.54: Average error matrix for transformer binary classifier with Intervals data (entire term)

	Predicted Fail	Predicted Pass
Actual Fail	3	14
Actual Pass	5	40

Table 4.55: Average error matrix for transformer binary classifier with Timestamp data (entire term)

	Predicted Fail	Predicted Pass
Actual Fail	6	11
Actual Pass	4	42

Table 4.56: Average error matrix for transformer binary classifier with Multivariable data (entire term)

	Predicted Fail	Predicted Pass
Actual Fail	8	10
Actual Pass	8	37

As shown in Table 4.57, the transformer binary classifier was capable of producing highly accurate predictions, depending on how it was initialized with the random seed. The best transformer binary classifier model, with the highest testing accuracy, for each type of input data is included in the table. Using interval data, the transformer binary classifier achieved a testing accuracy of 82.54%. The transformer binary classifier models were more successful with the timestamp data, achieving a best testing accuracy of 88.89% and 90.48% with the timestamp and multivariable data respectively.

Table 4.57: Best values for transformer binary classifier (entire term)

Type of Data	**Train (%)**	**Test (%)**	**Epochs**
Intervals	97.21	82.54	78
Timestamp	80.73	88.89	77
Multivariable	87.14	90.48	91

The best transformer binary classifier models produced in these experiments were able to correctly predict the majority of passing students, as shown in Tables 4.58 to 4.60. In fact, the best models produced with either the timestamp or the multivariable data correctly predicted all students in the passing group. However, the models were less successful at classifying the students who failed the course, leading to a higher percentage of false positives compared to false negatives.

Table 4.58: Best error matrix for transformer binary classifier with Intervals data (entire term)

	Predicted Fail	Predicted Pass
Actual Fail	5	8
Actual Pass	3	47

Table 4.59: Best error matrix for transformer binary classifier with Timestamp data (entire term)

	Predicted Fail	Predicted Pass
Actual Fail	6	7
Actual Pass	0	50

Table 4.60: Best error matrix for transformer binary classifier with Multivariable data (entire term)

	Predicted Fail	Predicted Pass
Actual Fail	10	6
Actual Pass	0	47

4.4.5 Transformer ternary classifiers: Letter Grades

Ternary transformer classifiers were trained to predict whether students were members of one of three groups, based on final letter grades. The first group consisted of students who achieved a high passing grade defined as $\geq 70\%$ (a grade of B or higher); the second group consisted of students at risk of failing (or "warning" group), defined as achieving a low passing grade of $\geq 50\%$ & $< 70\%$ (a grade of C+, C, or D); and the third group consisted of students who failed the course with a final grade of $< 50\%$ (a grade of F). Table 4.61 shows the average training and testing accuracies, as well as average number of epochs, over 20 trials of transformer ternary classifiers with different random seeds.

Using interval data as input, the transformer ternary classifiers achieved a 42.22% accuracy on average. This increased to an average of 48.02% and 49.84% when using the

timestamp and multivariable data as input, respectively.

Table 4.61: Average values for transformer ternary classifier based on letter grades (entire term)

Type of Data	Train (%)	Test (%)	Epochs
Intervals	63.57	42.22	176
Timestamp	64.90	48.02	80
Multivariable	66.43	49.84	90

The average numbers of students who were correctly and incorrectly classified into each of the three classification groups by the transformer ternary classifiers, using each of the three types of input data, are shown in Tables 4.62 to 4.64. Each row contains the students who were actually in each of the three groups of failing, warning, or passing. The columns indicate the average predictions of the transformer ternary classifiers.

Using the interval data, the transformer ternary classifiers predicted that that majority of students would pass, no matter which group they actually belonged to, on average. This is demonstrated in the third column of Table 4.62. With the timestamp data as input, the transformer ternary classifiers were typically able to identify more of the students who failed. However, the majority of students in the "warning" group were still predicted to be in the passing group instead.

Table 4.62: Average confusion matrix for transformer ternary classifier based on letter grades with Intervals data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	4	1	13
Actual Warning	2	2	14
Actual Pass	4	3	20

Table 4.63: Average confusion matrix for transformer ternary classifier based on letter grades with Timestamp data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	7	5	6
Actual Warning	3	4	11
Actual Pass	3	7	18

Table 4.64: Average confusion matrix for transformer ternary classifier based on letter grades with Multivariable data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	9	3	5
Actual Warning	5	3	11
Actual Pass	6	5	17

The training and testing accuracy of the best transformer ternary classifiers with the groups described above, for each type of input data, are shown in Table 4.65. A determination of the best model was made based on the highest testing accuracy. The accuracy of the best model increased depending on the type of input data, with the interval data producing an accuracy of 50.79%, and the timestamp and multivariable data producing models with accuracies of 55.56% and 58.73%, respectively. The number of epochs used by each model is also shown in Table 4.65. For all three of the best transformer ternary classifier, the number of epochs was less than one hundred.

Table 4.65: Best values for transformer ternary classifier based on letter grades (entire term)

Type of Data	**Train (%)**	**Test (%)**	**Epochs**
Intervals	41.04	50.79	76
Timestamp	62.72	55.56	77
Multivariable	73.97	58.73	91

The error matrices for each of the best transformer ternary classifiers using each input type of data are shown in Tables 4.66 to 4.68. The type of input data influenced the ability of the transformer ternary classifier to predict students in each of the the three groups. Using the interval data, the best transformer ternary classifier result of 50.79% testing accuracy was achieved by unhelpfully predicting that *all* students would pass the course.

The models produced using the timestamp data were better able to identify students in the failing and "warning" groups. The best model using the timestamp data correctly predicted 41% of the students who actually failed. The model also predicted that 82% of the students who failed would be in either the failing or "warning" groups, meaning that the majority of those students could be identified for further support. The same model correctly predicted 47% of students in the "warning" group. Of the students who were actually in the "warning group", 42% were predicted to pass.

Finally, the best model using the multivariable data correctly predicted 42% of students who failed, and identified 58% of that group as either failing or in the "warning" category. The model was less successful at identifying students that were truly in the "warning" group, correctly predicting of 7% of those students. Of the students who were actually in the "warning" group, 93% were predicted by this model to pass the course.

Table 4.66: Best confusion matrix for transformer ternary classifier based on letter grades with Intervals data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	0	0	13
Actual Warning	0	0	18
Actual Pass	0	0	32

Table 4.67: Best confusion matrix for transformer ternary classifier based on letter grades with Timestamp data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	7	7	3
Actual Warning	2	9	8
Actual Pass	0	8	19

Table 4.68: Best confusion matrix for transformer ternary classifier based on letter grades with Multivariable data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	8	3	8
Actual Warning	0	1	13
Actual Pass	1	1	28

4.4.6 Transformer ternary classifiers: Median Grade

The transformer ternary classifier experiments were also repeated with a different set of classification groups. In the following experiments, the group of students who passed the course was divided in half. The students with the top half of the passing marks were labelled as the "passing" group, and the other half of the students were labelled as the "warning" group. The "failing" group consisted of all students who failed the course. Table 4.69 shows the average training and testing accuracies and the average number of epochs over 20 trials of transformer ternary classifiers with different random seeds.

The transformer ternary classifiers in these experiments were sensitive to the format of the input data. On average, the classifiers trained on the interval data performed the worst with a testing accuracy of 37.54%. The classifiers trained on the timestamp data produced slightly better results, with average testing accuracies of 48.97% and 48.49% for the timestamp and the multivariable data, respectively.

Table 4.69: Average values for transformer ternary classifier based on median passing grade (entire term)

Type of Data	Train (%)	Test (%)	Epochs
Intervals	60.25	37.54	204
Timestamp	66.21	48.97	82
Multivariable	68.43	48.49	87

Tables 4.70 to 4.72 show the average numbers of students who were correctly and incorrectly classified into each of the three classification groups by the transformer ternary classifiers, using each of the three types of input data. Each row contains the students who were actually in each of the three groups of failing, warning, or passing. The columns indicate the average predictions of the transformer ternary classifiers.

The type of input data influenced the predictive results of the transformer ternary classifiers. Using the interval data as input, the transformer ternary classifiers were not able to correctly classify the majority of students. Of the students who failed, only 24% were predicted to do so. However, 82% of the students who failed were, on average, predicted to be part of the failing or "warning" groups, meaning they would be identified for additional support. On average, the classifiers trained on the interval data did a good job identifying students in the "warning" group, correctly classifying 61% of these students. However, only 28% of students in the "passing" group were classified correctly.

The transformer ternary classifiers trained on the timestamp data produced fewer false negatives on average. Using the timestamp data, the classifiers correctly predicted 35% of failing students (and classified 76% of the failing students as either "failing" or "warning"). In the warning group, 48% of students were classified correctly (and 65% were classified as either "failing" or "warning"), and 50% of the passing group was classified correctly on average. When the models were trained on the multivariable data, 59% of failing students were classified correctly (and 82% of the failing students were classified as either "failing"

or "warning"). About 30% of student actually in the "warning" group were classified as such, but 61% of students in that group were classified as one of "failing" or "warning", and 43% of students in the "passing" group were correctly predicted.

Table 4.70: Average confusion matrix for transformer ternary classifier based on median passing grade with Intervals data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	4	10	3
Actual Warning	3	14	6
Actual Pass	3	15	5

Table 4.71: Average confusion matrix for transformer ternary classifier based on median passing grade with Timestamp data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	6	7	4
Actual Warning	4	11	8
Actual Pass	2	9	11

Table 4.72: Average confusion matrix for transformer ternary classifier based on median passing grade with Multivariable data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	10	4	3
Actual Warning	7	7	9
Actual Pass	6	7	10

The training and testing accuracy of the best transformer ternary classifiers with the classification groups described above are shown in Table 4.73 for each type of input data. As previously, a determination of the best model was made based on the highest testing accuracy. The accuracy of the best model with the interval data was 46.03%, and the best

timestamp and multivariable data producing models had accuracies of 61.90% and 58.73%, respectively. The number of epochs used by each model was less than one hundred, as shown in the table.

Table 4.73: Best values for transformer ternary classifier based on median passing grade (entire term)

Type of Data	Train (%)	Test (%)	Epochs
Intervals	95.48	46.03	77
Timestamp	61.92	61.90	76
Multivariable	71.50	58.73	97

The error matrices for each of the best transformer ternary classifiers using each input type of data are shown in Tables 4.74 to 4.76. Using the interval data, the best transformer ternary classifier correctly predicted 50% of the failing students, 48% of students in the "warning" group, and 42% of students in the "passing" group. If students who were predicted to be in either of the "failing" or "warning" categories received additional support, then 94% of students who failed would have received that support.

The best model using the timestamp data correctly predicted 46% of the students who actually failed, 86% of students in the "warning" group, and 43% of students in the "passing" group. Using the multivariable data, the best model correctly predicted 37.5% of the students who actually failed but 88% of students in the warning group were classified correctly. Yet, if students who were predicted to be in either of the "failing" or "warning" categories received additional support, then 96% of students who were actually in the "failing" group would have received that support.

Table 4.74: Best confusion matrix for transformer ternary classifier based on median passing grade with Intervals data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	8	7	1
Actual Warning	1	10	10
Actual Pass	7	8	11

Table 4.75: Best confusion matrix for transformer ternary classifier based on median passing grade with Timestamp data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	6	5	2
Actual Warning	2	19	1
Actual Pass	1	13	14

Table 4.76: Best confusion matrix for transformer ternary classifier based on median passing grade with Multivariable data (entire term)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	9	14	1
Actual Warning	0	22	3
Actual Pass	0	8	6

4.5 Group 4 RQ: Time Series Classification (Early Prediction)

The experiments of Sections 4.4.1 and 4.4.4 of pass/fail prediction using CNN and transformer binary classifiers were repeated using timeline data only up until the VW deadline for earlier prediction. These experiments address research question 4.3. The results of these experiments are provided in Sections 4.5.1 to 4.5.2. Tables 4.77 to 4.84 show the results from the CNN models, while Tables 4.85 to 4.92 show the results from

the transformer models. The results of early prediction at the VW deadline using ternary classifiers are available in Appendix D.

4.5.1 CNN Binary Classifiers (Early Prediction)

Tables 4.77 to 4.80 present the average results of early prediction with a CNN binary classifier. The CNN models trained with the multivariable input data had the highest testing accuracy on average at 82.32%. These models also had the fewest false positives on average, however the number of false positives was relatively higher than the number of false negatives regardless of the type of input data.

Table 4.77: Average values for CNN binary classifier (up to VW date)

Type of Data	**Train (%)**	**Test (%)**	**Epochs**
Intervals	73.81	72.81	145
Timestamp	79.95	78.70	244
Multivariable	87.13	82.32	245

Table 4.78: Average error matrix for CNN binary classifier with Intervals data (up to VW date)

	Predicted Fail	Predicted Pass
Actual Fail	2	16
Actual Pass	1	44

Table 4.79: Average error matrix for CNN binary classifier with Timestamp data (up to VW date)

	Predicted Fail	Predicted Pass
Actual Fail	6	12
Actual Pass	2	43

Table 4.80: Average error matrix for CNN binary classifier with Multivariable data (up to VW date)

	Predicted Fail	Predicted Pass
Actual Fail	10	8
Actual Pass	4	41

Tables 4.81 to 4.84 present the results of the best early prediction CNN binary classifier model out of 100 trials. At the VW deadline, the best CNN models were able to achieve high prediction accuracies. The testing accuracies were 84.1%, 90.48%, and 90.48% for the best models trained using interval, timestamp, and multivariable data, respectively. However, each of these models still had a relatively higher number of false positives compared to false negatives, as shown in the error matrices of Tables 4.82 to 4.84.

Table 4.81: Best values for CNN binary classifier (up to VW date)

Type of Data	**Train (%)**	**Test (%)**	**Epochs**
Intervals	77.67	84.13	136
Timestamp	77.84	90.48	500
Multivariable	87.49	90.48	151

Table 4.82: Best error matrix for CNN binary classifier with Intervals data (up to VW date)

	Predicted Fail	Predicted Pass
Actual Fail	8	8
Actual Pass	2	45

Table 4.83: Best error matrix for CNN binary classifier with Timestamp data (up to VW date)

	Predicted Fail	Predicted Pass
Actual Fail	9	4
Actual Pass	2	48

Table 4.84: Best error matrix for CNN binary classifier with Multivariable data (up to VW date)

	Predicted Fail	Predicted Pass
Actual Fail	5	6
Actual Pass	0	52

4.5.2 Transformer Binary Classifiers (Early Prediction)

Tables 4.85 to 4.88 present the average results of early prediction with a transformer binary classifier. On average, all of the transformer models had similar testing accuracies, regardless of the format of the input data. The models trained with the timestamp and multivariable data had slightly higher testing accuracies (76.19% and 75.56%, respectively) on average compared to those trained on the interval data (72.22%). For each type of input data, the majority of students who would go on to pass the course were predicted correctly. However, the transformer models had a tendency to predict that most students would pass the course. The average number of students who actually failed and were incorrectly predicted to pass was 94%, 71%, and 59% using the interval, timestamp, and multivariable data, respectively.

Table 4.85: Average values for transformer binary classifier (up to VW date)

Type of Data	**Train (%)**	**Test (%)**	**Epochs**
Intervals	76.60	72.22	180
Timestamp	81.85	76.19	81
Multivariable	78.24	75.56	84

Table 4.86: Average error matrix for transformer binary classifier with Intervals data (up to VW date)

	Predicted Fail	Predicted Pass
Actual Fail	1	16
Actual Pass	2	44

Table 4.87: Average error matrix for transformer binary classifier with Timestamp data (up to VW date)

	Predicted Fail	Predicted Pass
Actual Fail	5	12
Actual Pass	5	41

Table 4.88: Average error matrix for transformer binary classifier with Multivariable data (up to VW date)

	Predicted Fail	Predicted Pass
Actual Fail	7	10
Actual Pass	12	34

Tables 4.89 to 4.92 present the results of the best early prediction transformer binary classifier model out of 20 trials. At the VW deadline, the best transformer models achieved testing accuracies of 80.95%, 85.71%, and 82.54% for the best models trained using interval, timestamp, and multivariable data, respectively. However, each of these models still had a relatively higher number of false positives compared to false negatives, as shown in the error matrices of Tables 4.90 to 4.92. The best model achieved using the interval data did so by predicting that all students would pass the class, resulting in a relatively high testing accuracy but not very actionable predictions. Both of the best models trained using the timestamp and multivariable data were able to correctly predict just over half of the students who would eventually fail the course.

Table 4.89: Best values for transformer binary classifier (up to VW date)

Type of Data	Train (%)	Test (%)	Epochs
Intervals	69.56	80.95	218
Timestamp	80.73	85.71	78
Multivariable	75.68	82.54	76

Table 4.90: Best error matrix for transformer Binary classifier with Intervals data (up to VW date)

	Predicted Fail	Predicted Pass
Actual Fail	0	12
Actual Pass	0	51

Table 4.91: Best error matrix for transformer binary classifier with Timestamp data (up to VW date)

	Predicted Fail	Predicted Pass
Actual Fail	7	6
Actual Pass	6	44

Table 4.92: Best error matrix for transformer binary classifier with Multivariable data (up to VW date)

	Predicted Fail	Predicted Pass
Actual Fail	7	6
Actual Pass	5	45

4.6 Results Summary

The results of the experiments from the four groups of research questions were presented in this chapter. In the first group it was shown that the intervals between student interactions with the LMS do not follow a normal distribution. Shorter intervals were more common, and the distribution also had a long tail extending to the right indicating that

there were extreme outliers of long interval lengths. The correlations between the temporal features based on the Student Model timelines and student outcomes, at various points during the term, were presented in the second group of research questions. Both the hand-crafted temporal features and the features learned by a ML algorithm were used to predict student outcomes in multiple ML regression models, as well as to group students based on the features with unsupervised clustering. In each case, the ML models trained on the hand-crafted features performed as well or slightly better than when the models were trained with the learned features. When student outcomes were defined as passing or failing the course, the ML models with either set of features had more false positive errors relative to false negative errors. The experimental results of time series classification indicate that both CNN and transformer neural networks are promising techniques to classify students as passing or failing using only the LMS timestamp information without hand-crafted features,

Chapter 5

Discussion

The experimental results that were presented in Chapter 4 are discussed further in this chapter. In Sections 5.1 to 5.4 the observed trends are grouped into the four sets of research questions, and the numerical results are examined in the context of the people behind the data: undergraduate students interacting with a synchronous and fully online course. In Section 5.5, the implications of the results are discussed within the context of education and online courses.

5.1 Group 1 RQs: Patterns of Behaviours

Students' patterns of behaviour throughout a term for a class delivered fully online were explored in this set of research questions by creating quantitative features to represent the students' temporal interactions with the LMS.

The average number of interactions per student during the term was double the average number of timeline items when rounded to the nearest whole number, as shown in Table 4.5. This implies that, on the whole, students were not returning to the same LMS pages many

times. This finding is notable because all of the interval-based features were created using the date-timestamped interactions recorded in the Data Hub, which does not record the dates or repeated interactions with the same content. Since the number of number of timeline items relative to the number of interactions per student remained nearly constant, the timelines can be considered to be a reasonable representation of student interactions with the course LMS.

The amount of time between when students, on average, submitted assignments compared to the due date decreased over the term from nearly two days (43 hours) to just over one day (27 hours). This means that students submitted later assignments closer to the due date compared to earlier assignments in the term. There could be a variety of reasons for this trend, such as: assignments that cover more course material require more time, students become busier as the term progresses and need to balance their time, or students begin to start assignments later either out of necessity or because they have a better idea of how long they will need to complete the work.

At each point that it was calculated during the term, the burstiness measure was nearly 1, implying that students are predominately accessing the material in a bursty way, rather than in consistently spread out throughout the term. One explanation for the high degree of burstiness could be if students are downloading multiple files when they interact with the course LMS.

The fraction of non-zero days is the fraction of days that student's interacted with the course LMS at least once. On average, this value decreased over the term from approximately 0.47 at the one-quarter mark to 0.43 by the final exam, corresponding to a minimum of three days with interactions every week (three out of seven days is a fraction of 0.43 non-zero days). By the end of the term, the fraction of non-zero days dropped to 0.36, which may be attributed to decreased activity during the period of time after the final

exam when the LMS course page remained open to students. Since it was shown earlier that the number of interactions with the LMS remained, on average, relatively constant during the term while the fraction of non-zero days decreased, this implies that students' behaviour became more bursty over time.

The distribution of intervals was calculated, as shown in Figure 4.1. From this *probability mass function* (PMF), the length of the intervals does not have a normal distribution. The PMF of the intervals instead had a very long right tail, meaning that although the majority intervals were relatively short, there were some intervals that were much longer. The distribution is best described using multiple units of time: the shortest intervals were seconds or minutes long, and longer intervals could be measured in days or weeks. The longest interval in this dataset was just over 42 days. Figure 4.2 shows the same PMF with intervals less than one hour or greater than one week cut off. In doing so, it can be seen that the distribution follows a decaying sinusoidal pattern, with peaks corresponding to roughly 24-hour periods. This implies that students were most likely to access course material at the same time of day as their previous interaction. Each student effectively set a personal schedule.

5.2 Group 2 RQs: Correlation of Features and Outcomes

The second set of research questions explored the correlations between students patterns of behaviour at different points in time during the course and student outcomes. As observed in the figures, the number of interactions is highly correlated with the number of timeline items at all points during the term. This again shows that although the timelines do not capture every interaction the students have with the course LMS, they are representative of those interactions.

The number of interactions, and number of timeline items, are also highly correlated

with the fraction of non-zero days. This indicates that students who interact with the course LMS more, do so across a larger number of days compared to students who have fewer interactions.

The fraction of non-zero days at the quarter-point in the term with the midterm grades is 0.18. However, the correlation between the fraction of non-zero days with the final grade increases from 0.33 to 0.58 over the term. These trends support the assertion found in the literature and supported by behaviourism that students who are more engaged with the course material, whatever their motivation may be, tend to have higher final grades [BrPH22][MaDa10].

5.3 Group 3 RQs: Feature-Based Prediction and Early Prediction

In the third set of research questions, predictive ML models were used to explore which features of the temporal behaviours had the greatest predictive capabilities for student course outcomes. The results were then compared to predictions using features that were automatically generated from the raw timestamp data. Overall, the temporal features that were hand-crafted from the LMS data yielded similar or fewer prediction errors compared to the predictions with the automatically generated features, indicating that these hand-crafted temporal features may add predictive value.

For early prediction, the feature values at earlier points during the term were used as input to the predictive models. Using linear regression with multiple features in combination, the MAE could be reduced to just under 15.5, meaning that the final grade could be predicted to within nearly 15 percentage points. Although this is not accurate enough to predict final letter grades, it could be helpful in predicting students who will be closer to a

failing grade.

