People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Ghardaia
Faculty of Science and Technology
Laboratory of Mathematics and Applied Sciences
Department of Mathematics and Computer Science

MASTER THESIS

Presented to obtain the Master's degree in Computer Science
Specialization: Intelligent Systems for Knowledge Extraction

Validating AI Predictions in Education: A Comparative Study of Methods and Datasets

---

Presented by:
Toufik Bazemlal & Messaoud Mosbah

Jury Members:

Mr.Nacera BRAHIM	MAA	Univ. Ghardaia	President
Mr.Messaoud BEN GUENAN	MCB	Univ. Ghardaia	Examiner
Ms.Asma BOUCHEKOUF	MAA	Univ. Ghardaia	Thesis Supervisor

University Year: **2024/2025**

ABSTRACT

The proliferation of digital learning platforms has generated vast amounts of student interaction data, creating opportunities to leverage machine learning for early prediction of student performance and timely intervention for at-risk learners. This thesis investigates the effectiveness of traditional machine learning algorithms (Random Forest, Logistic Regression, Decision Tree, KNN) versus sequential deep learning models (LSTM, GRU, RNN) in forecasting student outcomes, with particular focus on capturing temporal patterns in learning behaviors.

The models were trained and evaluated on two real-world datasets: the UK OULAD Dataset and the Chinese TsinghuaX MOOC Dataset, using comprehensive performance metrics including accuracy, precision, recall, F1-score, and AUC-ROC across different course timeline stages.

Results show that sequential models, especially GRU, outperform traditional methods, achieving 93.09% accuracy on the OULAD dataset and 85.84% on the TsinghuaX dataset, particularly excelling in mid-course predictions. Temporal analysis highlights that predictive accuracy improves as more sequential data accumulates, emphasizing the value of temporal modeling in educational data mining.

These findings demonstrate the effectiveness of deep learning methods in modeling sequential educational data. They also support the development of more accurate early warning systems and adaptive learning interventions, ultimately enhancing student retention and success in online education environments.

Keywords: Student performance prediction, Machine Learning, Deep Learning, Sequential modeling, Educational Data Mining, Early warning system.

RÉSUMÉ

La prolifération des plateformes d'apprentissage numérique a généré d'importantes quantités de données sur les interactions des étudiants, offrant des opportunités d'exploiter l'apprentissage automatique pour prédire précocement les performances des étudiants et intervenir rapidement auprès des apprenants à risque. Cette thèse examine l'efficacité des algorithmes d'apprentissage automatique traditionnels (Forêt Aléatoire, Régression Logistique, Arbre de Décision, KNN) par rapport aux modèles d'apprentissage profond séquentiels (LSTM, GRU, RNN) dans la prévision des résultats des étudiants, avec un accent particulier sur la capture des motifs temporels dans les comportements d'apprentissage.

Les modèles ont été entraînés et évalués sur deux ensembles de données réelles : l'ensemble de données OULAD du Royaume-Uni et l'ensemble de données TsinghuaX MOOC de la Chine, en utilisant des métriques de performance complètes incluant l'exactit--ude, la précision, le rappel, le score F1 et l'AUC-ROC à différents stades de la chronologie des cours.

Les résultats montrent que les modèles séquentiels, en particulier le GRU, surpassent les méthodes traditionnelles, atteignant un taux de précision de 93,09% sur l'ensemble de données OULAD et de 85,84% sur l'ensemble de données TsinghuaX MOOC, en particuler lors de la prédictions aux milieu de cours. L'analyse temporelle révèle que la précision prédictive s'améliore à mesure que davantage de données séquentielles sont disponibles, soulignant l'importance du modélisation temporelle dans le domaine du data mining éducatif.

Ces résultats contribuent au domaine en démontrant l'efficacité des méthodes d'apprent--issage profond pour traiter les données éducatives séquentielles. ils soutiennent également le développement de systèmes d'alerte précoce plus précis et d'interventions d'apprentissage adaptatives, améliorant ainsi la rétention et la réussite des étudiants dans les environnements d'apprentissage en ligne.

Mots-clés : Prédiction des performances des étudiants, Apprentissage automatique, Apprentissage profond, Modélisation séquentielle, Fouille de données éducatives, Système dálerte précoce,

الملخص

أدى انتشار منصات التعلم الرقمي إلى توليد كميات هائلة من بيانات تفاعل الطلاب، مما خلق فرصاً لاستغلال التعلم الآلي للتنبؤ المبكر بأداء الطلاب والتدخل في الوقت المناسب للمتعلمين المعرضين للخطر. تبحث هذه الأطروحة في فعالية خوارزميات التعلم الآلي التقليدية (الغابة العشوائية، الانحدار اللوجستي، شجرة القرار، خوارزمية أقرب الجيران) مقابل نماذج التعلم العميق التسلسلية (LSTM، GRU، RNN) في التنبؤ بنتائج الطلاب، مع التركيز بشكل خاص على التقاط الأنماط الزمنية في سلوكيات التعلم.

تم تدريب النماذج وتقييمها على مجموعتي بيانات واقعيتين: مجموعة بيانات OULAD البريطانية ومجموعة بيانات Ts-inghuaX MOOC الصينية، باستخدام مقاييس أداء شاملة تشمل الدقة، والضبط، والاستدعاء، ودرجة F1، ومنحنى AUC-ROC عبر مراحل زمنية مختلفة من المقرر الدراسي.

تُظهر النتائج أن النماذج التسلسلية، وخاصة GRU، تتفوق على الطرق التقليدية، محققةً دقة بنسبة 93.09% على مجموعة بيانات OULAD ونسبة 85.84% على مجموعة بيانات TsinghuaX، وتتفوق بشكل خاص في التنبؤات في منتصف المقرر. يُبرز التحليل الزمني أن دقة التنبؤ تتحسن مع تراكم المزيد من البيانات التسلسلية، مما يؤكد على قيمة النمذجة الزمنية في تعدين البيانات التعليمية.

تُظهر هذه النتائج فعالية أساليب التعلم العميق في نمذجة البيانات التعليمية التسلسلية. كما أنها تدعم تطوير أنظمة إنذار مبكر أكثر دقة وتدخلات تعلم تكيفية، مما يعزز في نهاية المطاف الاحتفاظ بالطلاب ونجاحهم في بيئات التعليم عبر الإنترنت.

الكلمات المفتاحية: التنبؤ بأداء الطلاب، التعلم الآلي، التعلم العميق، النمذجة التسلسلية، تعدين البيانات التعليمية، نظام الإنذار المبكر.

Dedication

All praise to Allah, today we fold the days weariness and the journey's efforts between the covers of this humble work.

In the first place, I dedicate this work to my beloved parents, who have always given themselves and sacrificed for me, who helped me as best they could to succeed, who have always been there in my moments of anguish with their unwavering presence, steadfast support, and thoughtful counsel that have guided me through every step of this path.

To my cherished brothers and sisters, for their constant encouragement, unwavering moral support, and for standing by me throughout this academic journey.

To all my friends.

To those who reworded to us their knowledge simply and from their thoughts made a lighthouse guides us through the knowledge and success path, To our honoured teachers and professors.

I express my deepest gratitude and dedicate this humble work to all of you, with profound appreciation for your role in making this achievement possible.

Messaoud

Dedication

All praise and thanks to Allah, whose mercy and guidance have carried me through every step of this journey.

I dedicate this humble work to my beloved parents, whose sacrifices, and constant support have been the foundation of my success. Their prayers, patience, and encouragement have always been my source of strength and motivation.

To my dear brother and sister, thank you for your unwavering moral support and belief in me. Your encouragement gave me the confidence to keep moving forward, even in difficult times.

To my friends, who sharing in both the challenges and achievements of this academic journey.

And to my esteemed teachers and professors, whose dedication to sharing knowledge and inspiring learning has guided me throughout.

To all of you, I offer my sincere gratitude and dedicate this work in appreciation of your presence in my life and your contribution to this achievement.

Toufik

ACKNOWLEDGEMENT

First of all, all praise is due to Allah, who has guided us in our steps and given us the bravery , the will , and the fortitude to endure the challenges we have met throughout our journey .

We extend our sincere appreciation to our supervisor, Dr. Asma Bouchekouf, whose exceptional guidance and valuable feedback were instrumental in shaping this research. Your wisdom , encouragement, and constructive critiques made this thesis what it is today. We are also thankful to all our professors for their dedication and guidance throughout our studies.

We are deeply grateful to our colleagues and acquaintances for their friendship and collaboration. Your presence made this challenging journey possible.