Logistic regression was used to predict whether students passed or failed the course, based on a minimum final grade of 50% needed to pass. Overall, the logistic regression algorithm was successful at predicting students with passing final grades. False positives, students predicted to pass but who actually failed, were common. In each trial using the hand-crafted features from the Student Model, approximately 20% of the outcomes were false positives. Larger training sets decreased the number of false positives, and it is possible that with larger student datasets to train the model, the number of false-positives could decrease further. The number of false negatives, students predicted to fail but who actually passed, were rare with at most one student out of each trial in this category. Logistic regression trials using the automatically generated features, that were learned from the raw data using ML, were inconclusive as they failed to converge.

The nearest neighbours algorithm had fewer prediction errors than logistic regression. All values of k greater than two performed as well as, or better, than logistic regression at predicting pass or fail outcomes. For this dataset, using the temporal features from the Student Model, the optimal value of k was 3, which resulted in 10 false positives and 4 false negatives. The number of overall prediction errors, as well as false positive errors, using the ML generated features was higher for all values of k. From an educational standpoint, in order to identify students at risk of failing the course, minimizing the number of false positives is more helpful than minimizing the number of false negatives.

5.4 Group 4 RQs: Time Series Classification

In the fourth set of research questions, NN classifiers were used for student course outcome prediction using LMS data. After performing the student course outcome prediction experiments with CNN and transformer classifiers, a number of trends were observed. Both

types of classifiers showed promising results, with testing accuracies of up to 98.41%, indicating that these types of models can provide insights into LMS data. The binary classifiers in these experiments produced testing accuracies that met or exceeded those of other models in the literature [ArBR22].

As shown in the tables presented in Subsections 4.4.1 to 4.4.3, the accuracy of the CNN classifiers was influenced by the format of the input data. Across both the binary and ternary CNN classifiers, the prediction accuracy was lowest with the interval input data and highest with the multivariable data. Across the experiments, the multivariable data produced results in range of 10-20% more accurate than when the interval data was used as input. The best CNN binary classifier model was 98.41% accurate using the multivariable data. The best CNN ternary classifier model achieved a testing accuracy of 82.54%, also using the multivariable data, when the classification groups were based on the final letter grades.

Similarly, the accuracy of the transformer classifiers was also influenced by the format of the input data. The transformer classifiers that were trained on the interval input data were also the least accurate, producing results that were approximately 10% less accurate than the timestamp data on average. However, for both the binary and ternary transformer classifiers, the accuracy between the models trained on the timestamp and multivariable data was negligible. The best binary transformer classifier model was 90.48% accurate using the multivariable data, and the best ternary transformer classifier model was 61.90% accurate using the timestamp data.

Another observed trend was the relative performance of ternary classifiers using classification groups based on letter grades compared to the groups based on the median passing grade. Both the CNN and transformer ternary classifiers achieved slightly higher accuracies when the groups were based on the final letter grades, as shown in Table 5.1. This

suggests that these types of models may be useful in predicting students who may receive a near-failing grade, but how that is defined impacts the accuracy and therefore usefulness of the model.

Table 5.1: Comparison between ternary classification groups accuracies using multivariable data

Model Type	Groups based on	Average (%)	Best (%)
CNN	letter grades	64.57	82.54
CNN	median of passing grade	61.95	76.19
Transformer	letter grades	49.84	58.73
Transformer	median of passing grade	48.49	58.73

Trends were also observed between the performance of the CNN and transformer classifiers, when they were run under the same conditions. Tables 5.2 to 5.4 compare the average and best predication accuracies of the CNN and transformer binary and ternary classifiers using the multivariable data as input. As shown in these tables, in each case the CNN models outperform the transformer models. The number of training epochs was also higher for the CNN classifiers, with the specific numbers provided in the results section.

Table 5.2: Comparison between average and best binary CNN and transformer models using multivariable data

	Average (%)	Best (%)
CNN	86.35	98.41
Transformer	79.37	90.48

Table 5.3: Comparison between average and best ternary CNN and transformer models using multivariable data and groups based on letter grades

	Average (%)	Best (%)
CNN	64.57	82.54
Transformer	49.84	58.73

Table 5.4: Comparison between average and best ternary CNN and transformer models using multivariable data and groups based on the median passing grade

	Average (%)	Best (%)
CNN	61.95	76.19
Transformer	48.49	58.73

In addition to the testing accuracy, another way to quantify the performance of the CNN and transformer models is with the number of false positives and false negatives. Table 5.5 shows the average number of false positives and false negatives for each type of binary classifier and type of input, as well as the percentage of passing and failing students that were classified incorrectly in each case.

Table 5.5: Average false positives and false negatives binary CNN models and binary transformer model

	False Positives	Failing students classified incorrectly	False Negatives	Passing students classified incorrectly
CNN intervals	15	83.0%	2	4.4%
Transformer intervals	14	82.0%	5	11.1%
CNN timestamps	9	52.9%	4	8.7%
Transformer timestamps	11	64.7%	4	8.7%
CNN multivariable	6	33.3%	3	6.7%
Transformer multivariable	10	55.6%	8	17.8%

Table 5.6 also shows the number of false positives, false negatives, and percentage of passing and failing students that were classified incorrectly for the best model trained on each type of input data.

Table 5.6: False positives and false negatives in the best binary CNN models and binary transformer models

	False Positives	Failing students classified incorrectly	False Negatives	Passing students classified incorrectly
CNN intervals	4	26.7%	3	6.3%
Transformer intervals	8	61.5%	3	6.0%
CNN timestamps	6	37.5%	0	0%
Transformer timestamps	7	53.8%	0	0%
CNN multivariable	1	8.3%	0	0%
Transformer multivariable	6	37.5%	0	0%

As shown in Tables 5.5 and 5.6, the number of false negatives (i.e. students who passed but were predicted to fail) is low across nearly all models, and also lower than the number of false positives (i.e. students who failed but were predicted to pass) in most cases. The percentage of failing students who were, on average, classified incorrectly ranged from 33.3% to 83% depending on the type of model and type of input data. In contrast, the percentage of passing students who were classified incorrectly on average was in the lower range of 4.4% to 17.8% over the same models. This implies that some students who failed had LMS interaction patterns which were similar to students who passed the course.

For student course outcome prediction, the number of false positives is more important than the number of false negatives. Students who are predicted to pass, but who actually

are more at risk of failing, are done a disservice by being misclassified compared to students in the opposite situation. On average, the number of false positives was influenced by the type of input data as shown in Table 5.5. For both the CNN and transformer models, the models trained on the interval data resulted in the highest number of false positives, while the models trained on the multivariable data resulted in the fewest number of false positives.

It was also observed that in these experiments the CNN models produced the same or fewer mis-classifications (false positives or false negatives) compared to the transformer models, when comparing average results of models trained with the same type of input data. As shown in Table 5.5, on average the percentage of failing students who were classified incorrectly was the same or lower for the CNN models with each type of input data (between 33.3% and 83.0%) compared to the transformer models trained on the same type of input data (55.6% to 82.0%).

An additional way to directly compare the performance of the CNN and transformer classifiers is shown in Table 5.7. This table shows the results of training both a CNN and transformer binary classifier with multivariable data as input using the same random starting seed and same test-train split of the input data. After training these specific models, the CNN binary classifier had a testing accuracy of 88.89% and the transformer binary classifier had a testing accuracy of 76.19%. Table 5.7 presents the number of students who were classified correctly by both of these models, only one of these models, or by neither model, delineated by course outcome. As shown in the table, both models correctly classified the majority of students who passed, as well as just over half of the students who failed. The CNN model correctly classified an additional 7 passing students and 5 failing students over the transformer model, while the transformer correctly classified an additional 4 passing students over the CNN model. All of the passing students were correctly classified by at least one of the CNN or transformer models, however both models failed to predict 3 of the

students who failed.

Table 5.7: Comparison of a CNN binary classifier and transformer binary classifier trained with the same initial conditions and multivariable input data.

	True Pass	True Fail
Total number students in group	46	17
Number of students predicted correctly by both CNN & Transformer	35	9
Number of students predicted correctly by CNN only	7	5
Number of students predicted correctly by Transformer only	4	0
Number of students predicted correctly by neither	0	3

The experimental results of prediction at the VW deadline using both CNN and transformer binary classifiers indicate that this approach may be useful for course outcome prediction earlier in the term as well. The best CNN binary classifiers achieved testing accuracies of 90.48% and the best transformer binary classifier had a testing accuracy of 85.71%. These results were lower than the predictions achieved using the full timelines (98.41% and 90.48%, respectively), which is to be expected with less input data.

The results suggest that the CNN models are better suited to student course outcome prediction using behavioural data based on LMS timestamps than transformers, at this time, when using the metrics of testing accuracy and false positives. However, unlike feature-based prediction methods, neither the CNN or transformer classifiers offer insight into why students were classified into each group. Nevertheless, in many settings it may not be necessary to know the reasoning behind the prediction. For example, in deciding whether to offer additional support to a student, their predicted course outcome may be sufficient to act upon. To that end, both types of neural network classifiers produced promising results at course outcome prediction, and there is room for future study on optimizing CNN, transformer, and other types of neural networks for time series classification of LMS

data.

5.5 Broader Implications in Education

A number of trends and points for future consideration emerged through the experiments presented in Chapters 3-4 that could be informative to educators and students. However, it is important to note that the correlations discussed in this work are observed trends. In particular, correlations between features and final grades do not mean that the final grade was caused by those feature values.

From the study of timeline interval characteristics, including the PMF, it was found that students naturally develop a schedule of interacting with the LMS at daily or multi-day intervals around a personally consistent time of day. Working within this structure, educators can potentially help to minimize the cognitive burden on students by uploading new course content at the same time each day. Doing so will mimic students' behaviour of consistent daily or multi-day interaction within the LMS, possibly creating a sense of stability that new content will not be missed. As well, since the number of timeline items was about half of the student's total number of interactions with the LMS at all points during the term, it implies that students did not revisit the same content pages many times over. This could be because each LMS item was visited only a few times, or because students downloaded the content and did not need to revisit it on the LMS. The high degree of burstiness in the intervals also could be explained by multiple content pages being downloaded in quick succession every few days. Once students have downloaded files (such as course notes, practice problems, or assignment instructions) from the LMS to their own devices, they have little reason to check that material again on the LMS for any updates.

In previous research, the impact of sending messages to students as reminders to visit

content was inconclusive [HaBH17]. By understanding how students naturally engage with the LMS, it may be possible to optimize the timing of such messages. Furthermore, if instructors interact with the LMS in a predictable pattern of behaviour, implicitly or explicitly, students may respond by incorporating this knowledge into their own patterns of behaviour in order to avoid missing content. If an instructor deviates from their established pattern, they can potentially help students by prominently communicating this change in classroom and LMS announcements, to ensure the changes aren't missed. This could streamline both instructor's and students' interactions with LMS, making the interactions more efficient and thereby potentially more effective (e.g., students won't miss uploaded content), which could reduce students' cognitive load and improve students' learning [Kirs02].

Using the feature-based ML prediction algorithms, it was shown that students' behaviour interacting with the LMS was more successful at predicting a binary outcome of pass or fail, compared to membership in a final grade group (strong, competent, developing, passing, or failing). The neural network-based classifiers were also more successful at predicting a binary groups compared to ternary groups using the LMS data. In part, this can be attributed to the relatively larger sizes of the binary groups compared to when the students are divided into three or more grade groups. Performing the analysis on larger data sets with additional real or synthetic student data could yield better results. However, from a practical standpoint, it is more important to be able to predict whether a student is likely to pass or fail, rather than which letter grade they will receive [DeDS22] [MBKK03]. For example, additional supports are more critical for a student who is not on track to pass at all, compared to one who is on track to complete the course between 'strong' and 'developing'.

It is possible that course outcome predictions based on interactions with the LMS may be improved through the implementation of additional behavioural features, such as the fractal dimension and learning entropy for polyscale and multiscale analysis discussed

earlier. At this time, the LMS software does not track and record sufficient data to calculate these features. Computer storage continues to become more affordable and accessible, and as the importance and interest in tracking educational data increases the range of what is available through LMS may also grow.

Through the lens of behaviourism, students' observable behaviours can be used as a proxy for their learning. However, encouraging students to behave in certain ways throughout the term based on the feature correlations will not guarantee that the students learn, or even pass the course. Indeed, no prediction algorithm was able to use student behaviours with the LMS to predict course outcomes with complete accuracy. However, students' behaviours in terms of their interactions with an LMS can help to identify when to provide additional support and who may most benefit from it. Furthermore, educators can promote the behavioural features that correlate with higher grades, or are predictive of higher grades, for all students as well as particularly to help students who are new to an online course environment and who are forming new habits as they learn how to learn within the environment. Although the NN classifiers can not provide an indication of how students should behave, given the predictive success of these classifiers with the LMS data it is conceivable that they could be used to develop decision support tools to support educators in determining where to direct their support.

Even with additional data there would still be much work to be done to improve the accuracy of student course outcome predictions. There is an art to the science of ML. Feature selection and hyperparameter tuning can be aided by algorithms, but the development of features from the raw data is still often a human endeavour [Feat22]. The data can be overfit to the ML models, making the model very good at predicting the training data but bad at predicting any new data points. Overfitting is of particular concern in datasets where one of the prediction groups is has many more members than the other. This can make it difficult to accurately predict data points in the group with fewer members. For example, if

only 1% of students in a dataset are in the 'failed the course' group, then a ML model will be correct 99 out of 100 times just by predicting everyone as passing. In this dataset, there was a nearly 70-30 split between students who passed and those who failed. However, with the goal of increasing the number of students who pass, the datasets will become even more imbalanced. Data sets from other courses may naturally have a higher pass-to-fail ratio already and would therefore be more susceptible to overfitting. Additional data collection, or the creation of synthetic data, may be required to overcome this imbalance[DoAB19].

It is also important to note that the dataset in this study contained LMS interactions from students in a first year computer science course. One of the characteristics of this course is the large number opportunities for interaction with the LMS since there are many quizzes, assignments, and files (content pages) throughout the term. It is expected that student course outcome prediction using LMS data would be relatively more difficult in courses with limited opportunities for interaction within the LMS. As well, as discussed in Chapter 2, most studies on student outcome prediction focus on a set of students at the same education level [Spit21] [LoBe21] [LiCh20] [TiLW20] [SoOk20] [JiNT22]. The behaviours that correlate with and predict success may differ between these different types of students. Even within an undergraduate degree, introductory courses have different expectations compared to upper-level courses, and also typically cater to students who have less experience with learning management systems. As well, the behaviours that correlate with and predict success in a course may differ depending on the format of the course, such as discussion-based course compared to a programming course [YuPS19]. The method of course delivery may also impact student behaviours, and typically studies on student outcome prediction focus on courses with the same mode of delivery, whether that is face-to-face, online, or a mixture [HaBH17] [RGPO21] [PaML19] [YuWu21].

5.6 Discussion Summary

Students' behavioural patterns as represented by their interactions with an LMS during a synchronous online course, the correlation of the temporal features and student outcomes, the predictive capabilities of the temporal features to predict student outcomes, and the suitability of neural network-based classifiers for LMS timestamp data were discussed. Although there was not one feature or subset of features that was highly predictive of student outcomes, the features illustrated trends in student behaviours within the LMS. The neural network-based classifiers produced more accurate student outcome predictions than the feature-based ML models at the expense of interpretability. That is, it was not possible to identify why these models classified a student into a particular course outcome group. Broader implications of the findings were also discussed in the context of education and student support, including potential considerations on how to work with students' natural patterns of interaction with the LMS in order to reduce their cognitive load.

Chapter 6

Conclusions

This thesis presented a code-based tool to create features describing student interactions over time within multiple iterations of a single-term online course, which were extracted from raw LMS date-time stamp data. This was followed by an investigation of the successes and limitations of these features to predict student grade outcomes using ML models. Then the suitability of CNN and transformer NN classifiers for student course outcome prediction using LMS data was investigated. Chapter 2 began with an introduction to the evolution of AI and ML; trends in educational learning theories; and a discussion of how ML methods can support research in education. As well, the current state of ML being used in education-based research was presented. The method of developing individual student timelines out of LMS interaction intervals and assessing time-based features from the raw LMS data was presented in Chapter 3, and ML prediction models were introduced. The statistical analysis of the features and experimental results of the ML prediction models were presented in Chapter 4. In Chapter 5, the results were discussed in relationship to the research questions, and within the broader context education field.

6.1 Thesis Conclusions

This thesis addressed several research questions related to using time-based features created from raw LMS data to understand student behaviour and predict final grade outcomes. This section links back to the research questions first outlined in Section 1.2.3, to provide insight into them from the experimental results.

A statistical analysis of the student timeline intervals showed a highly skewed distribution. Although the majority of intervals measured less than two days, some intervals were as long as 42 days. Within the PMF, there was a decaying sinusoidal pattern with periods of approximately twenty-four hours.

The number of student interactions with the LMS was found to be highly correlated with the number of timeline items at all points during the term, showing that the timelines are a reasonable representation of student interactions with the LMS over time. The number of interactions, number of timeline items, fraction of days with at least one interaction (e.g. non-zero days), and fraction of allowed writing time spent on the midterm (first term test) each positively correlated with the final grade, whereas the average interval length and variance of the interval length negatively correlated with the final grade. The correlation between the features and the final grade remains consistent or increases over the term.

Using ML models, the time-based features predicted students' final grades or whether a student would pass or fail with greater than random accuracy. The features that most contributed to the predictions were the fraction of days with at least one interaction, and the fraction of allowed writing time spent on the midterm (first term test). In the binary prediction algorithms, the number of false positives was higher than the number of false negatives. Of the prediction models used, the k-Nearest Neighbours algorithm was the most accurate. Features created from the student timelines at the one quarter point in the term were little better than random chance at predicting final grade outcomes. However, by the

halfway point in the term the predictions improved, and by the three-quarter point in the term the predictions were nearly the same as using features from the full timelines.

Student course outcome prediction was also performed with CNN and transformer neural network classifiers, using LMS data without hand-crafted features. After training, the binary classifiers for pass/fail prediction achieved testing accuracies above 90%, reaching as high as 98%. Both CNN and transformer classifiers produced promising results at course outcome prediction, and there is room for future study on optimizing these models for time series classification of LMS data.

6.2 Contributions

This thesis contributes to the body of knowledge on student modelling and prediction, as well as student behaviour within an LMS in an online course. The following are the main contributions:

1. A tool was designed and built in code to process date-timestamp information from an LMS in a novel way. The tool is generalizable to large, online undergraduate courses that make use of LMS assignments, quizzes, and content pages, as it uses standardized data collected from an LMS. Further, the models built in this work use anonymous, quantitative measurements that are already collected by post-secondary institutions. Therefore, other researchers can build on this work. For example, the model could be further trained with additional features to improve pass/fail and grade predictability.

2. The statistical properties of the intervals between student interactions with the LMS in a synchronous and fully online environment were studied. Previous work on intervals was in a face-to-face class, which did not capture as many student interactions with the course through the LMS [DeBr20]. It was found that students predominately

interact with the system in bursts, and in patterns of interactions at the same time on any given day relative to their previous interaction. These patterns emerged by studying the intervals between interactions, rather than the dominant access times. The findings can also help inform instructors with how to manage their own LMS interactions. By limiting LMS updates to the same time each day, and using LMS announcements if a deviation from this pattern is necessary, instructors may students to be confident that they are not missing any content; this would lead to a reduction of the cognitive load on students as per *Cognitive Load Theory* (CLT).

3. The hand-crafted time-based features for early prediction of student course outcomes, as measured by final grade, were used. It was found that these features could predict student final grades, as well as whether students would pass or fail the course, with greater than random accuracy before the end of the term. Furthermore, there were the same number or fewer prediction errors when using these features based on the Student Model timeline compared to when the automatically generated features were used. Therefore, the results can help identify the students who are potentially on a path of behaviour to fail and can notify instructors of the need for an intervention, which is particularly useful in large classes of students.

4. Student course outcome prediction from LMS timestamp data without hand-crafted features was explored using CNN and transformer time series classifiers. The input time series data was prepared in three ways: as intervals, as unlabelled timestamps, and as timestamps labelled with the type of LMS interaction. After training, both CNN and transformer time series classifiers were shown to be able to achieve testing accuracies above 90% with the labelled timestamp data. This indicates that high course outcome prediction accuracies can be achieved with ML techniques using only behavioural data sets that are readily available from an LMS.

5. The work supports the collection of temporal data in LMS data sets, and can inform developers of features that could be added to support student learning and successful course outcomes. For example, personalized notifications could let students see their own patterns of behaviour and how their behaviour correlates with students who have previously passed the course. Early warning systems with messages directed to the learner, instructor, or both are a current area of study [MaDa10]. These messages could be displayed within the LMS, sent as automated e-mails, or integrated into other digest e-mails such as MS Teams which already tracks general account usage. This research into students' natural behavioural patterns can support work currently being done to determine when messages to students would be most effective [HaBH17].

6. This study illustrates preliminary ways in which an LMS can be used to identify students' behaviour, encourage interventions, and potentially increase the number of students who can positively increase their learning outcomes by being made aware of, and changing their behaviours.

6.3 Limitations and Future Work

There is still much work that can be done to improve the area of student course outcome prediction. Humans are complicated and individual, and so is human behaviour and learning. Completely accurate early prediction of course outcomes may not be achievable, but there is room for future improvements. There are still limitations and unanswered questions within the field that provide rich ground for further research.

In this study, the student interaction data within an LMS consisted of just over 300 students. Increasing the sample size to more students could increase the accuracy of the ML prediction models. One way to increase the sample size is to increase the number of students included in the study [MbMG22] [TeBP19]. This is the most common approach,

however it requires time to collect the data. For example, if data is being collected about a particular type of course, such as COMP 1010 from UM Learn, increasing the sample size requires waiting for the course to be held over multiple terms. Additionally, this approach requires coordination between the researchers and the team managing the LMS or MOOC data.

Another way to increase the sample size of the data set is to create synthetic data. Synthetic data is generated artificially, such as by a computer, and can be created to have particular statistical characteristics. It can be useful to explore how a ML model will handle cases that rarely occur naturally or were previously unknown [DiRM19]. Synthetic data is also advantageous when there are privacy concerns with the real-world datasets, as is the case with many types of education data, which may include demographic information or past academic performance. The generation of synthetic time series data is still an active area of research. Approaches such as dynamic stationary processes, *Markov models* (MM models), and *auto-regressive models* (AR models) have been used to generate large quantities of synthetic time series data in multiple fields. However the ability of these techniques to accurately represent the characteristics of the original data set have so far been limited and may rely on *a priori* knowledge of correlations in the data, making them unsuitable for pattern discovery [FoPD17] [LiJW20]. More recently, RNN and *generative adversarial network* (GAN) methods are being researched for synthetic time series generation with promising results that approximate the distributions of the original data sets in areas such as medical and transmission systems time series [ZhKK18]. However, these models are less effective at generating synthetic data when the original time series can be variable in length or is highly skewed, such as in networking data or LMS interaction data [LiJW20]. Therefore, there is room for additional research into this area of synthetic time series generation.

Future work in this area could benefit education research as well, by supporting the creation of an open repository that would provide researchers with a common set of syn-

thetic datasets allowing for greater transparency in student outcome prediction research and support the creation of standard metrics for comparisons between studies [DoAB19].

Even when using data directly collected from an LMS, a limitation to student modelling and prediction is the amount and type of data currently available about a given set of students, as collected by education software. Although the education software is always changing and improving, at present it was not possible to implement features such as the fractal dimension and learning entropy using the amount of data currently collected by the LMS. The UM Learn LMS only records the date-time stamp of the most recent interaction with each content page and total number of interactions, rather than a date-time stamp for every interaction. There are also interactions that are not recorded, such as attendance at the lectures, time-on-task measurements, or participation on the discussion forums. Some of these features are implemented in other LMS software packages and have been shown to have value in student outcome prediction [OrVa20] [TeRN19] [KuGI11] [TiLW20] [Bail20] [ThPA13] [Kim14]. Having access to the temporal information of all these interactions would allow for more complete and accurate student timelines in the Student Model presented here, without introducing features composed of private student information such as demographic or transcript data.

In this study, the dates and times stored by the LMS were assumed to be accurate. However, further study should be done to ensure the accuracy and reliability of the timing information as doing so would increase the effectiveness of the features and models for predicting student outcomes.

As well, the data in this study came from students who were enrolled over three terms in the same first-year computer science course delivered fully online. The high testing accuracies in this research indicate the potential of ML techniques for student course outcome predictions. Additional experiments could be performed to assess how the time-based fea-

tures and time series classification perform when the input data is collected from different types of online post-secondary courses such as courses in different subject areas, courses held at different years within an undergraduate program, courses offered during different academic terms, or courses of different durations. Future work that builds on this research using larger and more diverse input data sets could help to produce more generalized student course outcome prediction models for use in decision support system research.

The results of student outcome predictions using machine learning could also be explored within the context of different educational learning theories. For example, Fink's categories of significant learning [Fink03] includes "learning how to learn", which could be explored further using students' behavioural patterns within an LMS as well as support research into the personalisation of learning.

References

[Adam02] D. Adams, *The Salmon of Doubt*, MacMillan, London, 2002.

[Aira01] P. W. Airasian, *A taxonomy for learning, teaching, and assessing : a revision of Bloom's taxonomy of educational objectives*, Longman, New York, 2001, ISBN 080131903X.

[Anto09] F. Antonio, "The technological dimension of a massive open online course: The case of the CCK08 course tools," *International Review of Research in Open and Distributed Learning*, vol. 10(5), 2009.

[ArBR22] C. J. Arizmendi, M. L. Bernacki, M. Rakovic, R. D. Plumley, C. J. Urban, A. T. Panter, J. A. Greene, and K. M. Gates, "Predicting student outcomes using digital logs of learning behaviors: Review, current standards, and suggestions for future work," *Behavior research methods*, 2022.

[ArPi12] K. E. Arnold and M. D. Pistilli, "Course Signals at Purdue: Using Learning Analytics to Increase Student Success," *in Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, pp. 267–270, 2012.

[Ausu68] D. Ausubel, *Educational psychology: A cognitive view*, Holt, Rinehart & Winston, New York, NY, 1968.

[Bail20] J. L. Bailie, "Online Learner Analytics of Asynchronous Discussions as a Predictor of Final Grades," *Journal of Instructional Pedagogies*, vol. 24, 2020.

[BaLL15] R. Baker, D. Lindrum, M. Lindrum, and D. Perkowski, "Analyzing early at-risk factors in higher education e-learning courses," *International Educational Data Mining Society*, 2015.

[Bake14] R. S. Baker, "Educational Data Mining: An Advance for Intelligent Systems in Education," *IEEE intelligent systems*, vol. 29(3), pp. 78–82, 2014.

[BaZE17] B. Bakhshinategh, O. R. Zaiane, S. ElAtia, and D. Ipperciel, "Educational data mining applications and tasks: A survey of the last 10 years," *Education and information technologies*, vol. 23(1), pp. 537–553, 2017.

[Band77] A. Bandura, *Social Learning Theory*, Prentice Hall, University of Michigan, 1977, ISBN 9780138167516, 247 pp.

[Bara20] C. Barabas, "Beyond Bias: Ethical AI in Criminal Law," *in The Oxford Handbook of Ethics of AI*, Oxford University Press, 2020.

[Bara05] A.-L. Barabasi, "The origin of bursts and heavy tails in human dynamics," *Nature*, vol. 207(435), p. 207211, 2005.

[Bate19] A. Bates, *Teaching in a Digital Age - Second Edition*, Tony Bates Associates Ltd., Vancouver, B.C., 2019, URL `https://pressbooks.bccampus.ca/teachinginadigitalagev2/`.

[BeCU20] M. Bernacki, M. Chavez, and P. Uesbeck, "Predicting achievement and providing support before STEM majors begin to fail," *Computers and Education*, vol. 158, 2020, URL `https://doi.org/10.1016/j.compedu.2020.103999`.

[BiCS22] K. A. Bird, B. L. Castleman, Y. Song, and R. Yu, "Is Big Data Better? LMS Data and Predictive Analytic Performance in Postsecondary Education," *EdWorkingPaper*, vol. 22-647, 2022.

[Bool47] G. Boole, *The mathematical analysis of logic: being an essay towards a calculus of deductive reasoning*, Macmillan, Barclay, & Macmillan, Cambridge, 1847.

[BoSK16] M. S. Boroujeni, K. Sharma, Ł. Kidziński, L. Lucignano, and P. Dillenbourg, "How to Quantify Student's Regularity?" *in Adaptive and Adaptable Learning*, pp. 277–291, 2016.

[BoPM17] D. Borsa, B. Piot, R. Munos, and O. Pietquin, "Observational Learning by Reinforcement Learning," , 2017.

[BrPH22] S. Brdnik, V. Podgorelec, and T. Hericko, "Utilizing Interaction Metrics in a Virtual Learning Environment for Early Prediction of Students' Academic Performance," *in Proceedings of the 9th Workshop on Software Quality, Analysis, Monitoring, Improvement, and Applications*, Novi Sad, Serbia, 2022.

[Brit16] T. E. o. E. Britannica, "cogito, ergo sum," , 2016, URL `https://www.britannica.com/topic/cogito-ergo-sum`, accessed Aug. 1, 2022.

[Broc20] R. Brock, "ConnectionismEdward Thorndike," *in Science Education in Theory and Practice*, Springer Texts in Education, pp. 101–112, Springer International Publishing, 2020.

[Brun60] J. S. Bruner, *The process of education*, Harvard University Press, Cambridge, MA, 1960, 132 pp.

[Chan14] R. Chandrasekar, "Elementary? Question answering, IBMs Watson, and the Jeopardy! challenge," *Resonance*, vol. 19(3), pp. 222–241, 2014.

[Clar68] A. C. Clarke, *2001: A Space Odyssey*, New American Library, Inc., New York, NY, 1968.

[CrAA18] A. I. Cristea, M. Alshehri, A. Alamri, M. Kayama, C. Stewart, and L. Shi, "How is learning fluctuating? FutureLearn MOOCs fine-grained temporal analysis and feedback to teachers." *in* B. Andersson, B. Johansson, S. Carlsson, C. Barry, M. Lang, H. Linger, and C. Schneider, editors, *Proceedings of the 27th International Conference on Information Systems Development (ISD2018)*, p. 6, Association for Information Systems, Lund, Sweden, 2018, URL `http://dro.dur.ac.uk/25775/`.

[D2L22] D2L Corporation, "Brightspace Data sets," URL `https://documentation.brightspace.com/EN/insights/data_hub/admin/bds_title.htm?`

[Data21] DataShop@CMU, "Datashop Home," , July 2021, URL `https://pslcdatashop.web.cmu.edu`.

[DeDS22] G. Deeva, J. De Smedt, C. Saint-Pierre, R. Weber, and J. De Weerdt, "Predicting student performance using sequence classification with time-based windows," *Expert Systems with Applications*, vol. 209, 2022, URL `http://dx.doi.org/10.1016/j.eswa.2022.118182`.

[DeBr20] O. Dermy and A. Brun, "Can we Take Advantage of Time-Interval Pattern Mining to Model Students Activity?" *in Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, 2020.

[DiRM19] G. Dimic, D. Rancic, N. Macek, P. Spalevic, and V. Drasute, "Improving the prediction accuracy in blended learning environment using synthetic minority oversampling technique," *Information Discovery and Delivery*, vol. 47(2), pp. 76–83, 2019.

[Domi15] P. Domingos, *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*, Basic Books, New York, NY, 2015, ISBN 9780465065707, 329 pp.

[DoAB19] M. Dorodchi, E. Al-Hossami, A. Benedict, and E. Demeter, "Using Synthetic Data Generators to Promote Open Science in Higher Education Learning Analytics," *in 2019 IEEE International Conference on Big Data (Big Data)*, pp. 4672–4675, 2019.

[Duna11] M. K. Dunaway, "Connectivism," *Reference services review*, vol. 39(4), pp. 675–685, 2011.

[Fawa20] H. I. Fawaz, "Timeseries classification from scratch," , 2020, URL `https://keras.io/examples/timeseries/timeseries_classification_from_scratch/`.

[Feat22] "Featuretools: An open source python framework for automated feature engineering," URL `https://www.featuretools.com/`.

[FiBR15] C. J. Finelli, M. Borrego, and G. Rasoulifar, "Development of taxonomy of keywords for engineering education research," *J. Engineering Education*, vol. 104(4), pp. 365–387, 20195]]5.

[Fink03] L. D. Fink, *Creating significant learning experiences: an integrated approach to designing college courses*, Jossey-Bass, San Francisco, Calif, 1st edn., 2003, ISBN 0787960551.

[FoPD17] G. Forestier, F. Petitjean, H. A. Dau, G. I. Webb, and E. Keogh, "Generating Synthetic Time Series to Augment Sparse Datasets," *in 2017 IEEE International Conference on Data Mining (ICDM)*, pp. 865–870, New Orleans, LA, USA, 2017.

[Gagn85] R. M. Gagne, *The conditions of learning and theory of instruction*, Holt, Rinehart & Winston, New York, NY, 4th edn., 1985.

[Gard83] H. Gardner, *Frames of mind: the theory of multiple intelligences*, Basic Books, New York, 1983, ISBN 0465025080.

[Garv19] C. Garvey, "Artificial intelligence and Japan's fifth generation: The information society, neoliberalism, and alternative modernities," *Pacific historical review*, vol. 88(4), pp. 619–658, 2019.

[Gebr20] T. Gebru, "Race and Gender," *in The Oxford Handbook of Ethics of AI*, Oxford University Press, 2020.

[Gibn16] E. Gibney, "What Google's winning Go algorithm will do next: AlphaGo's techniques could have broad uses, but moving beyond games is a challenge," *Nature (London)*, vol. 531(7594), pp. 284–, 2016.

[GoBa08] K.-I. Goh and A.-L. Barabasi, "Burstiness and memory in compex systems," *A Letters Journal Exploring the Frontiers of Physics*, vol. 81(48002), pp. 1–5, 2008.

[GoBC16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016, http://www.deeplearningbook.org.

[Gres18] M. Greshko, "Meet Sophia, the Robot That Looks Almost Human," *National Geographic*, 2018, URL `https://www.nationalgeographic.com/photography/proof/2018/05/sophia-robot-artificial-intelligence-science/`, retrieved August 2, 2022.

[Hass11] O. Hassan, "Learning theories and assessment methodologies an engineering educational perspective," *European Journal of Engineering Education*, vol. 36(4), pp. 327–339, 2011.

[HaBH17] D. Hayes, M. Bernacki, W. Hong, J. Markle, and N. Voorhees, "Using LMS Data to provide early alerts to struggling students," *in 2017 FYEE Conference*, 2017.

[Hebb49] D. O. Hebb, *The organization of behavior: A neuropsychological theory*, Wiley, New York, NY, 1949, ISBN 0-8058-4300-0.

[HeLL18] S. Helal, J. Li, L. Liu, E. Ebrahimie, S. Dawson, D. J. Murray, and Q. Long, "Predicting academic performance by considering student heterogeneity," *Knowledge-Based Systems*, vol. 161, pp. 134–146, 2018.

[HeIP18] A. Hellas, P. Ihantola, A. Petersen, V. V. Ajanovski, M. Gutica, T. Hynninen, A. Knutas, J. Leinonen, C. Messom, and S. N. Liao, "Predicting Academic Performance: A Systematic Literature Review," *in Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE 2018 Companion, p. 175199, Association for Computing Machinery, 2018.

[HeHT19] A. Hernndez-Blanco, B. Herrera-Flores, D. Toms, and B. Navarro-Colorado, "A Systematic Review of Deep Learning Approaches to Educational Data Mining," *Complexity*, vol. 2019, p. 22, 2019.

[HiTu93] S. R. Hiltz and M. Turoff, *The Network Nation: Human Communication via Computer*, The MIT Press, 1993, ISBN 9780262291156.

[HoSc21] D. Hopkins and T. Schwanen, "Talking about automated vehicles: What do levels of automation do?" *Technology in society*, vol. 64, p. 101488, 2021.

[HuZZ19] M. Hussain, W. Zhu, W. Zhang, S. M. R. Abidi, and S. Ali, "Using machine learning to predict student difficulties from learning session data," *Artificial Intelligence Review*, vol. 52(1), pp. 381–407, 2019, accessed Oct. 19, 2021.

[JaSh19] M. V. Jamieson and J. M. Shaw, "Learning to Learn: Defining an Engineering Learning Culture," *in Proceedings of the 2019 Canadian Engineering Education Association (CEEA19) Conference*, 2019.

[JiNT22] S. Jiang, A. Nocera, C. Tatar, M. M. Yoder, J. Chao, K. Wiedemann, W. Finzer, and C. P. Rose, "An empirical analysis of high school students' practices of modelling with unstructured data," *British Journal of Educational Technology*, vol. 53(5), pp. 1114 – 1133, 2022.

[JiPa20] W. Jiang and Z. A. Pardos, "Evaluating sources of course information and models of representation on a variety of institutional prediction tasks," *in Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, 2021.

[Kasp17] G. Kasparov, *Deep Thinking: when machine intelligence ends and human creativity begins*, PublicAffairs, New York, NY, 2017.

[Kim14] I.-H. Kim, "Development of reasoning skills through participation in collaborative synchronous online discussions," *Interactive Learning Environments*, vol. 22(4), pp. 467–484, 2014.

[Kins15] W. Kinser, "Expanding the body of knowledge concept for professional practitioners," *in Proceedings of the 2015 Canadian Engineering Education Association (CEEA15) Conference*, 2015.

[Kins88] W. Kinsner and J. Pear, "Computer-aided personalized system of instruction for the virtual classroom," *Can. J. Educational Communication*, vol. 17(1), pp. 21–36, 2018.

[Kins90] W. Kinsner and J. J. Pear, "Dynamical educational system for the virtual campus," *in* U. E. Gattiker and L. Larwood, editors, *Technological Innovation Process and the Human Resources*, vol. II, pp. 201–228, Walter de Gruyter, Berlin, 1990, ISBN 0-89925-686-4.

[Kins19] W. Kinsner and R. Saracco, "Towards Evolving Symbiotic Cognitive Education Based on Digital Twins," *in 2019 IEEE 18th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC)*, pp. 13–21, 2019.

[Kirs02] P. Kirschner, "Cognitive load theory: implications of cognitive on the design of learning," *Learning and Instruction*, vol. 12(1), pp. 1–10, 2002.

[Knut14] D. Knuth, *Art of Computer Programming, Volume 2: Seminumerical Algorithms*, Pearson Education, 2014, ISBN 9780321635761, URL `https://books.google.ca/books?id=Zu-HAwAAQBAJ`.

[KoDM15] K. R. Koedinger, S. D'Mello, E. A. McLaughlin, Z. A. Pardos, and C. P. Ros, "Data mining and education," *WIREs Cognitive Science*, vol. 6(4), pp. 333–353, 2015.

[Koeh10] P. Koehn, *Statistical machine translation*, Cambridge University Press, Cambridge, 2010, ISBN 1-107-21063-1.

[Kolb84] D. A. Kolb, *Experiential learning: experience as the source of learning and development*, Prentice-Hall, Englewood Cliffs, N.J, 1984, ISBN 0132952610.

[Kots12] S. Kotsiantis, "Use of machine learning techniques for educational proposes: A decision support system for forecasting students' grades," *Artificial Intelligence Review*, vol. 37(4), pp. 331–344, 2012.

[KoTF13] S. Kotsiantis, N. Tselios, A. Filippidi, and V. Komis, "Using Learning Analytics to Identify Successful Learners in a Blended Learning Course," *Int. J. Technol.*

Enhanc. Learn., vol. 5(2), p. 133150, 2013, URL `https://doi.org/10.1504/IJTEL.2013.059088`.

[KuFH17] B. Kuipers, E. A. Feigenbaum, P. E. Hart, and N. J. Nilsson, "Shakey: From Conception to History," *AI Magazine*, vol. 38(1), pp. 88–103, 2017, URL `https://ojs.aaai.org/index.php/aimagazine/article/view/2716`.

[KuLB15] Y.-Y. Kuo, J. Luo, and J. Brielmaier, "Investigating Students' Use of Lecture Videos in Online Courses: A Case Study for Understanding Learning Behaviors via Data Mining," *in Advances in Web-Based Learning – ICWL 2015*, pp. 231–237, 2015.

[KuGI11] L. Kupczynski, A. M. Gibson, P. Ice, J. Richardson, and L. Challoo, "The Impact of Frequency on Achievement in Online Courses: A Study from a South Texas University," *Journal of Interactive Online Learning*, vol. 10(3), pp. 141–149, 2011.

[Kurz05] R. Kurzweil, *The Singularity is Near: When Humans Transcend Biology*, Viking, New York, NY, 2005.

[KuHZ17] J. Kuzilek, M. Hlosta, and Z. Zdrahal, "Open University Learning Analytics dataset," , 2017.

[LaNL22] R. Lamb, K. Neumann, and K. A. Linder, "Real-time prediction of science student learning outcomes using machine learning classification of hemodynamics during virtual reality and online learning sessions," *Computers and Education: Artificial Intelligence*, vol. 3, 2022.

[Lecu88] Y. Lecun, "A theoretical framework for back-propagation," *in* D. Touretzky, G. Hinton, and T. Sejnowski, editors, *Proceedings of the 1988 Connectionist Models Summer School, CMU, Pittsburg, PA*, pp. 21–28, Morgan Kaufmann, 1988.

[LeBB98] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86(11), pp. 2278–2324, 1998.

[LePS86] M. Lenat, Doug; Prakash and M. Shepherd, "CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquistion Bottlenecks," *AI Magazine*, vol. 6(4), pp. 65–85, 1986.

[Leve14] H. J. Levesque, "On our best behaviour," *Artificial intelligence*, vol. 212(1), pp. 27–35, 2014.

[LiCh20] L.-Y. Li and C.-C. Tsai, "Students' patterns of accessing time in a text structure learning system: relationship to individual characteristics and learning performance." *Educational Technology Research & Development*, vol. 68(5), pp. 2569 – 2594, 2020.

[LiJW20] Z. Lin, A. Jain, C. Wang, G. Fanti, and V. Sekar, "Using GANs for Sharing Networked Time Series Data," *in Proceedings of the ACM Internet Measurement Conference*, ACM, 2020.

[LiFJ19] A. Lincke, D. Fellman, M. Jansen, M. Milrad, E. Berge, and B. Jonsson, "Correlating working memory capacity with learners study behavior in a web-based learning platform," *ICCE 2019 - 27th International Conference on Computers in Education, Proceedings*, vol. 1, pp. 90 – 92, 2019.

[Lind20] I. Lindgren, "Dealing with Highly Dimensional Data using Principal Component Analysis (PCA)," *Towards Data Science*, 2020.

[LoBo56] W. N. Locke and A. D. Booth, "Machine translation of languages," *American Documentation*, vol. 7(2), pp. 135–136, 1956, URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.5090070209`.

[LoBe21] E. Loginova and D. F. Benoit, "Embedding Navigation Patterns for Student Performance Prediction," *in International Conference on Educational Data Mining (EDM)*, pp. 391–399, 2021.

[MaDa10] L. P. Macfadyen and S. Dawson, "Mining LMS data to develop an "early warning system" for educators: A proof of concept," *Computers and Education*, vol. 54(2), pp. 588–599, 2010.

[MaME18] M. J. Mahzoon, M. L. Maher, O. Eltayeby, W. Dou, and K. Grace, "A Sequence Data Model for Analyzing Temporal Patterns of Student Data," *Journal of Learning Analytics*, vol. 5(1), pp. 55–74, 2018.

[MaCC21] V. Mandalapu, L. K. Chen, Z. Chen, and J. Gong, "Student-Centric Model of Login Patterns: A Case Study with Learning Management Systems," *in International Conference on Educational Data Mining (EDM)*, pp. 263–274, 2021.

[MaSG17] M. C. S. Manzanares, R. M. Sánchez, C. I. García Osorio, and J. F. Díez-Pastor, "How Do B-Learning and Learning Patterns Influence Learning Outcomes?" *Frontiers in Psychology*, vol. 8, 2017.

[MaMP20] Y. Mao, S. Marwan, T. W. Price, T. Barnes, and M. Chi, "What Time Is It? Student Modeling Needs to Know," *in Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, 2021.

[MaUW21] F. Marbouti, J. Ulas, and C.-H. Wang, "Academic and Demographic Cluster Analysis of Engineering Student Success," *IEEE Transactions on Education*, vol. 64(3), pp. 261–266, 2021.

[MaDa19] G. Marcus and E. Davis, *Rebooting AI: Building Artificial Intelligence We Can Trust*, Pantheon Books, New York, NY, 2019.

[MaWa19] B. Marr and M. Ward, *Artificial intelligence in practice: how 50 successful companies used artificial intelligence to solve problems*, John Wiley & Sons Ltd, Chichester, West Sussex, United Kingdom, 2019, ISBN 9781119548980.