Finally , we acknowledge all those who directly or indirectly contributed to the completion of this work. Your support has been greatly appreciated.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
KNN	k-Nearest Neighbors
LR	Logistic Regression
RF	Random Forest
SVM	Support Vector Machine
DT	Decision Tree
DL	Deep Learning
ML	Machine Learning
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
MSE	Mean Squared Error
PBC	Process-Behavior Classification
BCEP	Behavior Classification-based E-learning Performance
GBDT	Gradient Boosting Decision Tree
MOOC	Massive Open Online Course
VLE	Virtual Learning Environment
LA	Learning Analytics
EDM	Educational Data Mining
BART	Bayesian Additive Regression Trees
OULAD	Open University Learning Analytics Dataset
TSC	Time Series Classification
LMS	Learning Management System
ARIMA	Autoregressive integrated moving average
AUC-ROC	Area Under the Receiver Operating Characteristic Curve
1D	One-Dimensional
2D	Two-Dimensional
NLP	Natural Language Processing
lightGBM	Light Gradient Boosting Machine
FPR	False Positive Rate

INTRODUCTION

The fast development of digital technologies has led to major changes in many areas, especially in the field of education. In recent years, e-learning platforms such as Massive Open Online Courses (MOOCs) and Virtual Learning Environments (VLEs) have redefined how educational content is delivered and consumed by students.
As these platforms have become more popular, they have generated large amounts of data from learners. This data provides important opportunities to better understand the learning process using data-driven methods.

The availability of this data has supported the growth of two research areas: Learning Analytics (LA) and Educational Data Mining (EDM). Researchers in these fields use analytical tools to find insights and patterns about student behavior, predict learning outcomes, and detect students who may be at risk of failure or dropping out.

One important type of educational data is time series data. This includes repeated and ordered records such as weekly quiz scores, assignment results, login times, video views, discussion participation, and time spent on study materials. This type of data is useful because it can show changes in student behavior over time.

Time series classification (TSC) is different from regular classification tasks. In TSC, the order of the data matters, and the goal is to understand patterns that unfold over time. To handle this, models must be able to capture the sequence and timing of learner actions.

Despite this, most existing research still relies heavily on traditional machine learning approaches that overlook temporal dependencies. In contrast, sequential models like Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Recurrent Neural Networks (RNN) are specifically designed to capture time-based patterns. These models are better suited to detecting early warning signs of academic risk, such as declining engagement or erratic behavior, which are crucial for timely intervention.

This highlights an important research problem:

- How can educational time series data be effectively used to develop early prediction models for identifying at-risk students?

- how do temporal deep learning models (such as LSTM, GRU, and RNN) in capturing temporal dynamics and enhancing predictive accuracy?

To address these gaps, our study aims to:

- develop new models and compare both traditional and temporal models including LSTM, GRU, and RNN.
- and build early prediction models that can provide actionable insights during the early stages of a course .

This study is structured as follows:

- Chapter 1: Introduces the main concept necessary to understand the rest of the study.
- Chapter 2: Reviews the related work, providing an overview of previous studies
- Chapter 3: Outlines the research design and methodology.
- Chapter 4: Presents the implementation and evaluation results, along with the discussion.

CHAPTER 1

MAIN CONCEPTS

1.1 Introduction

This chapter introduces the foundational concepts and core pillars that underpin our research, providing the necessary context for understanding the subsequent chapters . It is organized into several key sections : defining education and its importance, exploring learning analytics and educational data mining, the significance of predicting at-risk students , and the role of time series data. Finally, we present the machine learning (ML) and deep learning (DL) methods employed in our study. The goal is to establish a foundation for using machine learning approaches to predict and prevent student failure through early intervention.

1.2 Definition and role of education

1.2.1 Definition of education

Education is a process of acquiring knowledge, skills, values, beliefs, and habits that enable an individual to develop and grow throughout their life, for themselves or for the betterment of society. Education encompasses various forms of learning, such as basic education , humanistic education, technical education, vocational education, financial education, social education, science and technology education, and more [1].

Education plays a crucial role in the overall development of individuals and societies. It helps individuals develop their critical thinking, problem-solving, decision-making, and communication skills, which are necessary for personal and professional growth. Education also contributes to the social, economic , and cultural development of societies by producing a skilled and knowledgeable workforce, promoting scientific and technological advancements, and preserving cultural heritage [1].

1.2.2 Importance of Education

a. Personal Benefits

- **Personal Development:** Education provides basic skills such as writing, reading, and speaking, as well as foster critical thinking, creativity, and problem-solving skills.

- **Economic Opportunity:** Higher education expands career opportunities and improves long-term earning potential.
- **Healthier Lifestyle:** Education can make you wiser about your health choices. Studies highlight that there is a correlation between education and improved health outcomes [2].

b. Societal Benefits

- **Economic Growth:** Education plays an essential role in economic growth, development, and stabilization of society. That is because national economic growth requires individual economic growth.
- **Promotes Equality:** Education achieves true equity where everyone gains access to the same educational opportunities; this would help close the gaps in socioeconomic status.
- **Civic Engagement:** Educated individuals are more likely to participate in community activities and civic life, fostering social cohesion and democratic participation [3].

1.2.3 Impact of Technological Advancements

Technological advancements have significantly impacted multiple fields, specifically education, leading to the spread of e-learning platforms. Platforms such as Massive Open Online Courses (MOOCs) and Virtual Learning Environments (VLEs) have gained popularity among students , with their adoption increasing dramatically. This growth has resulted in the generation of vast amounts of educational data, which attracts researchers to analyze it. Consequently, the fields of Learning Analytics (LA) and Educational Data Mining (EDM) have emerged and expanded. Researchers now leverage predictive learning analytics and educational data mining techniques to uncover patterns and trends, gain insights into educational experiences, predict student performance, and identify at-risk students [4].

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

1.3.1 Definition of Learning Analytics

Learning Analytics (LA) can be defined as the processes of measurement, collection, analysis, and reporting of data about learners and their contexts, for the purposes of understanding and optimizing learning and the environments in which it occurs [5]. This field has gained attention as a result of the spread of digital educational platforms and the availability of educational data . LA techniques can be applied to educational datasets to understand the learning process and experience and gain insight into them, in order to enhance the educational environment.

1.3.2 Learning Analytics Methodologies

- **Descriptive Analytics:** Summarizes past learning data to describe what happened. For example, a heatmap showing which parts of a course are most accessed.
- **Diagnostic Analytics:** Explains why something happened by identifying patterns and correlations in the data. For example, analyzing why students performed poorly in a specific module by correlating quiz attempts, forum participation, and video watch time.
- **Predictive Analytics:** Uses historical data to predict future student performance or behavior. For example, predicting whether a student is at risk of failing based on their engagement and early assessments, using machine learning algorithms.
- **Prescriptive Analytics:** Recommends actions to improve learning outcomes based on predictions. For example, suggesting personalized learning materials to a student who is falling behind, using recommendation systems [6].

1.3.3 Definition of Educational Data Mining EDM

Educational Data Mining (EDM) is an emerging multidisciplinary research area in which methods and techniques for exploring data originating from various educational information systems have been developed. In short, EDM is a new growing field that focuses on using data analysis tools on educational data to gain insights into the learning experience and improve educational outcomes [7].

1.3.4 EDM Stakeholders

- **Students:** Can receive personalized recommendations or advises (e.g., courses, resources).
- **Teachers:** Assess the effectiveness of materials provided and monitor student progress.
- **Study Adviser:** Identify students needing intervention.
- **Directors:** Analyze curriculum bottlenecks [1].

1.3.5 EDM tasks

- **Classification:** Profiling students by learning styles and preferences.
- **Predictive Modeling:** Creating models that forecast course completion or dropout.
- **Clustering:** Grouping students by behavior or performance, or grouping courses or assessments.
- **Frequent Pattern Mining:** Discovering common course combinations, popular paths in study programs, or frequent actions in LMS platforms.
- **Process Mining:** Mapping student pathways through curricula.
- **Collaborative Filtering:** Recommending learning materials based on the analysis of other learners' performance [7].

1.3.6 Difference between LA and EDM

Learning Analytics (LA) and Educational Data Mining (EDM) share similar goals: both aim to improve educational practice through the analysis of large-scale data. They focus on enhancing the analysis of educational data to support decision-making and practice in educational context. In terms of their major differences, in LA, leveraging human judgement is key, and automated discovery is a tool to accomplish this goal, while in EDM, automated discovery is key, and human judgment is a tool to accomplish this goal; LA has a stronger emphasis on understanding systems as a whole in full complexity, while EDM has a stronger emphasis on reducing components and analyzing individual components and the relationships between them [8]. Table 1.1 summarizes the differences between them.

	Learning Analytics (LA)	**Educational Data Mining (EDM)**
Goals	Understand learning processes to improve teaching and learning experiences	Develop models to discover patterns and support decision-making from educational data
Methods	Descriptive analytics, visualization tools, dashboards	Machine learning, classification, clustering, statistical analysis
Applications	Adaptive learning systems, personalized feedback, early warning systems	Detecting student behavior patterns, performance prediction, adaptive learning
Users	Educators, students, policymakers	Data scientists, researchers, administrators

Table 1.1: Comparison between Learning Analytics and Educational Data Mining

1.4 Predicting At-Risk Students

1.4.1 Definition of at-risk students

At-risk students are learners who are considered to have a higher probability of failing academically or dropping out of school due to various factors such as socioeconomic challenges, low academic performance, language or cultural barriers, or family-related problems. Identifying at-risk students early is a critical process that enables early interventions. This can be achieved by building predictive models using machine learning algorithms and deep learning [39].

1.4.2 Importance of early prediction

- Early identification of at-risk students allows early interventions (e.g., academic counseling, tutoring).
- Reduces dropout rates and improves students outcomes.

- Personalized support systems based on predictions.
- Reduces institutional costs and enhances overall educational quality.

1.4.3 Data and features

- **Demographics:** Age, gender socioeconomic status ...etc.
- **Academic Records:** Exam scores, assessments results ...etc.
- **LMS data:** Log data, forum participation, video views, clickstream activity .....etc.

1.4.4 Challenges:

- **Privacy and Ethics:** Student data is sensitive must be anonymized and secured.
- **Model Interpretability:** Teachers may need to understand the reasons behind predictions specially when use deep learning models.
- **Data Quality:** like missing value, incomplete logs, imbalanced data.

1.5 Time series analysis in educational data

Time series data refers to data that are collected or recorded sequentially at regular time intervals. In educational contexts, this can include weekly quiz scores, assessment results, clickstream activity, LMS login frequency, video watch time, forum participation rates, and time spent on course materials. Time series classification (TSC) problems are different from traditional classification problems in that the attributes are ordered. There are several TSC algorithms such as deep learning approaches RNNs, LSTM, GRU...etc. or statistical methods like ARIMA. These algorithms can serve as powerful tools for uncovering patterns in learner interaction that correlate with dropout risk [5].

1.6 Machine Learning and Deep Learning Approaches

1.6.1 Definition of machine learning

Machine learning (ML) is a subfield of artificial intelligence (AI) that provides computers with the ability to learn from data. This contrasts with traditional computing algorithms, which are explicitly programmed by humans and rely on pre-defined rules. The purpose of machine learning is to understand data to build data-driven models and programming through the systematic detection of statistically significant patterns in the available data. The impact of ML methods on research and practical applications in the educational sciences is still limited, it continues to grow as larger and more complex datasets become available through massive open online courses (MOOCs) and large-scale investigations [9].

1.6.2 Types of machine learning

Machine learning can be divided into different types:

a. Supervised Learning

Supervised learning is a machine learning technique that uses labeled data, where the environment has a set of corresponding inputs and outputs (x, y). The model learns from this data in order to predict the output y for a new input x [37]. Supervised learning includes two tasks:

- Regression: In this task, the output is continuous (quantitative). For example predicting house prices. The most famous algorithm is Linear Regression.
- Classification: In this task, the output is discrete (qualitative), such as class labels. For example, the classification of email as spam or not spam. The most famous algorithm is SVM (Support Vector Machine).

b. Unsupervised Learning

Unsupervised learning is a technique where the machine uses unlabeled data. In this case, the algorithm learns the internal representation or important features to discover relationships or structures within the input data. [37]. Two common tasks in this category are:

- Clustering: This task is based on dividing data into groups (clusters) such that the data are similar within each cluster and dissimilar from the other. The most famous clustering algorithms is K-means.
- Dimensionality Reduction : The aim of this task is to reduce the complexity or size of the data by reducing features dimensions while trying to maintain important one's dimensions. PCA (Principal Component Analysis) is the most famous algorithm.

c. Reinforcement Learning

Reinforcement Learning (RL) is a type of machine learning where an agent learns to make decisions by interacting with an environment. The agent takes actions in the environment, receives feedback in the form of rewards or penalties, and uses this feedback to improve its future actions. The most famous algorithm is Q-Learning [37].

1.6.3 Logistic regression

Logistic regression is a statistical and machine learning algorithm used for binary classification problems, situations where the output is one of two possible classes (e.g., spam or not spam, success or failure) [10].

It works by modeling the probability that a given input belongs to a particular class using the logistic (sigmoid) function equation1.1, which maps any real-valued number into the range (0, 1). The algorithm estimates the parameters (weights) of a linear equation and then applies the sigmoid function to the result. If the predicted probability is above a certain threshold (commonly 0.5), the algorithm classifies the instance as class 1; otherwise, class 0, figure 1.1 provide logistic regression example using sigmoid function. [11].

$$P(y = 1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)}} \tag{1.1}$$

where:

- $X = (x_1, x_2, ..., x_n)$ is the input feature vector
- β_0 the intercept
- $\beta_1, ..., \beta_n$ are the model coefficients

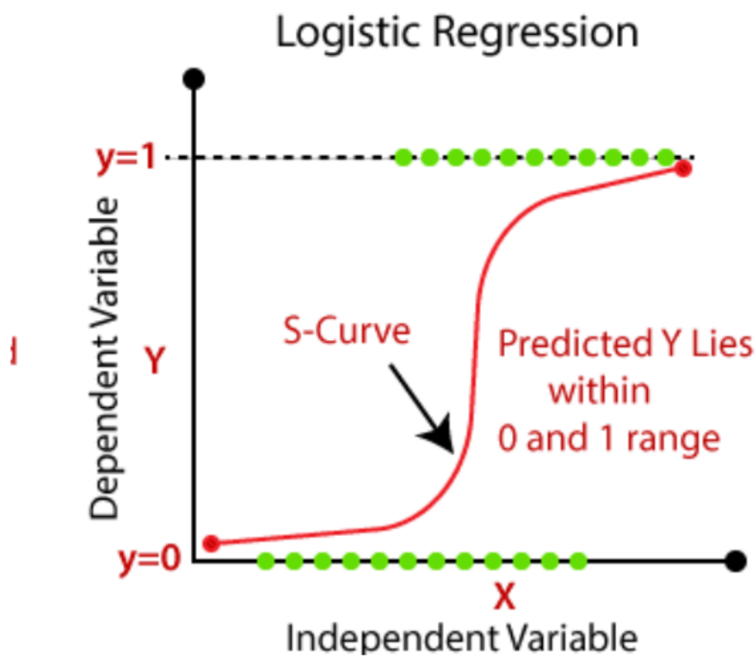

Figure 1.1: logistic regression using sigmoid function

1.6.4 K Nearest Neighbors

k-NN or K-Nearest Neighbors is a supervised classification algorithm. It works by finding the K closest training examples (neighbors) Typically uses Euclidean distance to find nearest neighbors to a given input and making predictions based on the majority class among those neighbors [12].

Choosing the right K is critical, too small can overfit, too large can underfit. KNN is a lazy learning algorithm meaning no training phase, all computation happens during prediction. It's easy to implement and interpret, but can be inefficient with large datasets or high-dimensional data, figure 1.2 shows an example of KNN [11].

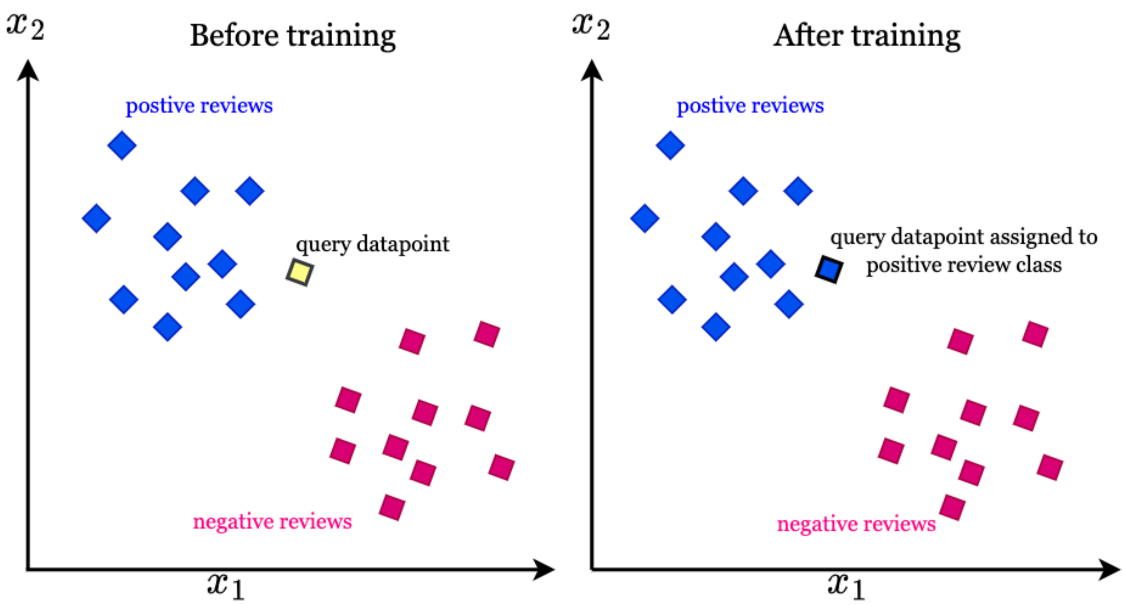

Figure 1.2: K Nearest Neighbors classification

1.6.5 Decision tree

A decision tree is a simple model for supervised classification. It is used for classify a single discrete target feature. Each internal node performs a Boolean test on an input feature (in general, a test may have more than two options, but they can be converted into a series of Boolean tests). The edges are labeled with the values of that input feature, and Each leaf node specifies a value for the target feature [13].