[Masl21] R. Masland, *We Know It When We See It: What the Neurobiology of Vision Tells Us About How We Think*, Oneworld Publications, 2021, ISBN 9781786078179.

[Masl43] A. H. Maslow, "A Theory of Human Motivation," *Psychological Review*, vol. 50(4), pp. 370–395, 1943.

[MbMG22] E. Mbunge, J. Batani, R. Mafumbate, C. Gurajena, S. Fashoto, T. Rugube, B. Akinnuwesi, and A. Metfula, "Predicting Student Dropout in Massive Open Online Courses Using Deep Learning Models - A Systematic Review," *in Cybernetics Perspectives in Systems*, pp. 212–231, 2022.

[McMR55] J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon, "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955," *AIMag*, vol. 27(4), p. 12, 2006.

[McPi43] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5(4), pp. 115–133, 1943.

[MBKK03] B. Minaei-Bidgoli, D. Kashy, G. Kortemeyer, and W. Punch, "Predicting student performance: an application of data mining methods with an educational Web-based system," *in 33rd Annual Frontiers in Education, 2003. FIE 2003.*, vol. 1, pp. T2A–13, 2003.

[MiPa69] M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry*, MIT Press, Cambridge, MA, USA, 1969.

[Mitc19] M. Mitchell, *Artificial Intelligence: A Guide for Thinking Humans*, Farrar, Straus and Giroux, New York, NY, 2019.

[Moor65] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38(8), 1965.

[MoGB21] M. Moresi, M. J. Gomez, and L. Benotti, "Predicting Students' Difficulties from a Piece of Code," *IEEE Transactions on Learning Technologies*, vol. 14(3), pp. 386 – 399, 2021.

[MoDO16] L. Morgenstern, E. Davis, and C. L. Ortiz, "Planning, executing, and evaluating the winograd schema challenge," *The AI magazine*, vol. 37(1), pp. 50–54, 2016.

[MuAC21] A. A. Mubarak, S. A. M. Ahmed, and H. Cao, "MOOC-ASV: analytical statistical visual model of learners' interaction in videos of MOOC courses," *Interactive Learning Environments*, pp. 1–16, 2021.

[MuAM22] M. D. Mubarak, Z. S. Ameen, A. S. Mubarak, and F. Al-Turjman, "A Step Ahead Students CGPA Prediction Based on Support Vector Machines," *in 2022 International Conference on Artificial Intelligence in Everything (AIE)*, pp. 189–192, 2022.

[Natu16] Nature Publishing Group, "AlphaGo victorious," *Nature (London)*, vol. 531(7594), pp. 280–, 2016.

[Nikn17] M. Niknam, *LPR: An Adaptive Learning Path Recommendation System Using ACO & Meaningful Learning Theory*, Ph.d. thesis, University of Manitoba, 2017.

[Ntak21] T. Ntakouris, "Timeseries classification with a Transformer model," , 2021, URL `https://keras.io/examples/timeseries/timeseries_transformer_classification/`.

[Ogor07] E. N. Ogor, "Student academic performance monitoring and evaluation using data mining techniques," *in Electronics, Robotics and Automotive Mechanics Conference, CERMA 2007 - Proceedings*, pp. 354–359, 2007.

[OrVa20] F. Orji and J. Vassileva, "Using Machine Learning to Explore the Relation Between Student Engagement and Student Performance," *in 2020 24th International Conference Information Visualisation (IV)*, pp. 480–485, 2020.

[PaJO17] Y. Pang, N. Judd, J. O'Brien, and M. Ben-Avie, "Predicting students' graduation outcomes through support vector machines," *in 2017 IEEE Frontiers in Education Conference (FIE)*, pp. 1–8, 2017.

[PaML19] E. Park, F. Martin, and R. Lambert, "Examining Predictive Factors for Student Success in a Hybrid Learning Course," *Quarterly Review of Distance Education*, vol. 20(2), pp. 11–27, 2019.

[PaSe21] J. W. Paul and J. S. Cicek, "The Cognitive Science of Powerpoint," *in Proceedings 2021 Canadian Engineering Education Association (CEEA-ACEG21) Conference*, pp. 1–8, University of Prince Edward Island, 2021.

[Pavl27] I. P. Pavlov, "Conditioned reflexes: An investigation of the physiological activity of the cerebral cortex," *Annals of Neurosciences*, vol. 17(3), pp. 136–141, 2010.

[Pear88] J. Pear and W. Kinsner, "Computer-Aided Personalized System of Instruction: An effective and economical method for long-distance education," *Intern. J. Machine Mediated Learning*, vol. 2, pp. 213–237, 1988.

[PeVG11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[PeON20] S. Peng, S. Ohira, and K. Nagao, "Reading Students' Multiple Mental States in Conversation from Facial and Heart Rate Cues." *in CSEDU 2020 - Proceedings of the 12th International Conference on Computer Supported Education*, vol. 1, pp. 68–76, 2020.

[Pete69] L. J. Peter, *The Peter principle*, W. Morrow, New York, 1969.

[Pena14] A. Pea-Ayala, editor, *Educational Data Mining Applications and Trends*, Studies in Computational Intelligence, 524, Springer International Publishing, Berlin, 1st edn., 2014, ISBN 3-319-02738-7.

[PlMB10] J. Plass, R. Moreno, and R. Brunken, *Cognitive Load Theory*, Cambridge University Press, Cambridge, 2010.

[QuLW19] S. Qu, K. Li, B. Wu, X. Zhang, and K. Zhu, "Predicting Student Performance and Deficiency in Mastering Knowledge Points in MOOCs Using Multi-Task Learning," *Entropy*, vol. 21(12), 2019.

[ReRa17] D. K. Renuka and P. V. S. Rajamohana, "An ensembled classifier for email spam classification in hadoop environment," *Applied Mathematics & Information Sciences*, vol. 11(4), pp. 1123–1128, 2017.

[RGPO21] M. Riestra-Gonzlez, M. del Puerto Paule-Ruz, and F. Ortin, "Massive LMS log data analysis for the early prediction of course-agnostic student performance," *Computers & Education*, vol. 163, 2021.

[RoTB21] N. M. Rodrigues, J. E. Batista, L. Trujillo, B. Duarte, M. Giacobini, L. Vanneschi, and S. Silva, "Plotting time: On the usage of CNNs for time series classification," , 2021.

[Roge79] C. Rogers, *Freedon to Learn*, Studies of the person, Merrill, 1979, ISBN 9780675010429.

[RoEZ10] C. Romero, P. G. Espejo, A. Zafra, J. R. Romero, and S. Ventura, "Web usage mining for predicting final marks of students that use Moodle courses," *Computer Applications in Engineering Education*, vol. 21(1), pp. 135–146, 2010.

[RoVe10] C. Romero and S. Ventura, "Educational data mining: A review of the state of the art," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40(6), p. 601618, 2010.

[RoVP11] C. Romero, S. Ventura, M. Pechenizkiy, and R. S. Baker, editors, *Handbook Of Educational Data Mining*, CRC, Boca Raton, FL, 2011.

[Ronc19] A. Roncin, *Development Of A Video Game To Teaching Engineering Ethics In Canada*, Ph.d. thesis, University of Manitoba, 2019.

[Rose58] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain." *Psychological Review*, vol. 65(6), pp. 386–408, 1958.

[Rose62] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Spartan Books, Washington, DC, 1962.

[RuHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323(6088), pp. 533–536, 1986.

[Sado19] L. Sadouk, "CNN approaches for time series classification," *Time series analysis-data, methods, and applications*, vol. 5, 2019.

[SaFD18] A. Sarra, L. Fontanella, and S. Di Zio, "Identifying Students at Risk of Academic Failure Within the Educational Data Mining Framework," *Soc Indic Res*, vol. 146, p. 4160, 2019.

[ScLL96] J. Schaeffer, R. Lake, P. Lu, and M. Bryant, "Chinook : The world man-machine checkers champion," *The AI magazine*, vol. 17(1), pp. 21–29, 1996.

[Scik22] sci-kit learn developers, "Selecting the number of clusters with silhouette analysis on KMeans clustering," , 2022, URL `https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html#`.

[SFMB21] J. Seniuk-Cicek, M. Friesen, D. Mann, N. Balakrishnan, R. B. Rodrigues, and J. Pail, "The Graduate Specialization in Engineering Education," *in Proceedings of the Canadian Engineering Education Association, (CEEA2021)*, PEI, 2021.

[Shor75] E. H. Shortliffe and B. G. Buchanan, "A model of inexact reasoning in medicine," *Mathematical Biosciences*, vol. 23(3), pp. 351–379, 1975.

[SiAd13] N. Siddique and H. Adeli, *Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing*, John Wiley and Sons Ltd, West Sussex, UK, 2013.

[SiSL20] C. A. Siebra, R. N. Santos, and N. C. Q. Lino, "A Self-Adjusting Approach for Temporal Dropout Prediction of E-Learning Students," *International Journal of Distance Education Technologies*, vol. 18(2), pp. 19–33, 2020.

[SiHS17] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm," , 2017.

[SiSS17] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre,

G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of Go without human knowledge," *Nature*, vol. 550(7676), pp. 354–359, 2017.

[SiCa15] T. Sinha and J. Cassell, "Connecting the Dots: Predicting Student Grade Sequences from Bursty MOOC Interactions over Time," *in Proceedings of the Second (2015) ACM Conference on Learning Scale*, pp. 249–252, 2015.

[Skin38] B. F. Skinner, *The Behavior of Organisms: An Experimental Analysis*, Appleton-Century-Crofts, New York, NY, 1938, 457 pp.

[SoOk20] P. Sokkhey and T. Okazaki, "Developing web-based support systems for predicting poor-performing students using educational data mining techniques," *International Journal of Advanced Computer Science and Applications*, vol. 11(7), pp. 23–32, 2020.

[Spit21] M. W. H. Spitzer, "Just do it! Study time increases mathematical achievement scores for grade 4-10 students in a large longitudinal cross-country study," *European Journal of Psychology of Education*, vol. 37(1), pp. 39–53, 2022.

[Stem91] M. Stember, "Advancing the social sciences through the interdisciplinary enterprise," *The Social Science Journal*, vol. 28(1), pp. 1–14, 1991.

[SwFr17] M. R. Swaine and P. A. Freiberger, "Step Reckoner," , 2017, URL `https://www.britannica.com/technology/Step-Reckoner`.

[TeBP19] M. Tellakat, R. L. Boyd, and J. W. Pennebaker, "How do online learners study? The psychometrics of students' clicking patterns in online courses," *PLOS ONE*, vol. 14(3), pp. 1–17, 2019.

[TeRN19] D. Tempelaar, B. Rienties, and Q. Nguyen, "Learning engagement, learning outcomes and learning gains: Lessons from la," *in 16th International Conference on Cognition and Exploratory Learning in Digital Age, CELDA 2019*, pp. 257–264, 2019.

[Theo19] O. Theobald, *Machine Learning with Python*, Scatterplot Press, Middletown, DE, 2019, 151 pp.

[ThPA13] R. Therón, F. J. G. Peñalvo, and D. A. Gómez Aguilar, "Reveal the Relationships among Students Participation and Their Outcomes on E-Learning Environments: Case Study," *in 2013 IEEE 13th International Conference on Advanced Learning Technologies*, pp. 443–447, 2013.

[Thor13] E. L. Thorndike, *Educational Psychology: Vol. II, The Psychology of Learning*, Teachers College, Columbia University, New York, 1913, 54-55 pp.

[Thor54] L. P. Thorpe and A. M. Schmuller, *Contemporary theories of learning with applications to education and psychology*, Ronald Press Co., New York, 1954.

[TiLW20] H. Tian, S. Lai, and F. Wu, "A Prediction Framework of Learning Outcomes Based on Meaningful Learning Features," *in 2020 Ninth International Conference of Educational Innovation through Technology (EITT)*, pp. 205–210, 2020.

[Turi36] A. M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society*, vol. 2(42), pp. 230–265, 1936, URL `http://www.cs.helsinki.fi/u/gionis/cc05/OnComputableNumbers.pdf`.

[Turi50] A. M. Turing, "Computing Machinery and Intelligence," *Mind*, vol. 59(236), pp. 433–460, 1950, URL `http://www.jstor.org/stable/2251299`.

[VaBD20] S. Van Goidsenhoven, D. Bogdanova, G. Deeva, S. v. Broucke, J. De Weerdt, and M. Snoeck, "Predicting Student Success in a Blended Learning Environment," *in Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, pp. 17–25, 2020.

[VaSP17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *in 31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 2017.

[voMo44] J. von Neumann and O. Morgenstern, *Theory of games and economic behavior*, Princeton University Press, Princeton, NJ, 1944.

[WaTS07] C. Wade, C. Tavris, D. Saucier, and L. Elias, *Psychology*, Pearson Education Canada Inc., Toronto, ON, 2nd edn., 2007, 758 pp.

[Wang21] F. H. Wang, "Interpreting log data through the lens of learning design: Second-order predictors and their relations with learning outcomes in flipped classrooms," *Computers & Education*, vol. 168, 2021.

[Wang14] V. C. X. Wang, *Handbook of Research on Education and Technology in a Changing Society*, IGI Global, Florida Atlantic University, USA, 2014, 1217 pp.

[WaYO17] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," *in 2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1578–1585, IEEE, 2017, ISBN 9781509061822.

[WaSh16] K. Warwick and H. Shah, "Can machines think? A report on Turing test experiments at the Royal Society," *Journal of experimental & theoretical artificial intelligence*, vol. 28(6), pp. 989–1007, 2016.

[Wass89] P. D. Wasserman, *Neural Computing: Theory and Practice*, Van Nostrand Reinhold, New York, NY, 1989, ISBN 0-442-20743-3, 230 pp.

[Wats13] J. B. Watson, *Psychology as the Behaviorist Views It*, Bobbs-Merrill, Indianapolis, IL, 1913, ISBN 32044028753861, 158-177 pp.

[Wats08] R. Watson, J. Coulter, W. Sharrock, A. Dennis, R. Read, G. Button, and R. Hamilton, "Cognitivism," *Theory, culture & society*, vol. 25(2), 2008.

[Weiz66] J. Weizenbaum, "ELIZA a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9(1), pp. 36–45, 1966.

[WeZZ22] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, "Transformers in time series: A survey," *arXiv preprint arXiv:2202.07125*, 2022.

[WiRa15] U. Wilensky and W. Rand, *An introduction to agent-based modeling: Modeling natural, social and engineered complex systems with NetLogo*, MIT Press, Cambridge, MA, 2015, 505 pp.

[Will19] D. L. Williams, *Predicting student success using digital text- book analytics in online courses*, Phd thesis, Liberty University, 2019.

[Wino71] T. Winograd, *Procedures as a Representation for Data in a Computer Program for Understanding Natural Language*, M.I.T. Project MAC, 1971, 472 pp.

[Wool20] M. Woolridge, *A Brief History of Artificial Intelligence: What It Is, Where We Are, and Where We Are Going*, Flatiron Books, New York, NY, 2020, 148 pp.

[WuSC16] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, . Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation," , 2016.

[XiGP15] W. Xing, R. Guo, E. Petakovic, and S. Goggins, "Participation-based student final performance prediction model through interpretable Genetic Programming: Integrating learning analytics, educational data mining and theory," *Computers in Human Behavior*, vol. 47, pp. 168–181, 2015, URL `https://www.sciencedirect.com/science/article/pii/S0747563214004865`.

[You16] J. W. You, "Identifying significant indicators using LMS data to predict course achievement in online learning," *The Internet and Higher Education*, vol. 29, pp. 23–30, 2016.

[YuWu21] C.-C. Yu and Y. L. Wu, "Early Warning System for Online STEM LearningA Slimmer Approach Using Recurrent Neural Networks," *Sustainability*, vol. 13(22), 2021.

[YuLF20] R. Yu, Q. Li, C. Fischer, S. Doroudi, and D. Xu, "Towards accurate and fair prediction of college success: Evaluating different sources of student data," *International educational data mining society*, 2020.

[YuPS19] R. Yu, Z. Pardos, and J. Scott, "Student behavioral embeddings and their relationship to outcomes in a collaborative online course," *in CEUR Workshop Proceedings*, vol. 2592, pp. 23–29, 2020.

[Zach15] N. Z. Zacharis, "A multivariate approach to predicting student outcomes in web-enabled blended learning courses," *The Internet and Higher Education*, vol. 27, pp. 44–53, 2015, URL `https://www.sciencedirect.com/science/article/pii/S1096751615000391`.

[Zach18] N. Z. Zacharis, "Classification and Regression Trees (CART) for Predictive Modeling in Blended Learning," *I.J. Intelligent Systems and Applications*, vol. 3, pp. 1–9, 2018.

[ZhKK18] C. Zhang, S. R. Kuppannagari, R. Kannan, and V. K. Prasanna, "Generative Adversarial Network for Synthetic Time Series Data Generation in Smart Grids," *in 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 1–6, Aalborg, Denmark, 2018.

[ZhUD22] G. Zhou, T. Umada, and S. D'Mello, "What Do Students' Interactions with Online Lecture Videos Reveal about Their Learning?" *in Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*, pp. 295–305, 2022.

There are 188 references in this thesis that range in publication date from 1847-2022.

Appendix A

Cognitive Digital Twins

Although the field of CDTs is still in its infancy, there is great potential for it to help prepare students for the demands of the future workforce. There are three main motivating factors behind bringing CDTs from the realm of manufacturing into education. The first is that todays students are not just training for one career, but they are studying to become the future workforce that will need to be capable of adapting to new jobs. Most people will undergo multiple career changes throughout their lifetimes, by choice or as the availability of jobs changes. The Prussian education model, that has influenced education since the 18th century, will no longer be sufficient to train workers for a set of tasks. The second factor is that CDTs can be predictive, aiding student's with personal recommendations to learn new material and suggestions to reinforce previously studied concepts. The third factor is that it is becoming increasingly difficult for learners to stay abreast of current developments in their field because the sum total of human knowledge is increasing exponentially, and information is also becoming irrelevant more quickly.

CDTs could help to address these factors by curating knowledge for the learner and performing up-to-date error checking to identify and remove erroneous or outdated information, also called "knowledge fusion" and "knowledge sunsetting" respectively. Such a personalised system could also aid learners in their work by detecting plagiarism and verifying sources, or "knowledge vetting." Further, CDTs could provide *intellectual property* (IP) management tools, to separate public and private information [Kins19].

Companies and institutions, including educational institutions already collect vast amounts of data about individuals. These data are the "distributed elements of our digital twins" [Kins19],

some of which could be pulled together and made to benefit the individual through CDTs. In order to be support the individual and their learning, a digital twin in an educational context should "represent our current skills, knowledge, and wisdom"; support knowledge retention by accounting for the loss of skills and knowledge over time without practice; and curate an educational plan that is proactive.

As outlined in [Kins19], an individual and their CDT would be considered a symbion, and the CDT would consist of three main parts: (i) a digital model as a "mirror of our knowledge"; (ii) a digital shadow, that "can learn what we learn; and (iii) digital threads, that know "the way we learn" and "can support new learning." However, before CDTs can be realized, there is still work to do to better understand how we learn so that the process can be modelled and accurately reflected in the CDTs.

Appendix B

Literature Review Search

A literature review search was conducted on ML in Education for prediction in ERIC, PsycInfo, Scopus, and Compendex (Engineering Village) databases.

Titles, abstracts, and keywords were searched on ERIC using the terms ("machine learning" or "artificial intelligence" or ai or predict* or supervise* or unsupervise* or cluster* or "feature engineering" or "feature selection") AND (temporal or time-based or interval or "inter-event time" or pattern or burstiness) AND ("learning management system" or lms or "learning management systems" or "learning system" or "learning systems" or "remote delivery" or "massive open online course" or mooc or moocs or "massive open online courses" or "data hub" or brightspace or d2l or "education software" or elearning or e-learning or synchronous or blackboard) AND (student near/3 outcome* or course near/3 outcome* or grade or pass or fail or "competency level" or "learning objective" or "learning objectives" or "competency levels"). The search returned 46 results.

Titles, abstracts, heading words, tables of contents, and key concepts in PsychInfo were searched using the similar keywords (machine learning or artificial intelligence or ai or predict* or supervise* or unsupervise* or cluster* or feature engineering or feature selection) AND (temporal or time* or interval* or pattern* or burstiness) AND (learning management system or lms or learning management systems or learning system or learning systems or remote delivery or massive open online course or mooc or moocs or massive open online courses or data hub or brightspace or d2l or education software or elearning or e-learning or synchronous or blackboard) AND ((student adj3 outcome*) or (course adj3 outcome*) or grade or pass or fail or competency level or learning objective or learning objectives or

competency levels). The search returned 71 results. A similar search of the titles, abstracts, and keywords was conducted in Scopus, which returned 155 results.

Titles abstracts, and keywords were searched in Compendex (Engineering Village) using the similar keywords (temporal OR inter-event or interevent or interval* or pattern* or burstiness) AND (learning management system or lms or learning management systems or learning system or learning systems or remote delivery or data hub or brightspace or d2l or education software or elearning or e-learning or synchronous or blackboard) AND ((student near/3 outcome) or (course near/3 outcome) or (student near/3 outcomes) or (course near/3 outcomes) or grade or pass or fail or competency level or learning objective or learning objectives or competency levels) AND (machine learning or artificial intelligence or ai or predict* or supervise* or unsupervise* or cluster* or feature engineering or feature selection). The search returned 103 results. The inclusion of "time*" as a keyword yielded 3400 results, of which 480 included in the category of 'Students' and 118 were included in the category of 'Education Computing'.

A first combined keyword and subject headings search was further conducted in Compendex (Engineering Village) which returned 28 results using the keywords (interevent OR inter-event OR temporal OR pattern* OR burstiness) combined with the subject headings (Machine learning AND Education computing AND Students).

A second combined keyword and subject headings search in Compendex (Engineering Village) returned 620 results using the keywords ((interevent or inter-event or temporal or pattern* or burstiness) combined with the subject headings ((artificial intelligence) or (machine learning)) AND (e-learning or education computing) AND (students)).

All searches were from the database inception to November 2022. Each keyword search included some results directed to other fields, most commonly medicine and medical education.

Inclusion/exclusion criteria for the search were as follows:

- In order to be present in the review, the studies required the following elements: the course or courses in the study were undergraduate level; the data was collected from an online or remote learning course; the study contained data collected through a LMS; the LMS data included timestamps, time intervals, or other time information; and the goal of the study was student course outcome prediction in the form of pass/fail or final grade prediction.