So, classifying an example using a decision tree is very intuitive, we traverse down the tree, evaluating each test and following the corresponding edge. When a leaf node is reached, we return the classification on that leaf. The figure 1.3 provide an example of Decision Tree. Decision trees are easy to interpret, can handle both numerical and categorical data, and don't require much data preprocessing. However, they can overfit if not pruned or regularized properly. [11].

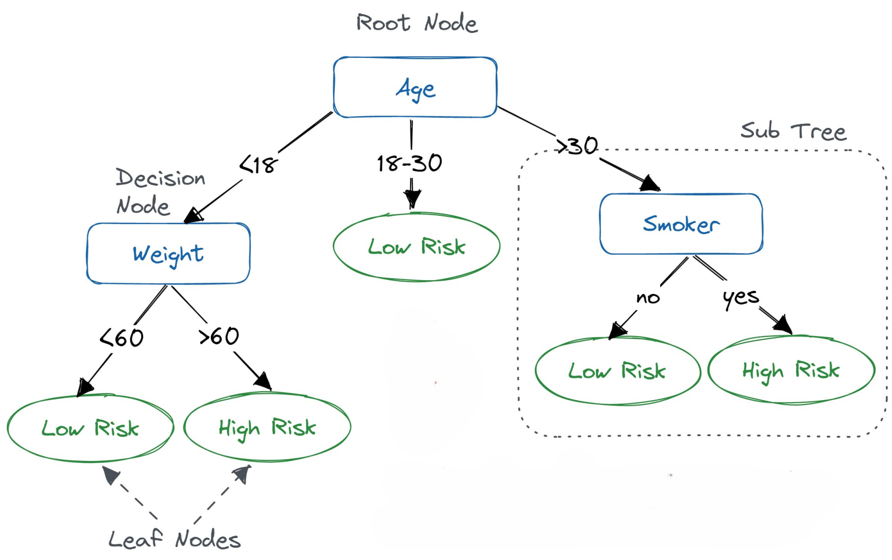

Figure 1.3: Decision Tree example

1.6.6 Random Forest

Random Forest is a powerful ensemble learning algorithm used for classification tasks. It operates by constructing multiple decision trees during the training process and aggregating their outputs to make a final prediction, typically using majority vote for classification. The core idea is to combine the predictions of several base estimators, each built with a certain degree of randomness, to produce a more accurate and stable prediction than any individual model. Each tree in the forest is trained on a random subset of the training data using a technique called bootstrap sampling. At each split in the tree, a random subset of features is considered, rather than using all available features. figure 1.4 provide an example of Random forest [11].

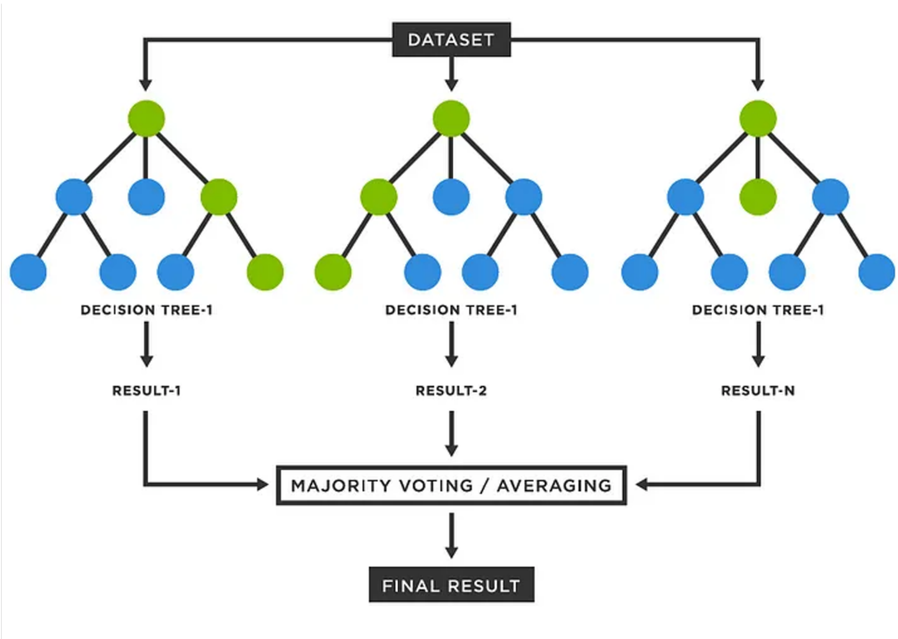

Figure 1.4: Ransom Forest Example

1.6.7 Deep learning definition

Deep learning is an advanced branch of machine learning that uses neural networks with many layers (also called deep neural networks) to model complex patterns in large datasets. What makes deep learning different from traditional machine learning is its ability to automatically extract relevant features from raw data, without needing for manual feature engineering.

Deep learning is inspired by the structure of the human brain, where layers of neurons process information step by step. These models are particularly powerful when it comes to working with high-dimensional and unstructured data like images, speech, or text. For example, deep learning has been responsible for major improvements in tasks such as image recognition, machine translation, and speech processing [14].

1.6.8 Neural Network

A neural network is an interconnected assembly of simple processing elements, called units or nodes, whose functionality is loosely based on the biological neuron. The processing capability of the network is stored in the inter-unit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns [15].

1.6.9 Artificial neuron

Artificial neural networks are popular machine learning techniques that simulate the mechanism of biological learning, where each neuron from a network can be implemented as shown in Figure 1.5 , its parameters are:

- Input connections (inputs) : is a vector $(a_1, a_2, ..., a_n)$ with weights $(w_1, w_2, ..., w_n)$ each input is multiplied by its weight.
- Pre-activation function z : is a summation function that sums weights after multiplies each of input by their own associated weight, with the addition of the bias (used to adjust the output along with the weighted sum of the inputs to the neuron) $z = \sum_{i=1}^{n} a_i w_i + b$
- Activation function g : transforms the pre-activation; g(z).
- Output : output the final activation; aout = g(z).

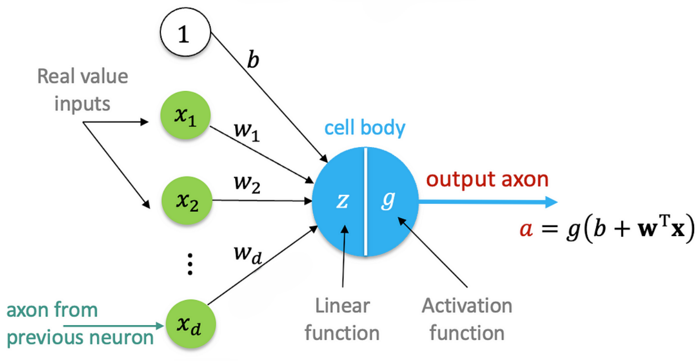

Figure 1.5: artificial neuron

Activation function

Are nonlinear mathematical functions that calculate the level of neuron activation. The function is chosen according to the problem and outputs, we list the most popular activation function:

- Sigmoid : Normalizes the output of each neuron, its output value is in the range [0,1].

$$g(z) = \frac{1}{1 + e^{-z}} \tag{1.2}$$

- Rictifier (ReLU) : It's gives 0 if it receives a negative input, and returns the same positive value x otherwise.

$$g(z) = max(z, 0) \tag{1.3}$$

- Hyperbolic Tangent (TanH) : Normalizes the output of each neuron, its output value is in the range [-1,1].

$$g(z) = \frac{1 - e^{-2z}}{1 + e^{-2z}} \tag{1.4}$$

- Softmax : Used when outputs are multi class (multi-classification) and its output value is in the range [0,1].

$$g(z_i) = \frac{e^{z_i}}{\sum_{j=0}^{n} e^{z_j}}, \quad i = 1, 2, 3, \ldots, n \tag{1.5}$$

1.6.10 Architecture of ANN

The basic architecture of an artificial neural network consists of three basic parts, figure1.6 shows that:

- Input layer : it is the first layer responsible for receiving information (data) from the external environment.
- Hidden (intermediate or invisible) layers : these layers perform most of the basic work in a network. The layers are made up of neurons responsible for excreting features.

- Output layer : after processing with neurons in the previous layers, this layer produces and delivers the final network outputs.

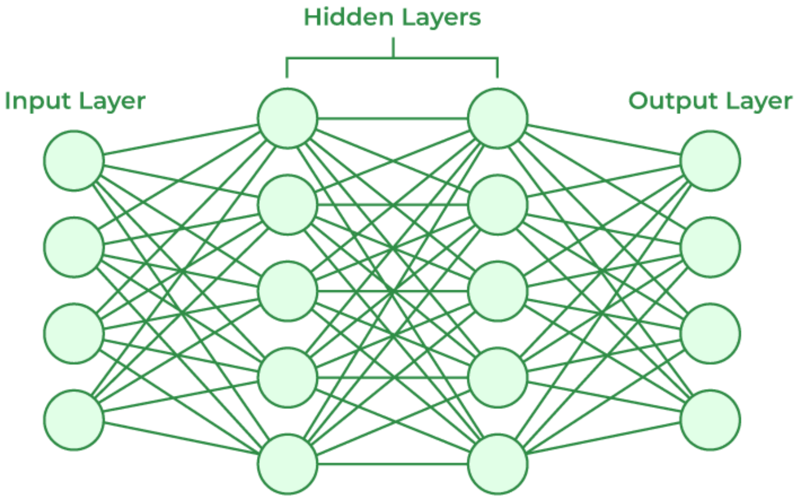

Figure 1.6: ANN architecture

1.6.11 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are a class of neural networks that are used for sequence modeling tasks, such as Natural Language Processing (NLP) and speech processing applications.

RNNs are called recurrent because they perform the same task for every element in a sequence. RNN can be described in other way that they have a “memory” which captures information about what has been calculated so far. Figure1.7 shows a graphical illustration of a structure of an RNN [16].

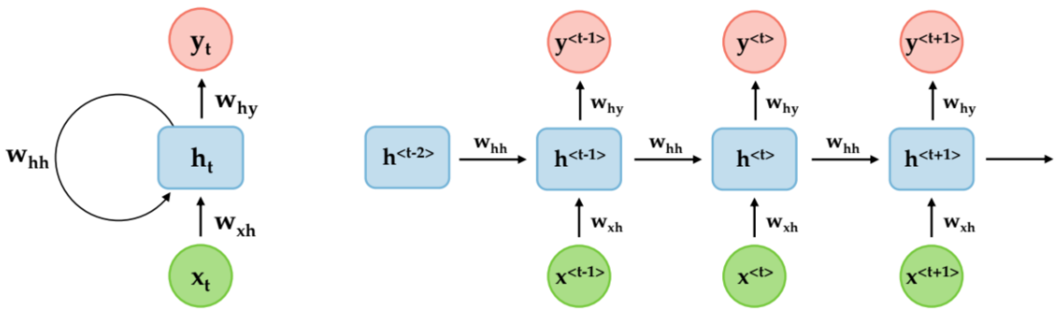

Figure 1.7: RNN architecture

- $x^{\langle t \rangle}$: designates the input of the network at time step t.

- $x^{<h>}$: represents the hidden state of the network at time step, t and it is computed by equation:
$$h^{\langle t\rangle} = g_1\left(W_{hh}h^{\langle t-1\rangle} + W_{xh}x^{\langle t\rangle} + b_h\right) \tag{1.6}$$
- $y^{<t>}$: Signified the output of the network at time step t and it is computed by equation:
$$y^{\langle t\rangle} = g_2\left(W_{hy}h^{\langle t\rangle} + b_y\right) \tag{1.7}$$

Where $w_{xh}, w_{hh}, w_{hy}, b_h, b_y$ are coefficients that are shared temporally and g_1, g_2 activation functions.

1.6.12 Long Short-Term Memory (LSTM)

LSTM was first introduced in 1997 by Sepp Hochrieiter and Jürgen Schmidhuber. LSTMs are capable of bridging time intervals in excess of 1000 time steps, even in the case of noisy incompressible input sequences, without loss of short time lag capabilities. The architecture enforces constant error flow through internal states of special unit known as the memory cell.

There are three gates to the cell: the forget gate, input gate, and output gate. These gates are sigmoid functions that determine how much information to pass or block from the cell. Sigmoid functions takes in values and outputs them in the range of [0,1]. In terms of acting as a gate, a value of 0 means let nothing through, and a value of 1 means let everything through. These gates have their own weights that are adjusted via gradient descent. the figure1.8 provide LSTM memory cell with their gates[17].

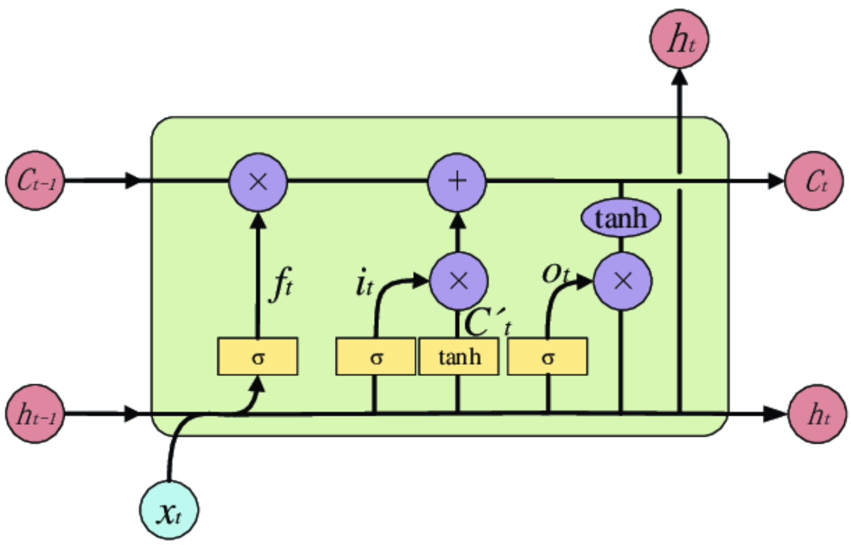

Figure 1.8: LSTM memory cell

- Cell state (ct): Vector of fixed shape with random value initialization. It contains the information that was present in the memory after the previous time step.
- Forget gate (ft): Changes the cell state, intending to eliminate non-important values from previous time steps. This helps the LSTM network to forget the irrelevant information that does not have any impact on the future price prediction.

- Input gate (it): Changes the cell state with the aim of adding new information about the current time step. It adds new information that may affect the stock price movement.
- Output gate (ot): Decides what the next hidden state should be. The new cell state and the new hidden is then carried over to the next time step. Returns the final relevant information, which will be used for stock price prediction.
- Hidden state (ht): It is calculated by multiplying output gate vector by cell state vector.

1.6.13 Gated Recurrent Unit (GRU)

To address the vanishing gradient problem in Recurrent Neural Networks (RNNs), the Gated Recurrent Unit (GRU) was proposed by Cho et al.[33]. The GRU shares many similarities with the Long Short-Term Memory (LSTM) model, such as gating mechanisms to control the flow of information. However, unlike LSTM, GRU does not include a separate output gate, making it computationally simpler while retaining comparable performance.

As shown in Figure 1.9, the GRU architecture includes two main gates: the *update gate* and the *reset gate*. These gates determine the information to be passed through the network at each time step, enabling the GRU to capture long-term dependencies efficiently.

- **Update gate** (z_t): Determines how much of the past information needs to be passed along to the future.

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z) \tag{1.8}$$

- **Reset gate** (r_t): Controls how much of the previous hidden state to forget.

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r) \tag{1.9}$$

- **Candidate hidden state** ($\tilde{h}_t$): Computes a new memory content using the reset gate.

$$\tilde{h}_t = \tanh(W_h[r_t \odot h_{t-1}, x_t]) \tag{1.10}$$

- **Final memory** (h_t): Combines the previous hidden state and the candidate hidden state.

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{1.11}$$

- **Output vector** (o_t): The output at time t based on the hidden state.

$$o_t = \sigma_o(W_o h_t + b_o) \tag{1.12}$$

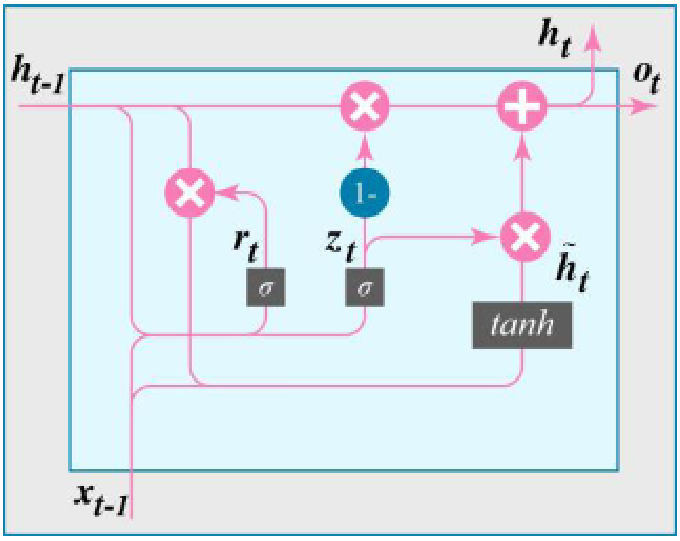

Figure 1.9: The structure of a GRU unit [32].

The use of fewer gates in GRU, compared to LSTM, allows for a simpler design and faster training time, while still maintaining the capability of capturing dependencies in sequential data, making it highly effective for classification and time-series prediction tasks.