- Studies with the following elements were excluded: courses in the study were at the grade school level; the data was collected from a MOOC; studies that included specialized data collection software not commonly available in an LMS, such as mouse-click or eye tracking software; studies that used surveys or student journals as part of the data sets; or ML techniques were used in the study for course dropout prediction.

Previously, a hand search of the UM Libraries, Google Scholar, Google, IEEE, and CEEA-ACÉG and EDM conference proceedings (Canadian Engineering Education Association-Association Canadienne de léducation en génie and Educational Data Mining) was conducted broadly on artificial intelligence, machine learning, and educational learning theories, and more narrowly on machine learning in education and machine learning for student outcome prediction.

Appendix C

Correlation and p values

In all of Sections C.2-C.7, p values between 0.01 and 0.05 are highlighted in yellow and p values greater than 0.05 are highlighted in red.

C.1 Correlation and p values for features at the first quarter in the term

Feature 1	Feature 2	Correlation	p value
Num Interactions (1/4)	Num Timeline Items (1/4)	0.835	6.712e-82
Num Interactions (1/4)	Fraction NonZero Days (1/4)	0.593	8.008e-31
Num Interactions (1/4)	Mean Interval Length S (1/4)	-0.4992	6.162e-21
Num Interactions (1/4)	Var Intervals Length H (1/4)	-0.2101	0.0001952
Num Interactions (1/4)	Skewness Intervals H (1/4)	0.4172	1.755e-14
Num Interactions (1/4)	Kurtosis Intervals H (1/4)	0.3813	3.627e-12
Num Interactions (1/4)	Burstiness (1/4)	0.05235	0.3583
Num Interactions (1/4)	Fraction of Midterm time	0.1828	0.001223
Num Interactions (1/4)	Assign Hand In Avg (1/4)	0.3155	1.357e-08
Num Interactions (1/4)	Midterm Grade	0.1371	0.01569
Num Interactions (1/4)	Final Grade	0.2007	0.0003768
Num Timeline Items (1/4)	Num Interactions (1/4)	0.835	6.712e-82
Num Timeline Items (1/4)	Fraction NonZero Days (1/4)	0.5508	5.469e-26

Num Timeline Items (1/4)	Mean Interval Length S (1/4)	-0.5774	5.858e-29
Num Timeline Items (1/4)	Var Intervals Length H (1/4)	-0.2305	4.166e-05
Num Timeline Items (1/4)	Skewness Intervals H (1/4)	0.5289	9.607e-24
Num Timeline Items (1/4)	Kurtosis Intervals H (1/4)	0.4692	2.253e-18
Num Timeline Items (1/4)	Burstiness (1/4)	0.08099	0.1549
Num Timeline Items (1/4)	Fraction of Midterm time	0.2762	7.821e-07
Num Timeline Items (1/4)	Assign Hand In Avg (1/4)	0.3074	3.306e-08
Num Timeline Items (1/4)	Midterm Grade	0.2264	5.75e-05
Num Timeline Items (1/4)	Final Grade	0.248	9.917e-06
Fraction NonZero Days (1/4)	Num Interactions (1/4)	0.593	8.008e-31
Fraction NonZero Days (1/4)	Num Timeline Items (1/4)	0.5508	5.469e-26
Fraction NonZero Days (1/4)	Mean Interval Length S (1/4)	-0.484	1.32e-19
Fraction NonZero Days (1/4)	Var Intervals Length H (1/4)	-0.3063	3.718e-08
Fraction NonZero Days (1/4)	Skewness Intervals H (1/4)	0.09064	0.1112
Fraction NonZero Days (1/4)	Kurtosis Intervals H (1/4)	0.1017	0.07371
Fraction NonZero Days (1/4)	Burstiness (1/4)	0.1047	0.06556
Fraction NonZero Days (1/4)	Fraction of Midterm time	0.3507	2.108e-10
Fraction NonZero Days (1/4)	Assign Hand In Avg (1/4)	0.2386	2.172e-05
Fraction NonZero Days (1/4)	Midterm Grade	0.1805	0.001415
Fraction NonZero Days (1/4)	Final Grade	0.3316	2.16e-09
Mean Interval Length S (1/4)	Num Interactions (1/4)	-0.4992	6.162e-21
Mean Interval Length S (1/4)	Num Timeline Items (1/4)	-0.5774	5.858e-29
Mean Interval Length S (1/4)	Fraction NonZero Days (1/4)	-0.484	1.32e-19
Mean Interval Length S (1/4)	Var Intervals Length H (1/4)	0.8115	7.442e-74
Mean Interval Length S (1/4)	Skewness Intervals H (1/4)	-0.3278	3.39e-09
Mean Interval Length S (1/4)	Kurtosis Intervals H (1/4)	-0.2879	2.494e-07
Mean Interval Length S (1/4)	Burstiness (1/4)	0.1211	0.03306
Mean Interval Length S (1/4)	Fraction of Midterm time	-0.3367	1.181e-09
Mean Interval Length S (1/4)	Assign Hand In Avg (1/4)	-0.1962	0.0005115
Mean Interval Length S (1/4)	Midterm Grade	-0.2587	3.931e-06
Mean Interval Length S (1/4)	Final Grade	-0.3227	6.037e-09
Var Intervals Length H (1/4)	Num Interactions (1/4)	-0.2101	0.0001952
Var Intervals Length H (1/4)	Num Timeline Items (1/4)	-0.2305	4.166e-05

Var Intervals Length H (1/4)	Fraction NonZero Days (1/4)	-0.3063	3.718e-08
Var Intervals Length H (1/4)	Mean Interval Length S (1/4)	0.8115	7.442e-74
Var Intervals Length H (1/4)	Skewness Intervals H (1/4)	-0.04367	0.4435
Var Intervals Length H (1/4)	Kurtosis Intervals H (1/4)	-0.05451	0.3388
Var Intervals Length H (1/4)	Burstiness (1/4)	0.05138	0.3673
Var Intervals Length H (1/4)	Fraction of Midterm time	-0.2869	2.742e-07
Var Intervals Length H (1/4)	Assign Hand In Avg (1/4)	-0.1196	0.03527
Var Intervals Length H (1/4)	Midterm Grade	-0.2069	0.0002451
Var Intervals Length H (1/4)	Final Grade	-0.2584	4.049e-06
Skewness Intervals H (1/4)	Num Interactions (1/4)	0.4172	1.755e-14
Skewness Intervals H (1/4)	Num Timeline Items (1/4)	0.5289	9.607e-24
Skewness Intervals H (1/4)	Fraction NonZero Days (1/4)	0.09064	0.1112
Skewness Intervals H (1/4)	Mean Interval Length S (1/4)	-0.3278	3.39e-09
Skewness Intervals H (1/4)	Var Intervals Length H (1/4)	-0.04367	0.4435
Skewness Intervals H (1/4)	Kurtosis Intervals H (1/4)	0.9587	3.247e-170
Skewness Intervals H (1/4)	Burstiness (1/4)	0.1935	0.0006153
Skewness Intervals H (1/4)	Fraction of Midterm time	0.006913	0.9035
Skewness Intervals H (1/4)	Assign Hand In Avg (1/4)	0.08862	0.1194
Skewness Intervals H (1/4)	Midterm Grade	0.01971	0.7296
Skewness Intervals H (1/4)	Final Grade	0.0271	0.6346
Kurtosis Intervals H (1/4)	Num Interactions (1/4)	0.3813	3.627e-12
Kurtosis Intervals H (1/4)	Num Timeline Items (1/4)	0.4692	2.253e-18
Kurtosis Intervals H (1/4)	Fraction NonZero Days (1/4)	0.1017	0.07371
Kurtosis Intervals H (1/4)	Mean Interval Length S (1/4)	-0.2879	2.494e-07
Kurtosis Intervals H (1/4)	Var Intervals Length H (1/4)	-0.05451	0.3388
Kurtosis Intervals H (1/4)	Skewness Intervals H (1/4)	0.9587	3.247e-170
Kurtosis Intervals H (1/4)	Burstiness (1/4)	0.1122	0.04839
Kurtosis Intervals H (1/4)	Fraction of Midterm time	-0.01466	0.7971
Kurtosis Intervals H (1/4)	Assign Hand In Avg (1/4)	0.05471	0.337
Kurtosis Intervals H (1/4)	Midterm Grade	0.01014	0.8588
Kurtosis Intervals H (1/4)	Final Grade	0.01787	0.7541
Burstiness (1/4)	Num Interactions (1/4)	0.05235	0.3583
Burstiness (1/4)	Num Timeline Items (1/4)	0.08099	0.1549

Burstiness (1/4)	Fraction NonZero Days (1/4)	0.1047	0.06556
Burstiness (1/4)	Mean Interval Length S (1/4)	0.1211	0.03306
Burstiness (1/4)	Var Intervals Length H (1/4)	0.05138	0.3673
Burstiness (1/4)	Skewness Intervals H (1/4)	0.1935	0.0006153
Burstiness (1/4)	Kurtosis Intervals H (1/4)	0.1122	0.04839
Burstiness (1/4)	Fraction of Midterm time	0.1675	0.003099
Burstiness (1/4)	Assign Hand In Avg (1/4)	0.04464	0.4336
Burstiness (1/4)	Midterm Grade	0.1126	0.04762
Burstiness (1/4)	Final Grade	0.1141	0.04473
Fraction of Midterm time	Num Interactions (1/4)	0.1828	0.001223
Fraction of Midterm time	Num Timeline Items (1/4)	0.2762	7.821e-07
Fraction of Midterm time	Fraction NonZero Days (1/4)	0.3507	2.108e-10
Fraction of Midterm time	Mean Interval Length S (1/4)	-0.3367	1.181e-09
Fraction of Midterm time	Var Intervals Length H (1/4)	-0.2869	2.742e-07
Fraction of Midterm time	Skewness Intervals H (1/4)	0.006913	0.9035
Fraction of Midterm time	Kurtosis Intervals H (1/4)	-0.01466	0.7971
Fraction of Midterm time	Burstiness (1/4)	0.1675	0.003099
Fraction of Midterm time	Assign Hand In Avg (1/4)	-0.004458	0.9377
Fraction of Midterm time	Midterm Grade	0.5811	2.168e-29
Fraction of Midterm time	Final Grade	0.5539	2.544e-26
Assign Hand In Avg (1/4)	Num Interactions (1/4)	0.3155	1.357e-08
Assign Hand In Avg (1/4)	Num Timeline Items (1/4)	0.3074	3.306e-08
Assign Hand In Avg (1/4)	Fraction NonZero Days (1/4)	0.2386	2.172e-05
Assign Hand In Avg (1/4)	Mean Interval Length S (1/4)	-0.1962	0.0005115
Assign Hand In Avg (1/4)	Var Intervals Length H (1/4)	-0.1196	0.03527
Assign Hand In Avg (1/4)	Skewness Intervals H (1/4)	0.08862	0.1194
Assign Hand In Avg (1/4)	Kurtosis Intervals H (1/4)	0.05471	0.337
Assign Hand In Avg (1/4)	Burstiness (1/4)	0.04464	0.4336
Assign Hand In Avg (1/4)	Fraction of Midterm time	-0.004458	0.9377
Assign Hand In Avg (1/4)	Midterm Grade	0.09139	0.1083
Assign Hand In Avg (1/4)	Final Grade	0.1066	0.06084
Midterm Grade	Num Interactions (1/4)	0.1371	0.01569
Midterm Grade	Num Timeline Items (1/4)	0.2264	5.75e-05

Midterm Grade	Fraction NonZero Days (1/4)	0.1805	0.001415
Midterm Grade	Mean Interval Length S (1/4)	-0.2587	3.931e-06
Midterm Grade	Var Intervals Length H (1/4)	-0.2069	0.0002451
Midterm Grade	Skewness Intervals H (1/4)	0.01971	0.7296
Midterm Grade	Kurtosis Intervals H (1/4)	0.01014	0.8588
Midterm Grade	Burstiness (1/4)	0.1126	0.04762
Midterm Grade	Fraction of Midterm time	0.5811	2.168e-29
Midterm Grade	Assign Hand In Avg (1/4)	0.09139	0.1083
Midterm Grade	Final Grade	0.7792	1.942e-64
Final Grade	Num Interactions (1/4)	0.2007	0.0003768
Final Grade	Num Timeline Items (1/4)	0.248	9.917e-06
Final Grade	Fraction NonZero Days (1/4)	0.3316	2.16e-09
Final Grade	Mean Interval Length S (1/4)	-0.3227	6.037e-09
Final Grade	Var Intervals Length H (1/4)	-0.2584	4.049e-06
Final Grade	Skewness Intervals H (1/4)	0.0271	0.6346
Final Grade	Kurtosis Intervals H (1/4)	0.01787	0.7541
Final Grade	Burstiness (1/4)	0.1141	0.04473
Final Grade	Fraction of Midterm time	0.5539	2.544e-26
Final Grade	Assign Hand In Avg (1/4)	0.1066	0.06084
Final Grade	Midterm Grade	0.7792	1.942e-64

C.2 Correlation and p values for features at the midterm

Feature 1	Feature 2	Correlation	p value
Num Interactions (test)	Num Timeline Items (test)	0.8056	5.439e-72
Num Interactions (test)	Fraction NonZero Days (test)	0.6047	2.678e-32
Num Interactions (test)	Mean Interval Length S (test)	-0.4657	4.346e-18
Num Interactions (test)	Var Intervals Length H (test)	-0.1948	0.0005612
Num Interactions (test)	Skewness Intervals H (test)	0.4861	8.646e-20
Num Interactions (test)	Kurtosis Intervals H (test)	0.4436	2.219e-16
Num Interactions (test)	Burstiness (test)	0.06416	0.26
Num Interactions (test)	Fraction of Midterm time	0.2669	1.87e-06
Num Interactions (test)	Assign Hand In Avg (test)	0.07222	0.2048
Num Interactions (test)	Midterm Grade	0.1639	0.003817
Num Interactions (test)	Final Grade	0.2605	3.346e-06
Num Timeline Items (test)	Num Interactions (test)	0.8056	5.439e-72
Num Timeline Items (test)	Fraction NonZero Days (test)	0.5595	6.266e-27
Num Timeline Items (test)	Mean Interval Length S (test)	-0.5251	2.279e-23
Num Timeline Items (test)	Var Intervals Length H (test)	-0.2132	0.0001557
Num Timeline Items (test)	Skewness Intervals H (test)	0.5584	8.308e-27
Num Timeline Items (test)	Kurtosis Intervals H (test)	0.4866	7.896e-20
Num Timeline Items (test)	Burstiness (test)	0.09162	0.1074
Num Timeline Items (test)	Fraction of Midterm time	0.3507	2.121e-10
Num Timeline Items (test)	Assign Hand In Avg (test)	0.03148	0.5809
Num Timeline Items (test)	Midterm Grade	0.27	1.41e-06
Num Timeline Items (test)	Final Grade	0.3392	8.725e-10
Fraction NonZero Days (test)	Num Interactions (test)	0.6047	2.678e-32
Fraction NonZero Days (test)	Num Timeline Items (test)	0.5595	6.266e-27
Fraction NonZero Days (test)	Mean Interval Length S (test)	-0.5148	2.252e-22
Fraction NonZero Days (test)	Var Intervals Length H (test)	-0.3326	1.934e-09
Fraction NonZero Days (test)	Skewness Intervals H (test)	0.07895	0.1655
Fraction NonZero Days (test)	Kurtosis Intervals H (test)	0.08253	0.1472
Fraction NonZero Days (test)	Burstiness (test)	0.114	0.04496
Fraction NonZero Days (test)	Fraction of Midterm time	0.4511	6.001e-17
Fraction NonZero Days (test)	Assign Hand In Avg (test)	0.2263	5.811e-05

Fraction NonZero Days (test)	Midterm Grade	0.2653	2.165e-06
Fraction NonZero Days (test)	Final Grade	0.4188	1.355e-14
Mean Interval Length S (test)	Num Interactions (test)	-0.4657	4.346e-18
Mean Interval Length S (test)	Num Timeline Items (test)	-0.5251	2.279e-23
Mean Interval Length S (test)	Fraction NonZero Days (test)	-0.5148	2.252e-22
Mean Interval Length S (test)	Var Intervals Length H (test)	0.7324	2.375e-53
Mean Interval Length S (test)	Skewness Intervals H (test)	-0.2694	1.483e-06
Mean Interval Length S (test)	Kurtosis Intervals H (test)	-0.2002	0.0003896
Mean Interval Length S (test)	Burstiness (test)	0.1046	0.06599
Mean Interval Length S (test)	Fraction of Midterm time	-0.4142	2.795e-14
Mean Interval Length S (test)	Assign Hand In Avg (test)	-0.1649	0.003601
Mean Interval Length S (test)	Midterm Grade	-0.2975	9.363e-08
Mean Interval Length S (test)	Final Grade	-0.3553	1.178e-10
Var Intervals Length H (test)	Num Interactions (test)	-0.1948	0.0005612
Var Intervals Length H (test)	Num Timeline Items (test)	-0.2132	0.0001557
Var Intervals Length H (test)	Fraction NonZero Days (test)	-0.3326	1.934e-09
Var Intervals Length H (test)	Mean Interval Length S (test)	0.7324	2.375e-53
Var Intervals Length H (test)	Skewness Intervals H (test)	-0.0145	0.7993
Var Intervals Length H (test)	Kurtosis Intervals H (test)	-0.02219	0.6972
Var Intervals Length H (test)	Burstiness (test)	0.05085	0.3723
Var Intervals Length H (test)	Fraction of Midterm time	-0.3057	3.977e-08
Var Intervals Length H (test)	Assign Hand In Avg (test)	-0.1098	0.05351
Var Intervals Length H (test)	Midterm Grade	-0.2174	0.0001138
Var Intervals Length H (test)	Final Grade	-0.2573	4.462e-06
Skewness Intervals H (test)	Num Interactions (test)	0.4861	8.646e-20
Skewness Intervals H (test)	Num Timeline Items (test)	0.5584	8.308e-27
Skewness Intervals H (test)	Fraction NonZero Days (test)	0.07895	0.1655
Skewness Intervals H (test)	Mean Interval Length S (test)	-0.2694	1.483e-06
Skewness Intervals H (test)	Var Intervals Length H (test)	-0.0145	0.7993
Skewness Intervals H (test)	Kurtosis Intervals H (test)	0.9411	4.74e-147
Skewness Intervals H (test)	Burstiness (test)	0.1987	0.0004321
Skewness Intervals H (test)	Fraction of Midterm time	0.07251	0.2029
Skewness Intervals H (test)	Assign Hand In Avg (test)	-0.004767	0.9334

Skewness Intervals H (test)	Midterm Grade	0.001062	0.9851
Skewness Intervals H (test)	Final Grade	0.02773	0.6267
Kurtosis Intervals H (test)	Num Interactions (test)	0.4436	2.219e-16
Kurtosis Intervals H (test)	Num Timeline Items (test)	0.4866	7.896e-20
Kurtosis Intervals H (test)	Fraction NonZero Days (test)	0.08253	0.1472
Kurtosis Intervals H (test)	Mean Interval Length S (test)	-0.2002	0.0003896
Kurtosis Intervals H (test)	Var Intervals Length H (test)	-0.02219	0.6972
Kurtosis Intervals H (test)	Skewness Intervals H (test)	0.9411	4.74e-147
Kurtosis Intervals H (test)	Burstiness (test)	0.1106	0.05177
Kurtosis Intervals H (test)	Fraction of Midterm time	0.0576	0.312
Kurtosis Intervals H (test)	Assign Hand In Avg (test)	-0.04237	0.4572
Kurtosis Intervals H (test)	Midterm Grade	0.01236	0.8285
Kurtosis Intervals H (test)	Final Grade	0.04422	0.4378
Burstiness (test)	Num Interactions (test)	0.06416	0.26
Burstiness (test)	Num Timeline Items (test)	0.09162	0.1074
Burstiness (test)	Fraction NonZero Days (test)	0.114	0.04496
Burstiness (test)	Mean Interval Length S (test)	0.1046	0.06599
Burstiness (test)	Var Intervals Length H (test)	0.05085	0.3723
Burstiness (test)	Skewness Intervals H (test)	0.1987	0.0004321
Burstiness (test)	Kurtosis Intervals H (test)	0.1106	0.05177
Burstiness (test)	Fraction of Midterm time	0.168	0.003013
Burstiness (test)	Assign Hand In Avg (test)	0.04544	0.4253
Burstiness (test)	Midterm Grade	0.1076	0.05845
Burstiness (test)	Final Grade	0.1076	0.05838
Fraction of Midterm time	Num Interactions (test)	0.2669	1.87e-06
Fraction of Midterm time	Num Timeline Items (test)	0.3507	2.121e-10
Fraction of Midterm time	Fraction NonZero Days (test)	0.4511	6.001e-17
Fraction of Midterm time	Mean Interval Length S (test)	-0.4142	2.795e-14
Fraction of Midterm time	Var Intervals Length H (test)	-0.3057	3.977e-08
Fraction of Midterm time	Skewness Intervals H (test)	0.07251	0.2029
Fraction of Midterm time	Kurtosis Intervals H (test)	0.0576	0.312
Fraction of Midterm time	Burstiness (test)	0.168	0.003013
Fraction of Midterm time	Assign Hand In Avg (test)	-0.01132	0.8427

Fraction of Midterm time	Midterm Grade	0.5811	2.168e-29
Fraction of Midterm time	Final Grade	0.5539	2.544e-26
Assign Hand In Avg (test)	Num Interactions (test)	0.07222	0.2048
Assign Hand In Avg (test)	Num Timeline Items (test)	0.03148	0.5809
Assign Hand In Avg (test)	Fraction NonZero Days (test)	0.2263	5.811e-05
Assign Hand In Avg (test)	Mean Interval Length S (test)	-0.1649	0.003601
Assign Hand In Avg (test)	Var Intervals Length H (test)	-0.1098	0.05351
Assign Hand In Avg (test)	Skewness Intervals H (test)	-0.004767	0.9334
Assign Hand In Avg (test)	Kurtosis Intervals H (test)	-0.04237	0.4572
Assign Hand In Avg (test)	Burstiness (test)	0.04544	0.4253
Assign Hand In Avg (test)	Fraction of Midterm time	-0.01132	0.8427
Assign Hand In Avg (test)	Midterm Grade	0.09233	0.1047
Assign Hand In Avg (test)	Final Grade	0.09836	0.08379
Midterm Grade	Num Interactions (test)	0.1639	0.003817
Midterm Grade	Num Timeline Items (test)	0.27	1.41e-06
Midterm Grade	Fraction NonZero Days (test)	0.2653	2.165e-06
Midterm Grade	Mean Interval Length S (test)	-0.2975	9.363e-08
Midterm Grade	Var Intervals Length H (test)	-0.2174	0.0001138
Midterm Grade	Skewness Intervals H (test)	0.001062	0.9851
Midterm Grade	Kurtosis Intervals H (test)	0.01236	0.8285
Midterm Grade	Burstiness (test)	0.1076	0.05845
Midterm Grade	Fraction of Midterm time	0.5811	2.168e-29
Midterm Grade	Assign Hand In Avg (test)	0.09233	0.1047
Midterm Grade	Final Grade	0.7792	1.942e-64
Final Grade	Num Interactions (test)	0.2605	3.346e-06
Final Grade	Num Timeline Items (test)	0.3392	8.725e-10
Final Grade	Fraction NonZero Days (test)	0.4188	1.355e-14
Final Grade	Mean Interval Length S (test)	-0.3553	1.178e-10
Final Grade	Var Intervals Length H (test)	-0.2573	4.462e-06
Final Grade	Skewness Intervals H (test)	0.02773	0.6267
Final Grade	Kurtosis Intervals H (test)	0.04422	0.4378
Final Grade	Burstiness (test)	0.1076	0.05838
Final Grade	Fraction of Midterm time	0.5539	2.544e-26