1.7 Conclusion

This chapter explained the main concepts and tools used in our research. We looked at how education is changing with technological advancements. We explained what Learning Analytics and Educational Data Mining are, and how they are used to understand student behavior and predict who might be at risk of failing. We also reviewed different machine learning and deep learning methods, especially those that work with time-based data such as LSTM. The next chapter reviews related works. Explores existing research efforts, methodologies, results achieved, and the gaps our study aims to address.

CHAPTER 2

RELATED WORKS

2.1 Introduction

This chapter presents a review of existing research on student performance prediction using machine learning and deep learning techniques. It is organized as follows: Literature Reviews provides a comprehensive overview covering traditional machine learning approaches, deep learning methods, time-series analysis, and early prediction models. The section on *Traditional Machine Learning Approaches* examines techniques including decision trees, random forests, and support vector machines. *Deep Learning Approaches* explores advanced neural network methods and their applications in educational contexts. *Time-Series Data and Temporal Models* discusses the importance of capturing sequential learning behaviors over time. *Early Prediction Models* focuses on methods that enable timely interventions for at-risk students, and finally, *Summary and Research Gaps* identifies current limitations and opportunities for improvement in existing methodologies.

The goal is to understand the current research landscape and highlight gaps that demonstrate the need for better temporal modeling techniques in educational analytics.

2.2 Literature Reviews

Prediction of student academic performance has gained significant attention from researchers in educational data mining (EDM) and learning analytics (LA). With the rise of online educational platforms that record student information such as demographics, assessments, interaction data. a large amount of data has been available, enabling data-driven approaches for early prediction and timely interventions that improve retention and learning outcomes. Several studies demonstrate that traditional machine learning methods such as logistic regression, decision tree, random forest achieve acceptable accuracy using static features, while they struggle with time series data. In contrast, temporal models offer useful insights about learners behaviours.

2.2.1 Traditional Machine Learning Approaches

According to [20], the most used machine learning algorithms include decision tree, random forest, support vector machine, and logistic regression. While deep learning approaches have also seen increased adoption in recent years and have showed good results compared to traditional ML. study [19] build a prediction models using non personal data, where it compare traditional ML algorithms (SVM, decision tree, naïve bayes) with deep learning approaches, and to enhance results authors apply feature engineering techniques, ML algorithms achieve good results with accuracy up to 92% with decision tree while deep learning achieves slightly better result with 93% of accuracy, The authors highlight the importance of using nonpersonal data to avoid ethical concerns, while it lack of benefits in the dataset due to the reduction of activity interactions into a single feature, "click-sum." This simplification may prevent the discovery of important and meaningful learning activities.

Study [22] compares ensemble learning algorithms with ML algorithms, where they demonstrates the effectiveness of ensemble learning in predicting student performance. The ensemble learning method LightGBM achieved the best performance with 94% of accuracy compared to traditional machine learning algorithms (Naïve Bayes, KNN, DT, LR, SVM, DNN) and others ensemble learning methods (Random Forest, AdaBoost, GBDT, XGBoost).

2.2.2 Deep Learning Approaches

Study [23] employed a deep learning model to predict student performance using student interaction data on VLEs, the authors minimized the number of features and select those that are more effective. The model achieve a high accuracy of 96%. While the study lacks of comparison with machine learning approaches, it also lacks comparison with other deep learning approaches.

The authors in study [8] proposed the BCEP prediction framework which includes feature selection and feature fusion. The authors introduced PBC model, which categorizes interactions behavior into four stages. They conducted experiments comparing three groups: one with all interaction data, one with feature selection, and one using BCEP framework (feature selection and fusion based on the PBC model) and they used SVM Naïve Bayes KNN SoftMax. The results showed that the BCEP model outperformed the others in terms of accuracy, F1-score, Kappa. Further experiments compared the proposed PBC classification model with three other common behavior classification methods. The PBC model demonstrated superior and more stable performance.

Despite the good results achieved by precedent studies, there is three main limitations can be observed. Firstly interpretability, while complex models such as deep learning and ensemble methods often achieve higher accuracy, they are typically black-box in nature, making it difficult to explain how specific predictions are made or what features make student success or failure. This lack of interpretability can hinder the adoption of these models by educators and policymakers who require transparent and actionable insights. Secondly generalizability, many studies develop and test models only on the OULAD dataset. As a result, models that perform well in one context may not maintain the same effectiveness when applied to different student populations, learning management systems, or course structures. thirdly the use of time series data, Many studies overlook the use of time series data even when their datasets include temporal information. This limits the ability to capture trends, and sequential patterns that could improve predictions and insights.

2.2.3 Time-Series Data and Temporal Models:

Time series data refers to data collected over time; this data can be log interactions. Time series classification (TSC) problems are different from traditional classification problems because the attributes are ordered. There are several TSC algorithms such as deep learning approaches RNNs, LSTM, GRU ...etc or statistical methods like ARIMA.

This article [5] explores how student interaction data, specifically clickstream logs from learning management systems, can be utilized to predict academic performance. Clickstream data was aggregated into weekly and monthly features. The researchers compared traditional machine learning models (Logistic Regression, Random Forest) with deep learning approaches (LSTM, 1D-CNN). LSTM outperformed other models, achieving up to 89.25% (±0.97%) accuracy with weekly data, with the authors emphasizing its strength in handling sequential patterns. This study employed only a single temporal model, whereas incorporating multiple comparative approaches would have been beneficial for validating the results and potentially improving accuracy.

2.2.4 Early Prediction Models

Early prediction is essential for early intervention where predictions are needed as early as possible to enable timely and effective support. This study [24] proposes an early prediction model for student success in MOOCs by integrating both behavioral and demographic indicators using the Open University Learning Analytics Dataset (OULAD). The authors employ Bayesian Additive Regression Trees (BART) to classify students into four categories Withdrawn, Fail, Pass, and Distinction achieving accuracies of 81%, 80%, 69%, and 92%, respectively.

Another notable study [31] proposes a predictive model that identifies at-risk students at different percentages of course completion using machine learning and deep learning algorithms. The model analyzes demographic data, assessment scores, and clickstream behavior to enable timely intervention. The results showed a progressive improvement in accuracy from 59% using only demographic data to 92% when trained on complete course data with Random Forest performing best. This study emphasizes the value of early-stage data and feature engineering to improve prediction and intervention strategies for at-risk students. However, this study is limited by its use of demographic data, which may raise ethical concerns regarding privacy and bias.

2.2.5 Summary and researcher gaps

The table 2.1 provide a complete comparison of various studies that focus on predicting student performance using OULAD dataset. Most studies successfully predicted student performance, achieving high accuracy up to 90%. Several machine learning methods were employed, including tree-based algorithms, Support Vector Machines (SVM), Logistic Regression (LR), ensemble learning, as well as deep learning approaches.

However, most studies used the complete course data, which poses a limitation for real-world applications, especially in the context of early intervention, where predictions are needed as early as possible to enable timely and effective support.

There is a noticeable lack of temporal methods such as LSTM, TSF, GRU, etc. these methods are capable of capturing and discovering sequential and time-dependent patterns. Interaction data between learners and resources provided by Virtual Learning

Environments VLEs collected over time can be considered as time series data that should be treated by temporal models.

Overall, the related works highlight two key gaps: the underutilization of temporal methods in prediction tasks and the scarcity of early prediction models. Our study addresses these gaps by employing temporal models, comparing them with traditional approaches, and developing a robust and efficient early prediction model that enables timely interventions.

Study	**Objective**	**Methods**	**Results**
[19]	Forecast whether a student will pass a course avoiding the use of personal/sensitive data.	ANN, SVM, Naïve Bayes, Random Forest	VLE interaction features achieved high AUC values (up to 0.91 for dropout prediction and 0.93 for result prediction with GBM).
[23]	To predict student academic performance based on interactions in Virtual Learning Environments (VLE).	Deep Neural Network	High accuracy with 96%
[22]	Predict student performance using LightGBM	Naïve Bayes, KNN, DT, LR, SVM, DNN, LightGBM	LightGBM outperformed other ML algorithms with 94.1% accuracy.
[25]	Predict academic success/failure using diverse student data	SVM, KNN, C4.5	Predicted academic success/failure using diverse student data
[24]	Predict student success early using behav-ioral/demographic data for finer classification (Withdrawn, Fail, Pass, Distinction).	BART (Bayesian Additive Regression Trees), Decision Tree, Random Forest	BART outperformed other models.
[5]	Predicting student performance using clickstream data and machine learning	Machine Learning (LR, k-NN, RF, GBT) and Deep Learning (1D-CNN, LSTM)	LSTM achieved the highest accuracy (89.25%). With weekly data

Study	Objective	Methods	Results
[8]	Predicting student performance behavior classification-based e-learning performance (BCEP) prediction framework and process-behavior classification (PBC) model.	SVC, Naïve Bayes, Softmax, KNN	BCEP framework achieved 95.44%–97.40% accuracy
[31]	Identifies at-risk students at different percentages of course	RF, SVM, KNN, ET, AdaBoost Classifier, Gradient Boosting Classifier, and ANN models	Random Forest achieved the best performance with accuracy from 59% to 92%
[38]	Predict students at risk of failure and enable early prediction using limited weeks of course data	Logistic Regression, Decision Tree, SVM, KNN, ANN, Gradient Boosting, and LSTM	LSTM outperformed other models even with only 5 weeks of data

Table 2.1: Related Work Summary

2.3 Conclusion

This chapter reviewed existing research in student performance prediction, examining approaches from traditional machine learning to deep learning methods. While many studies achieve good accuracy, most have significant limitations for practical educational applications.

Traditional machine learning methods like decision trees and logistic regression provide good interpretability but struggle with temporal patterns in learning data. Deep learning approaches show superior performance but lack explainability, making it difficult for educators to understand predictions and implement interventions.

The review revealed two critical research gaps: the under-utilization of temporal methods despite the sequential nature of learning data, and the scarcity of early prediction models that work with limited initial data. Most existing studies rely on complete course data, which reduces their practical value for timely interventions.

The next chapter presents the methodology used to address these gaps, covering data selection, preprocessing, model development with grid search, and evaluation metrics.

CHAPTER 3
METHODOLOGY

3.1 Introduction

This chapter presents the methodology employed to achieve the objectives of developing and comparing predictive models for identifying at-risk students in online learning environments using time series data. It outlines the research design, proposed system, data preparation, evaluation metrics, and experiments. By leveraging two datasets, the OULAD dataset and the TsinghuaX MOOC Dataset, this study compares traditional machine learning models (Random Forest, Decision Tree, Logistic Regression, KNN) with the deep learning models (RNN, LSTM, GRU) across one-dimensional and two-dimensional data representations. Through this comprehensive approach, our goal is to identify the most accurate and generalizable model for predicting student performance , enabling timely interventions to support student success.

3.2 Research Design

This research aims to develop predictive models to identify at-risk students in online learning environments by analyzing time-series data from Virtual Learning Environments (VLEs). The study compares traditional machine learning methods—such as Random Forest, Decision Tree, Logistic Regression, and K-Nearest Neighbors (KNN)—with deep learning models designed for sequential data, including Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Recurrent Neural Networks (RNN).

Two datasets are utilized in this research: the UK Open University Learning Analytics Dataset (OULAD) and the Chinese TsinghuaX MOOC dataset. The research design is structured into two main parts:

Part One: This phase focuses on data preprocessing and model development. Student interaction data is transformed into two input formats:

- A **one-dimensional (1D) model representation** that aggregates total clicks over time.
- A **two-dimensional (2D) model representation** that capturing activity-type-specific clicks over time.

Machine learning and deep learning models are trained on both representations. Hyperparameter optimization is conducted using Grid Search, and models are evaluated using classification metrics such as Accuracy, Precision, Recall, F1 Score, and AUC-ROC.

Part Two: This phase investigates early prediction by evaluating model performance at various stages of course progression (10% to 100% of the course timeline). This allows assessment of how early in the course reliable predictions can be made, which is essential for timely interventions and student support.

3.3 Proposed System

In this section we proposed a system aims to predict student risk status in online learning environments. The system follows a structured process contain of several key phases, as illustrated in Figure 3.1. The process begins with data selection, where two datasets were chosen: the OULAD dataset and the TsinghuaX MOOC dataset. These datasets provide detailed information on student interactions, academic performance, and engagement over time.

Once the data was selected, the next step is data preprocessing, which involved cleaning the data to handle missing values and inconsistencies. The activity records were then grouped by individual students to form time-series data. After cleaning and organizing the data, various transformations was applied to prepare it for modeling. A critical transformation involved reshaping the data into two formats:

- A two-dimensional (2D) representation, where each student is represented as a table with weeks as rows and learning activities as columns.
- A one-dimensional (1D) representation, where weekly values are aggregated to form a simplified feature vector for each student, resulting in a table where rows represent students and columns represent weeks.

Next, the model training phase involved applying both deep learning and classic machine learning models:

- Deep learning models such as GRU, LSTM, and RNN were used due to their ability to model time-series data and capture sequential patterns in student behavior.
- Traditional machine learning models, including Random Forest, Decision Tree, Logistic Regression, and K-Nearest Neighbors, were also tested to explore simpler and potentially more interpretable approaches.

To improve model performance, hyperparameter tuning was performed using a grid search strategy, allowing for optimal parameter selection based on validation results.

Once the best parameters were identified, the models were trained on the training dataset. During the training phase, key performance metrics — including accuracy, precision, recall, F1-score, and AUC-ROC — were recorded to evaluate the effectiveness of each model in identifying at-risk students.

Finally, the best-performing model was selected and used to make predictions on new student data. The ultimate goal of this system is to enable early intervention by identifying students at risk of failure or dropout based on their learning behavior. This allows educators to provide timely support and improve overall student success rates.

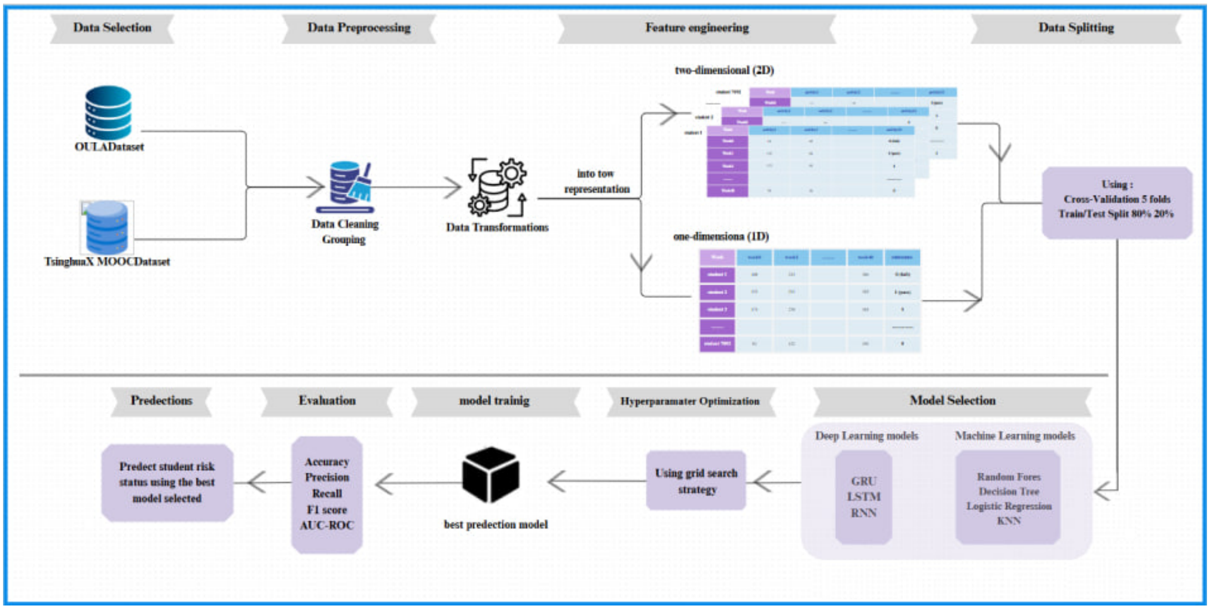

Figure 3.1: Architecture of the proposed system

3.4 Data preparation

3.4.1 Data selection

Our study leverages two datasets to analyze student engagement and predict performance in online learning environments. The first dataset is the Open University Learning Analytics Dataset (OULAD), this dataset is prepared by Open University (OU) of United Kingdom (UK), It is a subset of real students anonymized information, this dataset is highly regarded in the research community due to its breadth and quality. It provides detailed information on students demographics, course enrollments, and interactions within the Virtual Learning Environment (VLE). The OULAD is publicly available under a CC-BY 4.0 license, and has been officially verified by the Open Data Institute (ODI), ensuring its credibility and openness for research use. Access to this dataset can be obtained through the official ODI platform [35], can download it from this URL: https://www.kaggle.com/api/v1/datasets/download/anlgrbz/student-demographics-online-education-dataoulad

Figure 3.2 Overall dataset structure. The student is linked with the information about his/her demographics and registrations for the modules. For each student-module-presentation triplet, the dataset contains the results of the students assessments and logs of student interactions with VLE [35].