Final Grade	Assign Hand In Avg (test)	0.09836	0.08379
Final Grade	Midterm Grade	0.7792	1.942e-64

C.3 Correlation and p values for features at the second quarter in the term

Feature 1	Feature 2	Correlation	p value
Num Interactions (1/2)	Num Timeline Items (1/2)	0.7884	6.022e-67
Num Interactions (1/2)	Fraction NonZero Days (1/2)	0.5983	1.722e-31
Num Interactions (1/2)	Mean Interval Length S (1/2)	-0.4054	1.084e-13
Num Interactions (1/2)	Var Intervals Length H (1/2)	-0.2054	0.000272
Num Interactions (1/2)	Skewness Intervals H (1/2)	0.1768	0.001775
Num Interactions (1/2)	Kurtosis Intervals H (1/2)	0.1588	0.005059
Num Interactions (1/2)	Burstiness (1/2)	0.05553	0.3298
Num Interactions (1/2)	Fraction of Midterm time	0.2343	3.09e-05
Num Interactions (1/2)	Assign Hand In Avg (1/2)	0.1708	0.002553
Num Interactions (1/2)	Midterm Grade	0.1543	0.006498
Num Interactions (1/2)	Final Grade	0.2334	3.329e-05
Num Timeline Items (1/2)	Num Interactions (1/2)	0.7884	6.022e-67
Num Timeline Items (1/2)	Fraction NonZero Days (1/2)	0.627	2.882e-35
Num Timeline Items (1/2)	Mean Interval Length S (1/2)	-0.5059	1.514e-21
Num Timeline Items (1/2)	Var Intervals Length H (1/2)	-0.2478	1.009e-05
Num Timeline Items (1/2)	Skewness Intervals H (1/2)	0.2536	6.176e-06
Num Timeline Items (1/2)	Kurtosis Intervals H (1/2)	0.2231	7.425e-05
Num Timeline Items (1/2)	Burstiness (1/2)	0.08651	0.1286
Num Timeline Items (1/2)	Fraction of Midterm time	0.3508	2.092e-10
Num Timeline Items (1/2)	Assign Hand In Avg (1/2)	0.215	0.0001363
Num Timeline Items (1/2)	Midterm Grade	0.2549	5.507e-06
Num Timeline Items (1/2)	Final Grade	0.2966	1.034e-07
Fraction NonZero Days (1/2)	Num Interactions (1/2)	0.5983	1.722e-31
Fraction NonZero Days (1/2)	Num Timeline Items (1/2)	0.627	2.882e-35
Fraction NonZero Days (1/2)	Mean Interval Length S (1/2)	-0.4811	2.315e-19
Fraction NonZero Days (1/2)	Var Intervals Length H (1/2)	-0.3245	4.935e-09
Fraction NonZero Days (1/2)	Skewness Intervals H (1/2)	-0.1268	0.02556
Fraction NonZero Days (1/2)	Kurtosis Intervals H (1/2)	-0.1061	0.06213
Fraction NonZero Days (1/2)	Burstiness (1/2)	0.09032	0.1125
Fraction NonZero Days (1/2)	Fraction of Midterm time	0.4454	1.625e-16

Fraction NonZero Days (1/2)	Assign Hand In Avg (1/2)	0.1361	0.01647
Fraction NonZero Days (1/2)	Midterm Grade	0.2847	3.424e-07
Fraction NonZero Days (1/2)	Final Grade	0.4318	1.647e-15
Mean Interval Length S (1/2)	Num Interactions (1/2)	-0.4054	1.084e-13
Mean Interval Length S (1/2)	Num Timeline Items (1/2)	-0.5059	1.514e-21
Mean Interval Length S (1/2)	Fraction NonZero Days (1/2)	-0.4811	2.315e-19
Mean Interval Length S (1/2)	Var Intervals Length H (1/2)	0.8876	1.054e-105
Mean Interval Length S (1/2)	Skewness Intervals H (1/2)	-0.1797	0.001492
Mean Interval Length S (1/2)	Kurtosis Intervals H (1/2)	-0.1415	0.01264
Mean Interval Length S (1/2)	Burstiness (1/2)	0.1079	0.05766
Mean Interval Length S (1/2)	Fraction of Midterm time	-0.375	8.665e-12
Mean Interval Length S (1/2)	Assign Hand In Avg (1/2)	-0.1385	0.01466
Mean Interval Length S (1/2)	Midterm Grade	-0.2559	5.021e-06
Mean Interval Length S (1/2)	Final Grade	-0.3118	2.041e-08
Var Intervals Length H (1/2)	Num Interactions (1/2)	-0.2054	0.000272
Var Intervals Length H (1/2)	Num Timeline Items (1/2)	-0.2478	1.009e-05
Var Intervals Length H (1/2)	Fraction NonZero Days (1/2)	-0.3245	4.935e-09
Var Intervals Length H (1/2)	Mean Interval Length S (1/2)	0.8876	1.054e-105
Var Intervals Length H (1/2)	Skewness Intervals H (1/2)	-0.04873	0.3925
Var Intervals Length H (1/2)	Kurtosis Intervals H (1/2)	-0.02912	0.6095
Var Intervals Length H (1/2)	Burstiness (1/2)	0.04795	0.4002
Var Intervals Length H (1/2)	Fraction of Midterm time	-0.2871	2.693e-07
Var Intervals Length H (1/2)	Assign Hand In Avg (1/2)	-0.09774	0.08577
Var Intervals Length H (1/2)	Midterm Grade	-0.1977	0.000463
Var Intervals Length H (1/2)	Final Grade	-0.2356	2.792e-05
Skewness Intervals H (1/2)	Num Interactions (1/2)	0.1768	0.001775
Skewness Intervals H (1/2)	Num Timeline Items (1/2)	0.2536	6.176e-06
Skewness Intervals H (1/2)	Fraction NonZero Days (1/2)	-0.1268	0.02556
Skewness Intervals H (1/2)	Mean Interval Length S (1/2)	-0.1797	0.001492
Skewness Intervals H (1/2)	Var Intervals Length H (1/2)	-0.04873	0.3925
Skewness Intervals H (1/2)	Kurtosis Intervals H (1/2)	0.9733	5.343e-199
Skewness Intervals H (1/2)	Burstiness (1/2)	0.1997	0.0004032
Skewness Intervals H (1/2)	Fraction of Midterm time	0.1557	0.006001

Skewness Intervals H (1/2)	Assign Hand In Avg (1/2)	0.08009	0.1595
Skewness Intervals H (1/2)	Midterm Grade	0.119	0.03621
Skewness Intervals H (1/2)	Final Grade	0.1801	0.001452
Kurtosis Intervals H (1/2)	Num Interactions (1/2)	0.1588	0.005059
Kurtosis Intervals H (1/2)	Num Timeline Items (1/2)	0.2231	7.425e-05
Kurtosis Intervals H (1/2)	Fraction NonZero Days (1/2)	-0.1061	0.06213
Kurtosis Intervals H (1/2)	Mean Interval Length S (1/2)	-0.1415	0.01264
Kurtosis Intervals H (1/2)	Var Intervals Length H (1/2)	-0.02912	0.6095
Kurtosis Intervals H (1/2)	Skewness Intervals H (1/2)	0.9733	5.343e-199
Kurtosis Intervals H (1/2)	Burstiness (1/2)	0.1336	0.01861
Kurtosis Intervals H (1/2)	Fraction of Midterm time	0.1382	0.01491
Kurtosis Intervals H (1/2)	Assign Hand In Avg (1/2)	0.06511	0.253
Kurtosis Intervals H (1/2)	Midterm Grade	0.1232	0.03014
Kurtosis Intervals H (1/2)	Final Grade	0.1932	0.0006245
Burstiness (1/2)	Num Interactions (1/2)	0.05553	0.3298
Burstiness (1/2)	Num Timeline Items (1/2)	0.08651	0.1286
Burstiness (1/2)	Fraction NonZero Days (1/2)	0.09032	0.1125
Burstiness (1/2)	Mean Interval Length S (1/2)	0.1079	0.05766
Burstiness (1/2)	Var Intervals Length H (1/2)	0.04795	0.4002
Burstiness (1/2)	Skewness Intervals H (1/2)	0.1997	0.0004032
Burstiness (1/2)	Kurtosis Intervals H (1/2)	0.1336	0.01861
Burstiness (1/2)	Fraction of Midterm time	0.1792	0.001539
Burstiness (1/2)	Assign Hand In Avg (1/2)	0.05356	0.3473
Burstiness (1/2)	Midterm Grade	0.1158	0.04154
Burstiness (1/2)	Final Grade	0.1215	0.03247
Fraction of Midterm time	Num Interactions (1/2)	0.2343	3.09e-05
Fraction of Midterm time	Num Timeline Items (1/2)	0.3508	2.092e-10
Fraction of Midterm time	Fraction NonZero Days (1/2)	0.4454	1.625e-16
Fraction of Midterm time	Mean Interval Length S (1/2)	-0.375	8.665e-12
Fraction of Midterm time	Var Intervals Length H (1/2)	-0.2871	2.693e-07
Fraction of Midterm time	Skewness Intervals H (1/2)	0.1557	0.006001
Fraction of Midterm time	Kurtosis Intervals H (1/2)	0.1382	0.01491
Fraction of Midterm time	Burstiness (1/2)	0.1792	0.001539

Fraction of Midterm time	Assign Hand In Avg (1/2)	-0.02692	0.6368
Fraction of Midterm time	Midterm Grade	0.5811	2.168e-29
Fraction of Midterm time	Final Grade	0.5539	2.544e-26
Assign Hand In Avg (1/2)	Num Interactions (1/2)	0.1708	0.002553
Assign Hand In Avg (1/2)	Num Timeline Items (1/2)	0.215	0.0001363
Assign Hand In Avg (1/2)	Fraction NonZero Days (1/2)	0.1361	0.01647
Assign Hand In Avg (1/2)	Mean Interval Length S (1/2)	-0.1385	0.01466
Assign Hand In Avg (1/2)	Var Intervals Length H (1/2)	-0.09774	0.08577
Assign Hand In Avg (1/2)	Skewness Intervals H (1/2)	0.08009	0.1595
Assign Hand In Avg (1/2)	Kurtosis Intervals H (1/2)	0.06511	0.253
Assign Hand In Avg (1/2)	Burstiness (1/2)	0.05356	0.3473
Assign Hand In Avg (1/2)	Fraction of Midterm time	-0.02692	0.6368
Assign Hand In Avg (1/2)	Midterm Grade	0.08797	0.1222
Assign Hand In Avg (1/2)	Final Grade	0.118	0.0378
Midterm Grade	Num Interactions (1/2)	0.1543	0.006498
Midterm Grade	Num Timeline Items (1/2)	0.2549	5.507e-06
Midterm Grade	Fraction NonZero Days (1/2)	0.2847	3.424e-07
Midterm Grade	Mean Interval Length S (1/2)	-0.2559	5.021e-06
Midterm Grade	Var Intervals Length H (1/2)	-0.1977	0.000463
Midterm Grade	Skewness Intervals H (1/2)	0.119	0.03621
Midterm Grade	Kurtosis Intervals H (1/2)	0.1232	0.03014
Midterm Grade	Burstiness (1/2)	0.1158	0.04154
Midterm Grade	Fraction of Midterm time	0.5811	2.168e-29
Midterm Grade	Assign Hand In Avg (1/2)	0.08797	0.1222
Midterm Grade	Final Grade	0.7792	1.942e-64
Final Grade	Num Interactions (1/2)	0.2334	3.329e-05
Final Grade	Num Timeline Items (1/2)	0.2966	1.034e-07
Final Grade	Fraction NonZero Days (1/2)	0.4318	1.647e-15
Final Grade	Mean Interval Length S (1/2)	-0.3118	2.041e-08
Final Grade	Var Intervals Length H (1/2)	-0.2356	2.792e-05
Final Grade	Skewness Intervals H (1/2)	0.1801	0.001452
Final Grade	Kurtosis Intervals H (1/2)	0.1932	0.0006245
Final Grade	Burstiness (1/2)	0.1215	0.03247

Final Grade	Fraction of Midterm time	0.5539	2.544e-26
Final Grade	Assign Hand In Avg (1/2)	0.118	0.0378
Final Grade	Midterm Grade	0.7792	1.942e-64

C.4 Correlation and p values for features at the VW deadline

Feature 1	Feature 2	Correlation	p value
Num Interactions (VW)	Num Timeline Items (VW)	0.779	2.074e-64
Num Interactions (VW)	Fraction NonZero Days (VW)	0.6019	6.165e-32
Num Interactions (VW)	Mean Interval Length S (VW)	-0.4215	8.782e-15
Num Interactions (VW)	Var Intervals Length H (VW)	-0.217	0.0001177
Num Interactions (VW)	Skewness Intervals H (VW)	0.2166	0.0001212
Num Interactions (VW)	Kurtosis Intervals H (VW)	0.1986	0.0004354
Num Interactions (VW)	Burstiness (VW)	0.06082	0.2857
Num Interactions (VW)	Fraction of Midterm time	0.2654	2.155e-06
Num Interactions (VW)	Assign Hand In Avg (VW)	0.1462	0.009973
Num Interactions (VW)	Midterm Grade	0.1763	0.001831
Num Interactions (VW)	Final Grade	0.2846	3.445e-07
Num Timeline Items (VW)	Num Interactions (VW)	0.779	2.074e-64
Num Timeline Items (VW)	Fraction NonZero Days (VW)	0.6419	2.161e-37
Num Timeline Items (VW)	Mean Interval Length S (VW)	-0.5264	1.712e-23
Num Timeline Items (VW)	Var Intervals Length H (VW)	-0.2614	3.096e-06
Num Timeline Items (VW)	Skewness Intervals H (VW)	0.3278	3.366e-09
Num Timeline Items (VW)	Kurtosis Intervals H (VW)	0.3013	6.323e-08
Num Timeline Items (VW)	Burstiness (VW)	0.09546	0.0934
Num Timeline Items (VW)	Fraction of Midterm time	0.3873	1.567e-12
Num Timeline Items (VW)	Assign Hand In Avg (VW)	0.2085	0.0002184
Num Timeline Items (VW)	Midterm Grade	0.2912	1.784e-07
Num Timeline Items (VW)	Final Grade	0.3644	3.611e-11
Fraction NonZero Days (VW)	Num Interactions (VW)	0.6019	6.165e-32
Fraction NonZero Days (VW)	Num Timeline Items (VW)	0.6419	2.161e-37
Fraction NonZero Days (VW)	Mean Interval Length S (VW)	-0.5007	4.431e-21
Fraction NonZero Days (VW)	Var Intervals Length H (VW)	-0.3352	1.414e-09
Fraction NonZero Days (VW)	Skewness Intervals H (VW)	-0.06917	0.2246
Fraction NonZero Days (VW)	Kurtosis Intervals H (VW)	-0.0536	0.3469
Fraction NonZero Days (VW)	Burstiness (VW)	0.09494	0.0952
Fraction NonZero Days (VW)	Fraction of Midterm time	0.4836	1.407e-19

Fraction NonZero Days (VW)	Assign Hand In Avg (VW)	0.0867	0.1277
Fraction NonZero Days (VW)	Midterm Grade	0.3216	6.867e-09
Fraction NonZero Days (VW)	Final Grade	0.5031	2.709e-21
Mean Interval Length S (VW)	Num Interactions (VW)	-0.4215	8.782e-15
Mean Interval Length S (VW)	Num Timeline Items (VW)	-0.5264	1.712e-23
Mean Interval Length S (VW)	Fraction NonZero Days (VW)	-0.5007	4.431e-21
Mean Interval Length S (VW)	Var Intervals Length H (VW)	0.883	3.298e-103
Mean Interval Length S (VW)	Skewness Intervals H (VW)	-0.1862	0.0009868
Mean Interval Length S (VW)	Kurtosis Intervals H (VW)	-0.1458	0.01016
Mean Interval Length S (VW)	Burstiness (VW)	0.1089	0.05551
Mean Interval Length S (VW)	Fraction of Midterm time	-0.3785	5.396e-12
Mean Interval Length S (VW)	Assign Hand In Avg (VW)	-0.1378	0.01521
Mean Interval Length S (VW)	Midterm Grade	-0.2562	4.881e-06
Mean Interval Length S (VW)	Final Grade	-0.3296	2.74e-09
Var Intervals Length H (VW)	Num Interactions (VW)	-0.217	0.0001177
Var Intervals Length H (VW)	Num Timeline Items (VW)	-0.2614	3.096e-06
Var Intervals Length H (VW)	Fraction NonZero Days (VW)	-0.3352	1.414e-09
Var Intervals Length H (VW)	Mean Interval Length S (VW)	0.883	3.298e-103
Var Intervals Length H (VW)	Skewness Intervals H (VW)	-0.0507	0.3737
Var Intervals Length H (VW)	Kurtosis Intervals H (VW)	-0.03214	0.5729
Var Intervals Length H (VW)	Burstiness (VW)	0.04864	0.3935
Var Intervals Length H (VW)	Fraction of Midterm time	-0.2929	1.502e-07
Var Intervals Length H (VW)	Assign Hand In Avg (VW)	-0.09918	0.08124
Var Intervals Length H (VW)	Midterm Grade	-0.202	0.0003451
Var Intervals Length H (VW)	Final Grade	-0.2526	6.722e-06
Skewness Intervals H (VW)	Num Interactions (VW)	0.2166	0.0001212
Skewness Intervals H (VW)	Num Timeline Items (VW)	0.3278	3.366e-09
Skewness Intervals H (VW)	Fraction NonZero Days (VW)	-0.06917	0.2246
Skewness Intervals H (VW)	Mean Interval Length S (VW)	-0.1862	0.0009868
Skewness Intervals H (VW)	Var Intervals Length H (VW)	-0.0507	0.3737
Skewness Intervals H (VW)	Kurtosis Intervals H (VW)	0.9695	3.711e-190
Skewness Intervals H (VW)	Burstiness (VW)	0.1863	0.0009816
Skewness Intervals H (VW)	Fraction of Midterm time	0.1567	0.005696

Skewness Intervals H (VW)	Assign Hand In Avg (VW)	0.07985	0.1608
Skewness Intervals H (VW)	Midterm Grade	0.09365	0.09979
Skewness Intervals H (VW)	Final Grade	0.1331	0.01906
Kurtosis Intervals H (VW)	Num Interactions (VW)	0.1986	0.0004354
Kurtosis Intervals H (VW)	Num Timeline Items (VW)	0.3013	6.323e-08
Kurtosis Intervals H (VW)	Fraction NonZero Days (VW)	-0.0536	0.3469
Kurtosis Intervals H (VW)	Mean Interval Length S (VW)	-0.1458	0.01016
Kurtosis Intervals H (VW)	Var Intervals Length H (VW)	-0.03214	0.5729
Kurtosis Intervals H (VW)	Skewness Intervals H (VW)	0.9695	3.711e-190
Kurtosis Intervals H (VW)	Burstiness (VW)	0.1194	0.03562
Kurtosis Intervals H (VW)	Fraction of Midterm time	0.1344	0.0179
Kurtosis Intervals H (VW)	Assign Hand In Avg (VW)	0.05087	0.372
Kurtosis Intervals H (VW)	Midterm Grade	0.08795	0.1223
Kurtosis Intervals H (VW)	Final Grade	0.1323	0.01981
Burstiness (VW)	Num Interactions (VW)	0.06082	0.2857
Burstiness (VW)	Num Timeline Items (VW)	0.09546	0.0934
Burstiness (VW)	Fraction NonZero Days (VW)	0.09494	0.0952
Burstiness (VW)	Mean Interval Length S (VW)	0.1089	0.05551
Burstiness (VW)	Var Intervals Length H (VW)	0.04864	0.3935
Burstiness (VW)	Skewness Intervals H (VW)	0.1863	0.0009816
Burstiness (VW)	Kurtosis Intervals H (VW)	0.1194	0.03562
Burstiness (VW)	Fraction of Midterm time	0.1802	0.001442
Burstiness (VW)	Assign Hand In Avg (VW)	0.05485	0.3358
Burstiness (VW)	Midterm Grade	0.115	0.04295
Burstiness (VW)	Final Grade	0.1163	0.04071
Fraction of Midterm time	Num Interactions (VW)	0.2654	2.155e-06
Fraction of Midterm time	Num Timeline Items (VW)	0.3873	1.567e-12
Fraction of Midterm time	Fraction NonZero Days (VW)	0.4836	1.407e-19
Fraction of Midterm time	Mean Interval Length S (VW)	-0.3785	5.396e-12
Fraction of Midterm time	Var Intervals Length H (VW)	-0.2929	1.502e-07
Fraction of Midterm time	Skewness Intervals H (VW)	0.1567	0.005696
Fraction of Midterm time	Kurtosis Intervals H (VW)	0.1344	0.0179
Fraction of Midterm time	Burstiness (VW)	0.1802	0.001442

Fraction of Midterm time	Assign Hand In Avg (VW)	-0.03564	0.5319
Fraction of Midterm time	Midterm Grade	0.5811	2.168e-29
Fraction of Midterm time	Final Grade	0.5539	2.544e-26
Assign Hand In Avg (VW)	Num Interactions (VW)	0.1462	0.009973
Assign Hand In Avg (VW)	Num Timeline Items (VW)	0.2085	0.0002184
Assign Hand In Avg (VW)	Fraction NonZero Days (VW)	0.0867	0.1277
Assign Hand In Avg (VW)	Mean Interval Length S (VW)	-0.1378	0.01521
Assign Hand In Avg (VW)	Var Intervals Length H (VW)	-0.09918	0.08124
Assign Hand In Avg (VW)	Skewness Intervals H (VW)	0.07985	0.1608
Assign Hand In Avg (VW)	Kurtosis Intervals H (VW)	0.05087	0.372
Assign Hand In Avg (VW)	Burstiness (VW)	0.05485	0.3358
Assign Hand In Avg (VW)	Fraction of Midterm time	-0.03564	0.5319
Assign Hand In Avg (VW)	Midterm Grade	0.07264	0.2021
Assign Hand In Avg (VW)	Final Grade	0.09821	0.08428
Midterm Grade	Num Interactions (VW)	0.1763	0.001831
Midterm Grade	Num Timeline Items (VW)	0.2912	1.784e-07
Midterm Grade	Fraction NonZero Days (VW)	0.3216	6.867e-09
Midterm Grade	Mean Interval Length S (VW)	-0.2562	4.881e-06
Midterm Grade	Var Intervals Length H (VW)	-0.202	0.0003451
Midterm Grade	Skewness Intervals H (VW)	0.09365	0.09979
Midterm Grade	Kurtosis Intervals H (VW)	0.08795	0.1223
Midterm Grade	Burstiness (VW)	0.115	0.04295
Midterm Grade	Fraction of Midterm time	0.5811	2.168e-29
Midterm Grade	Assign Hand In Avg (VW)	0.07264	0.2021
Midterm Grade	Final Grade	0.7792	1.942e-64
Final Grade	Num Interactions (VW)	0.2846	3.445e-07
Final Grade	Num Timeline Items (VW)	0.3644	3.611e-11
Final Grade	Fraction NonZero Days (VW)	0.5031	2.709e-21
Final Grade	Mean Interval Length S (VW)	-0.3296	2.74e-09
Final Grade	Var Intervals Length H (VW)	-0.2526	6.722e-06
Final Grade	Skewness Intervals H (VW)	0.1331	0.01906
Final Grade	Kurtosis Intervals H (VW)	0.1323	0.01981
Final Grade	Burstiness (VW)	0.1163	0.04071

Final Grade	Fraction of Midterm time	0.5539	2.544e-26
Final Grade	Assign Hand In Avg (VW)	0.09821	0.08428
Final Grade	Midterm Grade	0.7792	1.942e-64

C.5 Correlation and p values for features at the third quarter in the term

Feature 1	Feature 2	Correlation	p value
Num Interactions (3/4)	Num Timeline Items (3/4)	0.7796	1.444e-64
Num Interactions (3/4)	Fraction NonZero Days (3/4)	0.6066	1.526e-32
Num Interactions (3/4)	Mean Interval Length S (3/4)	-0.4028	1.613e-13
Num Interactions (3/4)	Var Intervals Length H (3/4)	-0.2107	0.0001866
Num Interactions (3/4)	Skewness Intervals H (3/4)	0.1719	0.002392
Num Interactions (3/4)	Kurtosis Intervals H (3/4)	0.1588	0.005069
Num Interactions (3/4)	Burstiness (3/4)	0.05122	0.3688
Num Interactions (3/4)	Fraction of Midterm time	0.2847	3.42e-07
Num Interactions (3/4)	Assign Hand In Avg (3/4)	0.09807	0.08473
Num Interactions (3/4)	Midterm Grade	0.2092	0.0002078
Num Interactions (3/4)	Final Grade	0.3386	9.448e-10
Num Timeline Items (3/4)	Num Interactions (3/4)	0.7796	1.444e-64
Num Timeline Items (3/4)	Fraction NonZero Days (3/4)	0.6742	1.917e-42
Num Timeline Items (3/4)	Mean Interval Length S (3/4)	-0.5181	1.089e-22
Num Timeline Items (3/4)	Var Intervals Length H (3/4)	-0.2621	2.889e-06
Num Timeline Items (3/4)	Skewness Intervals H (3/4)	0.3089	2.801e-08
Num Timeline Items (3/4)	Kurtosis Intervals H (3/4)	0.2835	3.839e-07
Num Timeline Items (3/4)	Burstiness (3/4)	0.08885	0.1185
Num Timeline Items (3/4)	Fraction of Midterm time	0.4141	2.839e-14
Num Timeline Items (3/4)	Assign Hand In Avg (3/4)	0.1648	0.003616
Num Timeline Items (3/4)	Midterm Grade	0.3192	8.976e-09
Num Timeline Items (3/4)	Final Grade	0.4311	1.857e-15
Fraction NonZero Days (3/4)	Num Interactions (3/4)	0.6066	1.526e-32
Fraction NonZero Days (3/4)	Num Timeline Items (3/4)	0.6742	1.917e-42
Fraction NonZero Days (3/4)	Mean Interval Length S (3/4)	-0.4942	1.707e-20
Fraction NonZero Days (3/4)	Var Intervals Length H (3/4)	-0.3299	2.633e-09
Fraction NonZero Days (3/4)	Skewness Intervals H (3/4)	-0.04554	0.4243
Fraction NonZero Days (3/4)	Kurtosis Intervals H (3/4)	-0.02276	0.6898
Fraction NonZero Days (3/4)	Burstiness (3/4)	0.08912	0.1174
Fraction NonZero Days (3/4)	Fraction of Midterm time	0.5076	1.04e-21

Fraction NonZero Days (3/4)	Assign Hand In Avg (3/4)	0.04506	0.4292
Fraction NonZero Days (3/4)	Midterm Grade	0.3534	1.503e-10
Fraction NonZero Days (3/4)	Final Grade	0.5743	1.338e-28
Mean Interval Length S (3/4)	Num Interactions (3/4)	-0.4028	1.613e-13
Mean Interval Length S (3/4)	Num Timeline Items (3/4)	-0.5181	1.089e-22
Mean Interval Length S (3/4)	Fraction NonZero Days (3/4)	-0.4942	1.707e-20
Mean Interval Length S (3/4)	Var Intervals Length H (3/4)	0.8799	1.429e-101
Mean Interval Length S (3/4)	Skewness Intervals H (3/4)	-0.1846	0.001094
Mean Interval Length S (3/4)	Kurtosis Intervals H (3/4)	-0.1438	0.01124
Mean Interval Length S (3/4)	Burstiness (3/4)	0.1102	0.0525
Mean Interval Length S (3/4)	Fraction of Midterm time	-0.3765	7.109e-12
Mean Interval Length S (3/4)	Assign Hand In Avg (3/4)	-0.1149	0.04327
Mean Interval Length S (3/4)	Midterm Grade	-0.255	5.452e-06
Mean Interval Length S (3/4)	Final Grade	-0.3371	1.126e-09
Var Intervals Length H (3/4)	Num Interactions (3/4)	-0.2107	0.0001866
Var Intervals Length H (3/4)	Num Timeline Items (3/4)	-0.2621	2.889e-06
Var Intervals Length H (3/4)	Fraction NonZero Days (3/4)	-0.3299	2.633e-09
Var Intervals Length H (3/4)	Mean Interval Length S (3/4)	0.8799	1.429e-101
Var Intervals Length H (3/4)	Skewness Intervals H (3/4)	-0.05581	0.3274
Var Intervals Length H (3/4)	Kurtosis Intervals H (3/4)	-0.03705	0.5158
Var Intervals Length H (3/4)	Burstiness (3/4)	0.0486	0.3938
Var Intervals Length H (3/4)	Fraction of Midterm time	-0.291	1.822e-07
Var Intervals Length H (3/4)	Assign Hand In Avg (3/4)	-0.08708	0.126
Var Intervals Length H (3/4)	Midterm Grade	-0.205	0.0002791
Var Intervals Length H (3/4)	Final Grade	-0.2569	4.589e-06
Skewness Intervals H (3/4)	Num Interactions (3/4)	0.1719	0.002392
Skewness Intervals H (3/4)	Num Timeline Items (3/4)	0.3089	2.801e-08
Skewness Intervals H (3/4)	Fraction NonZero Days (3/4)	-0.04554	0.4243
Skewness Intervals H (3/4)	Mean Interval Length S (3/4)	-0.1846	0.001094
Skewness Intervals H (3/4)	Var Intervals Length H (3/4)	-0.05581	0.3274
Skewness Intervals H (3/4)	Kurtosis Intervals H (3/4)	0.9689	6.842e-189
Skewness Intervals H (3/4)	Burstiness (3/4)	0.1867	0.0009572
Skewness Intervals H (3/4)	Fraction of Midterm time	0.1653	0.003505

Skewness Intervals H (3/4)	Assign Hand In Avg (3/4)	0.0999	0.07906
Skewness Intervals H (3/4)	Midterm Grade	0.08595	0.131
Skewness Intervals H (3/4)	Final Grade	0.1234	0.02978
Kurtosis Intervals H (3/4)	Num Interactions (3/4)	0.1588	0.005069
Kurtosis Intervals H (3/4)	Num Timeline Items (3/4)	0.2835	3.839e-07
Kurtosis Intervals H (3/4)	Fraction NonZero Days (3/4)	-0.02276	0.6898
Kurtosis Intervals H (3/4)	Mean Interval Length S (3/4)	-0.1438	0.01124
Kurtosis Intervals H (3/4)	Var Intervals Length H (3/4)	-0.03705	0.5158
Kurtosis Intervals H (3/4)	Skewness Intervals H (3/4)	0.9689	6.842e-189
Kurtosis Intervals H (3/4)	Burstiness (3/4)	0.116	0.04121
Kurtosis Intervals H (3/4)	Fraction of Midterm time	0.1421	0.01225
Kurtosis Intervals H (3/4)	Assign Hand In Avg (3/4)	0.07353	0.1967
Kurtosis Intervals H (3/4)	Midterm Grade	0.08008	0.1596
Kurtosis Intervals H (3/4)	Final Grade	0.1301	0.02198
Burstiness (3/4)	Num Interactions (3/4)	0.05122	0.3688
Burstiness (3/4)	Num Timeline Items (3/4)	0.08885	0.1185
Burstiness (3/4)	Fraction NonZero Days (3/4)	0.08912	0.1174
Burstiness (3/4)	Mean Interval Length S (3/4)	0.1102	0.0525
Burstiness (3/4)	Var Intervals Length H (3/4)	0.0486	0.3938
Burstiness (3/4)	Skewness Intervals H (3/4)	0.1867	0.0009572
Burstiness (3/4)	Kurtosis Intervals H (3/4)	0.116	0.04121
Burstiness (3/4)	Fraction of Midterm time	0.1787	0.001583
Burstiness (3/4)	Assign Hand In Avg (3/4)	0.05545	0.3305
Burstiness (3/4)	Midterm Grade	0.1132	0.04648
Burstiness (3/4)	Final Grade	0.1116	0.04961
Fraction of Midterm time	Num Interactions (3/4)	0.2847	3.42e-07
Fraction of Midterm time	Num Timeline Items (3/4)	0.4141	2.839e-14
Fraction of Midterm time	Fraction NonZero Days (3/4)	0.5076	1.04e-21
Fraction of Midterm time	Mean Interval Length S (3/4)	-0.3765	7.109e-12
Fraction of Midterm time	Var Intervals Length H (3/4)	-0.291	1.822e-07
Fraction of Midterm time	Skewness Intervals H (3/4)	0.1653	0.003505
Fraction of Midterm time	Kurtosis Intervals H (3/4)	0.1421	0.01225
Fraction of Midterm time	Burstiness (3/4)	0.1787	0.001583

Fraction of Midterm time	Assign Hand In Avg (3/4)	-0.05921	0.2987
Fraction of Midterm time	Midterm Grade	0.5811	2.168e-29
Fraction of Midterm time	Final Grade	0.5539	2.544e-26
Assign Hand In Avg (3/4)	Num Interactions (3/4)	0.09807	0.08473
Assign Hand In Avg (3/4)	Num Timeline Items (3/4)	0.1648	0.003616
Assign Hand In Avg (3/4)	Fraction NonZero Days (3/4)	0.04506	0.4292
Assign Hand In Avg (3/4)	Mean Interval Length S (3/4)	-0.1149	0.04327
Assign Hand In Avg (3/4)	Var Intervals Length H (3/4)	-0.08708	0.126
Assign Hand In Avg (3/4)	Skewness Intervals H (3/4)	0.0999	0.07906
Assign Hand In Avg (3/4)	Kurtosis Intervals H (3/4)	0.07353	0.1967
Assign Hand In Avg (3/4)	Burstiness (3/4)	0.05545	0.3305
Assign Hand In Avg (3/4)	Fraction of Midterm time	-0.05921	0.2987
Assign Hand In Avg (3/4)	Midterm Grade	0.0711	0.2119
Assign Hand In Avg (3/4)	Final Grade	0.08859	0.1196
Midterm Grade	Num Interactions (3/4)	0.2092	0.0002078
Midterm Grade	Num Timeline Items (3/4)	0.3192	8.976e-09
Midterm Grade	Fraction NonZero Days (3/4)	0.3534	1.503e-10
Midterm Grade	Mean Interval Length S (3/4)	-0.255	5.452e-06
Midterm Grade	Var Intervals Length H (3/4)	-0.205	0.0002791
Midterm Grade	Skewness Intervals H (3/4)	0.08595	0.131
Midterm Grade	Kurtosis Intervals H (3/4)	0.08008	0.1596
Midterm Grade	Burstiness (3/4)	0.1132	0.04648
Midterm Grade	Fraction of Midterm time	0.5811	2.168e-29
Midterm Grade	Assign Hand In Avg (3/4)	0.0711	0.2119
Midterm Grade	Final Grade	0.7792	1.942e-64
Final Grade	Num Interactions (3/4)	0.3386	9.448e-10
Final Grade	Num Timeline Items (3/4)	0.4311	1.857e-15
Final Grade	Fraction NonZero Days (3/4)	0.5743	1.338e-28
Final Grade	Mean Interval Length S (3/4)	-0.3371	1.126e-09
Final Grade	Var Intervals Length H (3/4)	-0.2569	4.589e-06
Final Grade	Skewness Intervals H (3/4)	0.1234	0.02978
Final Grade	Kurtosis Intervals H (3/4)	0.1301	0.02198
Final Grade	Burstiness (3/4)	0.1116	0.04961

Final Grade	Fraction of Midterm time	0.5539	2.544e-26
Final Grade	Assign Hand In Avg (3/4)	0.08859	0.1196
Final Grade	Midterm Grade	0.7792	1.942e-64

C.6 Correlation and p values for features at the final exam

Feature 1	Feature 2	Correlation	p value
Num Interactions (exam)	Num Timeline Items (exam)	0.7731	7.251e-63
Num Interactions (exam)	Fraction NonZero Days (exam)	0.6082	9.617e-33
Num Interactions (exam)	Mean Interval Length S (exam)	-0.4182	1.498e-14
Num Interactions (exam)	Var Intervals Length H (exam)	-0.2065	0.000252
Num Interactions (exam)	Skewness Intervals H (exam)	0.3892	1.184e-12
Num Interactions (exam)	Kurtosis Intervals H (exam)	0.3959	4.506e-13
Num Interactions (exam)	Burstiness (exam)	0.06791	0.2332
Num Interactions (exam)	Fraction of Midterm time	0.2839	3.723e-07
Num Interactions (exam)	Assign Hand In Avg (exam)	0.04919	0.3881
Num Interactions (exam)	Midterm Grade	0.218	0.0001093
Num Interactions (exam)	Final Grade	0.3534	1.499e-10
Num Timeline Items (exam)	Num Interactions (exam)	0.7731	7.251e-63
Num Timeline Items (exam)	Fraction NonZero Days (exam)	0.7204	7.162e-51
Num Timeline Items (exam)	Mean Interval Length S (exam)	-0.566	1.173e-27
Num Timeline Items (exam)	Var Intervals Length H (exam)	-0.2834	3.877e-07
Num Timeline Items (exam)	Skewness Intervals H (exam)	0.3908	9.352e-13
Num Timeline Items (exam)	Kurtosis Intervals H (exam)	0.3622	4.858e-11
Num Timeline Items (exam)	Burstiness (exam)	0.1016	0.07396
Num Timeline Items (exam)	Fraction of Midterm time	0.4414	3.287e-16
Num Timeline Items (exam)	Assign Hand In Avg (exam)	0.1229	0.03051
Num Timeline Items (exam)	Midterm Grade	0.3346	1.523e-09
Num Timeline Items (exam)	Final Grade	0.475	7.5e-19
Fraction NonZero Days (exam)	Num Interactions (exam)	0.6082	9.617e-33
Fraction NonZero Days (exam)	Num Timeline Items (exam)	0.7204	7.162e-51
Fraction NonZero Days (exam)	Mean Interval Length S (exam)	-0.5172	1.324e-22
Fraction NonZero Days (exam)	Var Intervals Length H (exam)	-0.3456	3.985e-10
Fraction NonZero Days (exam)	Skewness Intervals H (exam)	0.1122	0.04834
Fraction NonZero Days (exam)	Kurtosis Intervals H (exam)	0.1385	0.01465
Fraction NonZero Days (exam)	Burstiness (exam)	0.1066	0.0609
Fraction NonZero Days (exam)	Fraction of Midterm time	0.5217	4.897e-23
Fraction NonZero Days (exam)	Assign Hand In Avg (exam)	0.03597	0.5281

Fraction NonZero Days (exam)	Midterm Grade	0.3663	2.796e-11
Fraction NonZero Days (exam)	Final Grade	0.6021	5.768e-32
Mean Interval Length S (exam)	Num Interactions (exam)	-0.4182	1.498e-14
Mean Interval Length S (exam)	Num Timeline Items (exam)	-0.566	1.173e-27
Mean Interval Length S (exam)	Fraction NonZero Days (exam)	-0.5172	1.324e-22
Mean Interval Length S (exam)	Var Intervals Length H (exam)	0.838	5.314e-83
Mean Interval Length S (exam)	Skewness Intervals H (exam)	-0.1964	0.0005039
Mean Interval Length S (exam)	Kurtosis Intervals H (exam)	-0.1557	0.006005
Mean Interval Length S (exam)	Burstiness (exam)	0.1127	0.04737
Mean Interval Length S (exam)	Fraction of Midterm time	-0.4108	4.73e-14
Mean Interval Length S (exam)	Assign Hand In Avg (exam)	-0.1058	0.06276
Mean Interval Length S (exam)	Midterm Grade	-0.2863	2.913e-07
Mean Interval Length S (exam)	Final Grade	-0.3706	1.576e-11
Var Intervals Length H (exam)	Num Interactions (exam)	-0.2065	0.000252
Var Intervals Length H (exam)	Num Timeline Items (exam)	-0.2834	3.877e-07
Var Intervals Length H (exam)	Fraction NonZero Days (exam)	-0.3456	3.985e-10
Var Intervals Length H (exam)	Mean Interval Length S (exam)	0.838	5.314e-83
Var Intervals Length H (exam)	Skewness Intervals H (exam)	-0.04987	0.3816
Var Intervals Length H (exam)	Kurtosis Intervals H (exam)	-0.03738	0.5121
Var Intervals Length H (exam)	Burstiness (exam)	0.05006	0.3798
Var Intervals Length H (exam)	Fraction of Midterm time	-0.3097	2.568e-08
Var Intervals Length H (exam)	Assign Hand In Avg (exam)	-0.08684	0.1271
Var Intervals Length H (exam)	Midterm Grade	-0.2248	6.527e-05
Var Intervals Length H (exam)	Final Grade	-0.2777	6.791e-07
Skewness Intervals H (exam)	Num Interactions (exam)	0.3892	1.184e-12
Skewness Intervals H (exam)	Num Timeline Items (exam)	0.3908	9.352e-13
Skewness Intervals H (exam)	Fraction NonZero Days (exam)	0.1122	0.04834
Skewness Intervals H (exam)	Mean Interval Length S (exam)	-0.1964	0.0005039
Skewness Intervals H (exam)	Var Intervals Length H (exam)	-0.04987	0.3816
Skewness Intervals H (exam)	Kurtosis Intervals H (exam)	0.9723	1.977e-196
Skewness Intervals H (exam)	Burstiness (exam)	0.1757	0.0019
Skewness Intervals H (exam)	Fraction of Midterm time	0.1613	0.004416
Skewness Intervals H (exam)	Assign Hand In Avg (exam)	0.09616	0.091

Skewness Intervals H (exam)	Midterm Grade	0.04223	0.4588
Skewness Intervals H (exam)	Final Grade	0.1141	0.04473
Kurtosis Intervals H (exam)	Num Interactions (exam)	0.3959	4.506e-13
Kurtosis Intervals H (exam)	Num Timeline Items (exam)	0.3622	4.858e-11
Kurtosis Intervals H (exam)	Fraction NonZero Days (exam)	0.1385	0.01465
Kurtosis Intervals H (exam)	Mean Interval Length S (exam)	-0.1557	0.006005
Kurtosis Intervals H (exam)	Var Intervals Length H (exam)	-0.03738	0.5121
Kurtosis Intervals H (exam)	Skewness Intervals H (exam)	0.9723	1.977e-196
Kurtosis Intervals H (exam)	Burstiness (exam)	0.1089	0.05546
Kurtosis Intervals H (exam)	Fraction of Midterm time	0.1314	0.02062
Kurtosis Intervals H (exam)	Assign Hand In Avg (exam)	0.06892	0.2263
Kurtosis Intervals H (exam)	Midterm Grade	0.0281	0.6221
Kurtosis Intervals H (exam)	Final Grade	0.1062	0.06193
Burstiness (exam)	Num Interactions (exam)	0.06791	0.2332
Burstiness (exam)	Num Timeline Items (exam)	0.1016	0.07396
Burstiness (exam)	Fraction NonZero Days (exam)	0.1066	0.0609
Burstiness (exam)	Mean Interval Length S (exam)	0.1127	0.04737
Burstiness (exam)	Var Intervals Length H (exam)	0.05006	0.3798
Burstiness (exam)	Skewness Intervals H (exam)	0.1757	0.0019
Burstiness (exam)	Kurtosis Intervals H (exam)	0.1089	0.05546
Burstiness (exam)	Fraction of Midterm time	0.1818	0.001308
Burstiness (exam)	Assign Hand In Avg (exam)	0.05591	0.3265
Burstiness (exam)	Midterm Grade	0.1104	0.05205
Burstiness (exam)	Final Grade	0.1131	0.04663
Fraction of Midterm time	Num Interactions (exam)	0.2839	3.723e-07
Fraction of Midterm time	Num Timeline Items (exam)	0.4414	3.287e-16
Fraction of Midterm time	Fraction NonZero Days (exam)	0.5217	4.897e-23
Fraction of Midterm time	Mean Interval Length S (exam)	-0.4108	4.73e-14
Fraction of Midterm time	Var Intervals Length H (exam)	-0.3097	2.568e-08
Fraction of Midterm time	Skewness Intervals H (exam)	0.1613	0.004416
Fraction of Midterm time	Kurtosis Intervals H (exam)	0.1314	0.02062
Fraction of Midterm time	Burstiness (exam)	0.1818	0.001308
Fraction of Midterm time	Assign Hand In Avg (exam)	-0.05681	0.3188

Fraction of Midterm time	Midterm Grade	0.5811	2.168e-29
Fraction of Midterm time	Final Grade	0.5539	2.544e-26
Assign Hand In Avg (exam)	Num Interactions (exam)	0.04919	0.3881
Assign Hand In Avg (exam)	Num Timeline Items (exam)	0.1229	0.03051
Assign Hand In Avg (exam)	Fraction NonZero Days (exam)	0.03597	0.5281
Assign Hand In Avg (exam)	Mean Interval Length S (exam)	-0.1058	0.06276
Assign Hand In Avg (exam)	Var Intervals Length H (exam)	-0.08684	0.1271
Assign Hand In Avg (exam)	Skewness Intervals H (exam)	0.09616	0.091
Assign Hand In Avg (exam)	Kurtosis Intervals H (exam)	0.06892	0.2263
Assign Hand In Avg (exam)	Burstiness (exam)	0.05591	0.3265
Assign Hand In Avg (exam)	Fraction of Midterm time	-0.05681	0.3188
Assign Hand In Avg (exam)	Midterm Grade	0.08531	0.134
Assign Hand In Avg (exam)	Final Grade	0.09878	0.0825
Midterm Grade	Num Interactions (exam)	0.218	0.0001093
Midterm Grade	Num Timeline Items (exam)	0.3346	1.523e-09
Midterm Grade	Fraction NonZero Days (exam)	0.3663	2.796e-11
Midterm Grade	Mean Interval Length S (exam)	-0.2863	2.913e-07
Midterm Grade	Var Intervals Length H (exam)	-0.2248	6.527e-05
Midterm Grade	Skewness Intervals H (exam)	0.04223	0.4588
Midterm Grade	Kurtosis Intervals H (exam)	0.0281	0.6221
Midterm Grade	Burstiness (exam)	0.1104	0.05205
Midterm Grade	Fraction of Midterm time	0.5811	2.168e-29
Midterm Grade	Assign Hand In Avg (exam)	0.08531	0.134
Midterm Grade	Final Grade	0.7792	1.942e-64
Final Grade	Num Interactions (exam)	0.3534	1.499e-10
Final Grade	Num Timeline Items (exam)	0.475	7.5e-19
Final Grade	Fraction NonZero Days (exam)	0.6021	5.768e-32
Final Grade	Mean Interval Length S (exam)	-0.3706	1.576e-11
Final Grade	Var Intervals Length H (exam)	-0.2777	6.791e-07
Final Grade	Skewness Intervals H (exam)	0.1141	0.04473
Final Grade	Kurtosis Intervals H (exam)	0.1062	0.06193
Final Grade	Burstiness (exam)	0.1131	0.04663
Final Grade	Fraction of Midterm time	0.5539	2.544e-26

Final Grade	Assign Hand In Avg (exam)	0.09878	0.0825
Final Grade	Midterm Grade	0.7792	1.942e-64

C.7 Correlation and p values for features calculated over the full term

Feature 1	Feature 2	Correlation	p value
Num Interactions (all)	Num Timeline Items (all)	0.7763	1.124e-63
Num Interactions (all)	Fraction NonZero Days (all)	0.564	1.981e-27
Num Interactions (all)	Mean Interval Length S (all)	-0.4095	5.819e-14
Num Interactions (all)	Var Intervals Length H (all)	-0.2026	0.0003299
Num Interactions (all)	Skewness Intervals H (all)	0.462	8.539e-18
Num Interactions (all)	Kurtosis Intervals H (all)	0.4792	3.326e-19
Num Interactions (all)	Burstiness (all)	0.07541	0.1854
Num Interactions (all)	Fraction of Midterm time	0.2864	2.899e-07
Num Interactions (all)	Assign Hand In Avg (all)	0.05062	0.3744
Num Interactions (all)	Midterm Grade	0.2104	0.0001898
Num Interactions (all)	Final Grade	0.3474	3.18e-10
Num Timeline Items (all)	Num Interactions (all)	0.7763	1.124e-63
Num Timeline Items (all)	Fraction NonZero Days (all)	0.6942	6.898e-46
Num Timeline Items (all)	Mean Interval Length S (all)	-0.5596	6.018e-27
Num Timeline Items (all)	Var Intervals Length H (all)	-0.2809	4.954e-07
Num Timeline Items (all)	Skewness Intervals H (all)	0.4487	9.139e-17
Num Timeline Items (all)	Kurtosis Intervals H (all)	0.4305	2.022e-15
Num Timeline Items (all)	Burstiness (all)	0.108	0.05758
Num Timeline Items (all)	Fraction of Midterm time	0.4441	2.048e-16
Num Timeline Items (all)	Assign Hand In Avg (all)	0.1214	0.03262
Num Timeline Items (all)	Midterm Grade	0.3253	4.518e-09
Num Timeline Items (all)	Final Grade	0.4685	2.547e-18
Fraction NonZero Days (all)	Num Interactions (all)	0.564	1.981e-27
Fraction NonZero Days (all)	Num Timeline Items (all)	0.6942	6.898e-46
Fraction NonZero Days (all)	Mean Interval Length S (all)	-0.5079	9.945e-22
Fraction NonZero Days (all)	Var Intervals Length H (all)	-0.3335	1.733e-09
Fraction NonZero Days (all)	Skewness Intervals H (all)	0.1233	0.02995
Fraction NonZero Days (all)	Kurtosis Intervals H (all)	0.1515	0.007555
Fraction NonZero Days (all)	Burstiness (all)	0.1027	0.07104
Fraction NonZero Days (all)	Fraction of Midterm time	0.5097	6.712e-22

Fraction NonZero Days (all)	Assign Hand In Avg (all)	0.003906	0.9454
Fraction NonZero Days (all)	Midterm Grade	0.3533	1.521e-10
Fraction NonZero Days (all)	Final Grade	0.5836	1.089e-29
Mean Interval Length S (all)	Num Interactions (all)	-0.4095	5.819e-14
Mean Interval Length S (all)	Num Timeline Items (all)	-0.5596	6.018e-27
Mean Interval Length S (all)	Fraction NonZero Days (all)	-0.5079	9.945e-22
Mean Interval Length S (all)	Var Intervals Length H (all)	0.8397	1.123e-83
Mean Interval Length S (all)	Skewness Intervals H (all)	-0.2004	0.0003858
Mean Interval Length S (all)	Kurtosis Intervals H (all)	-0.1597	0.004833
Mean Interval Length S (all)	Burstiness (all)	0.111	0.0508
Mean Interval Length S (all)	Fraction of Midterm time	-0.4147	2.572e-14
Mean Interval Length S (all)	Assign Hand In Avg (all)	-0.1026	0.07125
Mean Interval Length S (all)	Midterm Grade	-0.2837	3.782e-07
Mean Interval Length S (all)	Final Grade	-0.3674	2.424e-11
Var Intervals Length H (all)	Num Interactions (all)	-0.2026	0.0003299
Var Intervals Length H (all)	Num Timeline Items (all)	-0.2809	4.954e-07
Var Intervals Length H (all)	Fraction NonZero Days (all)	-0.3335	1.733e-09
Var Intervals Length H (all)	Mean Interval Length S (all)	0.8397	1.123e-83
Var Intervals Length H (all)	Skewness Intervals H (all)	-0.05083	0.3724
Var Intervals Length H (all)	Kurtosis Intervals H (all)	-0.0384	0.5005
Var Intervals Length H (all)	Burstiness (all)	0.04956	0.3845
Var Intervals Length H (all)	Fraction of Midterm time	-0.3098	2.557e-08
Var Intervals Length H (all)	Assign Hand In Avg (all)	-0.08618	0.13
Var Intervals Length H (all)	Midterm Grade	-0.224	6.909e-05
Var Intervals Length H (all)	Final Grade	-0.2763	7.719e-07
Skewness Intervals H (all)	Num Interactions (all)	0.462	8.539e-18
Skewness Intervals H (all)	Num Timeline Items (all)	0.4487	9.139e-17
Skewness Intervals H (all)	Fraction NonZero Days (all)	0.1233	0.02995
Skewness Intervals H (all)	Mean Interval Length S (all)	-0.2004	0.0003858
Skewness Intervals H (all)	Var Intervals Length H (all)	-0.05083	0.3724
Skewness Intervals H (all)	Kurtosis Intervals H (all)	0.968	6.697e-187
Skewness Intervals H (all)	Burstiness (all)	0.1707	0.002559
Skewness Intervals H (all)	Fraction of Midterm time	0.1617	0.004315

Skewness Intervals H (all)	Assign Hand In Avg (all)	0.09068	0.1111
Skewness Intervals H (all)	Midterm Grade	0.02909	0.6099
Skewness Intervals H (all)	Final Grade	0.1069	0.06012
Kurtosis Intervals H (all)	Num Interactions (all)	0.4792	3.326e-19
Kurtosis Intervals H (all)	Num Timeline Items (all)	0.4305	2.022e-15
Kurtosis Intervals H (all)	Fraction NonZero Days (all)	0.1515	0.007555
Kurtosis Intervals H (all)	Mean Interval Length S (all)	-0.1597	0.004833
Kurtosis Intervals H (all)	Var Intervals Length H (all)	-0.0384	0.5005
Kurtosis Intervals H (all)	Skewness Intervals H (all)	0.968	6.697e-187
Kurtosis Intervals H (all)	Burstiness (all)	0.103	0.07003
Kurtosis Intervals H (all)	Fraction of Midterm time	0.1279	0.02434
Kurtosis Intervals H (all)	Assign Hand In Avg (all)	0.06379	0.2628
Kurtosis Intervals H (all)	Midterm Grade	0.01343	0.8138
Kurtosis Intervals H (all)	Final Grade	0.09643	0.09008
Burstiness (all)	Num Interactions (all)	0.07541	0.1854
Burstiness (all)	Num Timeline Items (all)	0.108	0.05758
Burstiness (all)	Fraction NonZero Days (all)	0.1027	0.07104
Burstiness (all)	Mean Interval Length S (all)	0.111	0.0508
Burstiness (all)	Var Intervals Length H (all)	0.04956	0.3845
Burstiness (all)	Skewness Intervals H (all)	0.1707	0.002559
Burstiness (all)	Kurtosis Intervals H (all)	0.103	0.07003
Burstiness (all)	Fraction of Midterm time	0.1827	0.001234
Burstiness (all)	Assign Hand In Avg (all)	0.05577	0.3277
Burstiness (all)	Midterm Grade	0.1097	0.05369
Burstiness (all)	Final Grade	0.1131	0.04657
Fraction of Midterm time	Num Interactions (all)	0.2864	2.899e-07
Fraction of Midterm time	Num Timeline Items (all)	0.4441	2.048e-16
Fraction of Midterm time	Fraction NonZero Days (all)	0.5097	6.712e-22
Fraction of Midterm time	Mean Interval Length S (all)	-0.4147	2.572e-14
Fraction of Midterm time	Var Intervals Length H (all)	-0.3098	2.557e-08
Fraction of Midterm time	Skewness Intervals H (all)	0.1617	0.004315
Fraction of Midterm time	Kurtosis Intervals H (all)	0.1279	0.02434
Fraction of Midterm time	Burstiness (all)	0.1827	0.001234

Fraction of Midterm time	Assign Hand In Avg (all)	-0.05681	0.3188
Fraction of Midterm time	Midterm Grade	0.5811	2.168e-29
Fraction of Midterm time	Final Grade	0.5539	2.544e-26
Assign Hand In Avg (all)	Num Interactions (all)	0.05062	0.3744
Assign Hand In Avg (all)	Num Timeline Items (all)	0.1214	0.03262
Assign Hand In Avg (all)	Fraction NonZero Days (all)	0.003906	0.9454
Assign Hand In Avg (all)	Mean Interval Length S (all)	-0.1026	0.07125
Assign Hand In Avg (all)	Var Intervals Length H (all)	-0.08618	0.13
Assign Hand In Avg (all)	Skewness Intervals H (all)	0.09068	0.1111
Assign Hand In Avg (all)	Kurtosis Intervals H (all)	0.06379	0.2628
Assign Hand In Avg (all)	Burstiness (all)	0.05577	0.3277
Assign Hand In Avg (all)	Fraction of Midterm time	-0.05681	0.3188
Assign Hand In Avg (all)	Midterm Grade	0.08531	0.134
Assign Hand In Avg (all)	Final Grade	0.09878	0.0825
Midterm Grade	Num Interactions (all)	0.2104	0.0001898
Midterm Grade	Num Timeline Items (all)	0.3253	4.518e-09
Midterm Grade	Fraction NonZero Days (all)	0.3533	1.521e-10
Midterm Grade	Mean Interval Length S (all)	-0.2837	3.782e-07
Midterm Grade	Var Intervals Length H (all)	-0.224	6.909e-05
Midterm Grade	Skewness Intervals H (all)	0.02909	0.6099
Midterm Grade	Kurtosis Intervals H (all)	0.01343	0.8138
Midterm Grade	Burstiness (all)	0.1097	0.05369
Midterm Grade	Fraction of Midterm time	0.5811	2.168e-29
Midterm Grade	Assign Hand In Avg (all)	0.08531	0.134
Midterm Grade	Final Grade	0.7792	1.942e-64
Final Grade	Num Interactions (all)	0.3474	3.18e-10
Final Grade	Num Timeline Items (all)	0.4685	2.547e-18
Final Grade	Fraction NonZero Days (all)	0.5836	1.089e-29
Final Grade	Mean Interval Length S (all)	-0.3674	2.424e-11
Final Grade	Var Intervals Length H (all)	-0.2763	7.719e-07
Final Grade	Skewness Intervals H (all)	0.1069	0.06012
Final Grade	Kurtosis Intervals H (all)	0.09643	0.09008
Final Grade	Burstiness (all)	0.1131	0.04657

Final Grade	Fraction of Midterm time	0.5539	2.544e-26
Final Grade	Assign Hand In Avg (all)	0.09878	0.0825
Final Grade	Midterm Grade	0.7792	1.942e-64

Appendix D

Early Prediction with Ternary Classifiers

D.1 CNN ternary classifiers: Letter Grades (VW)

Tables D.1 to D.4 present the average results of early prediction with a CNN ternary classifier using classification categories based on the letter grades.

Table D.1: Average values for CNN ternary classifier based on letter grades (up to VW date)

Type of Data	**Train (%)**	**Test (%)**	**Epochs**
Intervals	46.73	43.60	160
Timestamp	54.64	50.87	259
Multivariable	71.95	57.19	289

Table D.2: Average error matrix for CNN ternary classifier based on letter grades with Intervals data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	3	1	13
Actual Warning	3	1	15
Actual Pass	3	2	21

Table D.3: Average error matrix for CNN ternary classifier based on letter grades with Timestamp data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	7	3	8
Actual Warning	3	4	11
Actual Pass	2	6	19

Table D.4: Average error matrix for CNN ternary classifier based on letter grades with Multivariable data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	10	2	6
Actual Warning	3	4	11
Actual Pass	1	4	21

Tables D.5 to D.8 present the results of the best early prediction CNN ternary classifier model out of 100 trials, using classification categories based on letter grades.

Table D.5: Best values for CNN ternary classifier based on letter grades (up to VW date)

Type of Data	**Train (%)**	**Test (%)**	**Epochs**
Intervals	40.98	58.73	75
Timestamp	49.25	65.08	75
Multivariable	69.63	69.84	192

Table D.6: Best error matrix for CNN ternary classifier based on letter grades with Intervals data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	0	0	11
Actual Warning	0	0	15
Actual Pass	0	0	37

Table D.7: Best error matrix for CNN ternary classifier based on letter grades with Timestamp data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	5	5	8
Actual Warning	0	14	3
Actual Pass	0	6	22

Table D.8: Best error matrix for CNN ternary classifier based on letter grades with Multivariable data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	9	1	2
Actual Warning	2	6	8
Actual Pass	2	4	29

D.2 CNN ternary classifiers: Median Grade (VW)

Tables D.9 to D.12 present the average results of early prediction with a CNN ternary classifier with classification categories based on the median passing grade.

Table D.9: Average values for CNN ternary classifier based on median passing grade (up to VW date)

Type of Data	**Train (%)**	**Test (%)**	**Epochs**
Intervals	42.26	38.06	152
Timestamp	52.54	47.94	256
Multivariable	71.98	54.89	282

Table D.10: Average error matrix for CNN ternary classifier based on median passing grade with Intervals data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	3	7	8
Actual Warning	3	9	11
Actual Pass	2	9	11

Table D.11: Average error matrix for CNN ternary classifier based on median passing grade with Timestamp data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	7	6	4
Actual Warning	3	12	8
Actual Pass	2	10	10

Table D.12: Average error matrix for CNN ternary classifier based on median passing grade with Multivariable data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	10	4	4
Actual Warning	3	9	10
Actual Pass	1	7	15

Tables D.13 to D.16 present the results of the best early prediction CNN ternary classifier model out of 100 trials, with the classification groups based on the median passing grade.

Table D.13: Best values for CNN ternary classifier based on median passing grade (up to VW date)

Type of Data	**Train (%)**	**Test (%)**	**Epochs**
Intervals	58.38	53.97	285
Timestamp	49.08	66.67	175
Multivariable	74.16	68.25	236

Table D.14: Best error matrix for CNN ternary classifier based on median passing grade with Intervals data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	8	6	5
Actual Warning	2	12	7
Actual Pass	3	6	14

Table D.15: Best error matrix for CNN ternary classifier based on median passing grade with Timestamp data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	15	2	6
Actual Warning	1	17	6
Actual Pass	2	4	10

Table D.16: Best error matrix for CNN ternary classifier based on median passing grade with Multivariable data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	10	3	7
Actual Warning	0	12	9
Actual Pass	0	4	18

D.3 Transformer ternary classifiers: Letter Grades (VW)

Tables D.17 to D.20 present the average results of early prediction using data up to the VW deadline with a transformer ternary classifier using classification categories based on the letter grades.

Table D.17: Average values for transformer ternary classifier based on letter grades (up to VW date)

Type of Data	**Train (%)**	**Test (%)**	**Epochs**
Intervals	53.03	42.54	225
Timestamp	65.28	44.44	83
Multivariable	65.33	43.08	91

Table D.18: Average error matrix for transformer ternary classifier based on letter grades with Intervals data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	1	1	15
Actual Warning	1	1	17
Actual Pass	1	1	24

Table D.19: Average error matrix for transformer ternary classifier based on letter grades with Timestamp data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	5	4	9
Actual Warning	4	4	12
Actual Pass	3	6	17

Table D.20: Average error matrix for transformer ternary classifier based on letter grades with Multivariable data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	5	3	9
Actual Warning	3	5	12
Actual Pass	3	5	17

Tables D.21 to D.24 present the results of the best early prediction transformer ternary classifier model out of 20 trials trained with data up to the VW deadline, using classification categories based on letter grades.

Table D.21: Best values for transformer ternary classifier based on letter grades (up to VW date)

Type of Data	**Train (%)**	**Test (%)**	**Epochs**
Intervals	39.63	55.56	500
Timestamp	64.94	52.38	75
Multivariable	56.20	50.79	76

Table D.22: Best error matrix for transformer ternary classifier based on letter grades with Intervals data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	0	0	12
Actual Warning	0	0	16
Actual Pass	0	0	35

Table D.23: Best error matrix for transformer ternary classifier based on letter grades with Timestamp data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	7	2	4
Actual Warning	7	4	12
Actual Pass	1	7	19

Table D.24: Best error matrix for transformer ternary classifier based on letter grades with Multivariable data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	4	1	10
Actual Warning	2	1	15
Actual Pass	0	3	27

D.4 Transformer ternary classifiers: Median Grade (VW)

Tables D.25 to D.28 present the average results of early prediction with a transformer ternary classifier with classification categories based on the median passing grade and input data up to the VW deadline.

Table D.25: Average values for transformer ternary classifier based on median passing grade (up to VW date)

Type of Data	**Train (%)**	**Test (%)**	**Epochs**
Intervals	47.88	37.19	102
Timestamp	65.73	42.25	84
Multivariable	60.31	42.33	86

Table D.26: Average error matrix for transformer ternary classifier based on median passing grade with Intervals data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	1	9	7
Actual Warning	1	13	10
Actual Pass	1	12	9

Table D.27: Average error matrix for transformer ternary classifier based on median passing grade with Timestamp data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	5	6	6
Actual Warning	3	10	10
Actual Pass	3	9	11

Table D.28: Average error matrix for transformer ternary classifier based on median passing grade with Multivariable data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	6	6	6
Actual Warning	4	10	9
Actual Pass	3	8	11

Tables D.29 to D.32 present the results of the best early prediction transformer ternary classifier model out of 20 trials, with the classification groups based on the median passing grade and input data up to the VW deadline.

Table D.29: Best values for transformer ternary classifier based on median passing grade (up to VW date)

Type of Data	**Train (%)**	**Test (%)**	**Epochs**
Intervals	37.40	47.62	75
Timestamp	62.72	55.56	75
Multivariable	59.91	55.56	98

Table D.30: Best error matrix for transformer ternary classifier based on median passing grade with Intervals data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	0	13	0
Actual Warning	0	30	0
Actual Pass	0	20	0

Table D.31: Best error matrix for transformer ternary classifier based on median passing grade with Timestamp data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	7	5	6
Actual Warning	4	12	8
Actual Pass	1	4	16

Table D.32: Best error matrix for transformer ternary classifier based on median passing grade with Multivariable data (up to VW date)

	Predicted Fail	Predicted Warning	Predicted Pass
Actual Fail	4	7	2
Actual Warning	0	18	12
Actual Pass	0	7	13

Appendix E

Colophon

This thesis is typeset in LaTeXusing a version of the custom template created by Steve Woodrow, modified by Dario Schor, and further modified by Kathryn Marcynuk in texstudio version 3.0.1. BibDesk version 1.6.3 was used to manage the references using BibTeX. The body of the report is written in 11 point Times New Roman, while the figure captions are printed in 10 point Arial.

The work was performed using macOS Mojave version 10.14.6 with a 2.3 GHz Intel Core i7 processor and 8 GB of 1600 MHz DDR3 memory using Python 3 in Spyder 4.1.5, and macOS Ventura version 13.3.1 with a Apple silicon M1 processor and 8 GB of LPDDR4 memory using Python 3 and TensorFlow 2.10.0.