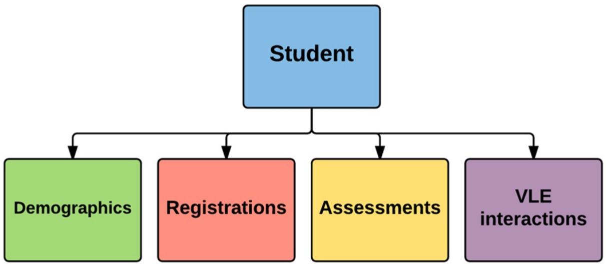

Figure 3.2: Overall OULAD structure [35]

These features are crucial for understanding the multifaceted nature of student learning and for building predictive models. So some features may contain missing or incomplete data, necessitating careful preprocessing and handling during analysis. The dataset covers information from 32,593 students across 22 module presentations during 2013 and 2014, over a 9 month period, resulting in a total of over 10 million clickstream entries [35]. Students engaged with 7 different modules, each offered at multiple points during the year, the clickstream data within the VLE specifically tracks student activities such as logging in, accessing content, participating in discussions, and navigating course pages, providing critical behavioral insights into the learning process [35]. The dataset comprises multiple interrelated tables, with key ones being studentInfo, studentVle, and courses. For this study, a focus was placed on the studentVle table, which provides time-stamped records of student interactions with various Virtual Learning Environment (VLE) activities. Each record includes a student ID, date, activity type, and number of clicks, to better visualize the dataset structure, Figure 3.3 illustrates how the 7 interconnected tables relate to each other :

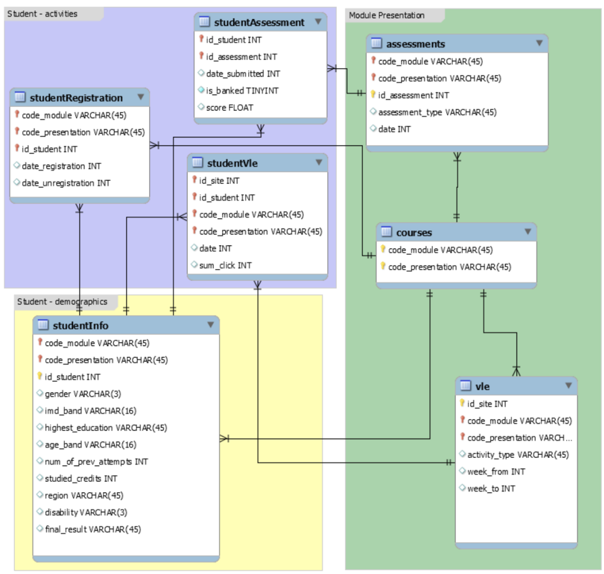

Figure 3.3: Schema of the OULAD Dataset Tables [36]

- assessments: it contains data related to assessments and final exams in course presentations, the table consists of 206 rows.
- courses: it contains the list of all courses and presentations, the table consists of 22 rows.

- studentAssessment: it contains the results of students' assessments (usually, final exam results are missing), the table consists of 173,912 rows.
- studentInfo: it contains demographics and students' final result in each course they studied, the table consists of 32,593 rows.
- studentRegistration: it contains the time when the students registered for (and eventually unregistered from) a course presentation, it consists of 32,593 rows.
- studentVle: it contains students interactions with the LMS, the table consists of 10,655,280 rows.
- vle: it contains data related to the online teaching material provided by the LMS, the table consists of 6,364 rows.

On this dataset we focused on the FFF module for this study because of it represents a Science Technology Engineering and Mathematics (STEM) course with a substantial number of enrolled students 7,762 providing a robust sample for analysis. The modules data includes diverse learning activities and assessment types, offering a comprehensive view of student engagement and performance. Additionally, the FFF module structure and content are conducive to exploring the impact of various factors on student success, making it an ideal candidate for this research [35].

The second dataset it is the TsinghuaX MOOC dataset.so it is a chinese dataset sourced from moocdata.cn , primarily derived from Massive Open Online Courses (MOOCs) offered on the XuetangX platform , represents a large-scale educational repository containing detailed student interaction logs across multiple courses spanning from 2013 to 2024 (11 years) [26]. It contains approximately 100–150 unique courses covering subjects such as Computer Science, Mathematics, Physics, Chinese, English, Economics, and Education , with each course typically consisting of 10–15 modules [27]. Across all datasets, there are an estimated 500,000 to over 1,000,000 unique students , generating 100 million to 500 million total interaction events , including video views, quiz submissions, and clickstream logs [28].

On the *the TsinghuaX MOOC dataset.*, we focused on two instances of the same course: TsinghuaX+30640014+2016_T2 and TsinghuaX+30640014+2015_T2. Together, these offerings include approximately 10,000–20,000 enrolled students, with around 40%–60% active users, and span a duration of about 12–14 weeks per offering, generating over 5 million interaction events combined [29].

Each dataset instance typically includes six structured files:

- `student.csv`: Contains student-level metadata and performance data.
- `log_data.csv`: Clickstream logs capturing student interactions with the platform.
- `video.csv`: Detailed records of video watching behavior.
- `quiz.csv`: Student quiz and problem submission logs.
- `forum.csv`: Forum activity including posts, replies, and comments.

- `course_structure.json`: Hierarchical structure of the course modules, sections, units, and components.

These files enable granular analysis of learning behaviors. The dataset has been widely used in Educational Data Mining (EDM) and Learning Analytics (LA) for tasks such as dropout prediction, engagement modeling, behavioral clustering, and adaptive learning systems [30].

Finlay, we selected these two datasets the OULAD dataset from UK, and the TsinghuaX MOOC dataset from China, because of they allow us to build a more generalizable model that can work across different regions, cultures and languages. So also we select them because of they public to access and use, and they offer rich and detailed information about student behavior, including demographics, assessment performance, and the most important the interaction logs with the online learning platforms. And also the size and the structure of these datasets provide enough variety and depth to build and test meaningful predictive models.

So these datasets are created ensuring its credibility and openness for research use. Having this kind of diversity in our data makes our model stronger and more likely to work well not just in one specific place or group, but in a wide range of online learning environments around the world.

3.4.2 Data preprocessing

Preprocessing plays a vital role in preparing raw educational data for machine learning and deep learning models. It ensures that the input data is structured consistently, noise is minimized, and temporal or behavioral patterns are captured effectively-all of which significantly impact model accuracy and generalizability. In this study, we applied a unified preprocessing pipeline to two distinct MOOC datasets: the Open University Learning Analytics Dataset (OULAD) and the TsinghuaX MOOC Dataset, enabling direct comparison of sequence-based models under the same structural and architectural assumptions. So it is the most importance phase to preparing the data to training the model so this phase is directly influences model accuracy and generalizability. The preprocessing included multiple transformations aimed at cleaning and preparing the data for machine learning and deep learning tasks, so we followed an identical preprocessing to ensure consistency across both datasets.

For our study using the OULAD dataset, we use two tables: studentInfo and studentVle. We start by selecting data from the studentVle table and grouped it by the code_module field. From this, we focused specifically on the FFF module, which includes the presentations 2013B, 2013J, 2014B, and 2014J.

Next, we merged this grouped data with the studentInfo table using the common fields id_student, code_module, and code_presentation. After merging, we grouped the resulting dataset by id_student, final_result, and code_presentation. This process resulted in 7,762 unique student records.

The date feature in the dataset is originally recorded in days. but, for the purpose of our study, we needed the data in weekly intervals. as result in a total of 41 weeks of data.

We prepared the data in two different formats, the first model student interaction data is aggregated into a one-dimensional array. the second model interaction data is structured into a two-dimensional array.

For the TsinghuaX MOOC dataset, we combining train_log.csv and test_log.csv table into one DataFrame with about 42 million rows of student activity. Next, we grouped the data by course_id and found 247 courses. we select the course with high the number of interaction records and the most stable performance and lowest overfitting was: course-v1:TsinghuaX+30640014+2015_T2 and its 2016 version. After selecting the course, We then added pass/fail labels from train_truth.csv and test_truth.csv, matching them to each student using enroll_id.

We then prepared the time column by converting it from string to datetime format. We removed any invalid or missing timestamps. A new date column was created to group student activity by day. Finally, we generated a continuous date range from the earliest to the latest date for each course.

3.4.3 Feature engineering

We used two different presentations of the data for both datasets to enable comprehensive model comparison:

a. 1D-Model: After grouping the data, we created a pivot table as the final target format. Each student ID was set as the row index, resulting in 7,762 rows (7,092 from OULAD and 6,822 from the MOOC dataset). The columns represent dates (by week), and the values are the total number of clicks for all activities during each week. We also included the final_result (pass/fail) for each student. This transformed the interaction data into a one-dimensional time series format, as shown in Figure 3.4.

Week	week0	week2	....	week40	outcomes
student 1	169	233		304	0 (fail)
student 2	153	214		362	1 (pass)
student 3	173	270		381	1
....					....
student 7092	82	122		292	0

Figure 3.4: One-Dimensional (1D) Model Representation

b. 2D-Model: After grouping the data, each group present a table for one student 7,092 tables for the OULAD dataset and 6,822 for the MOOC dataset. These tables were created using a pivot, where dates (41 weeks for OULAD, 37 days for MOOC) were set as row indices, and activity types were used as columns. For OULAD, we had 12 activity types, resulting in 12 columns. For the MOOC dataset, we had 20 activity types. The values in the table are the sum of clicks the total number of interactions per activity type, per time unit (week for OULAD, day for MOOC). This transformation created a time-series like structure for each student, as is illustrated in Figure 3.5

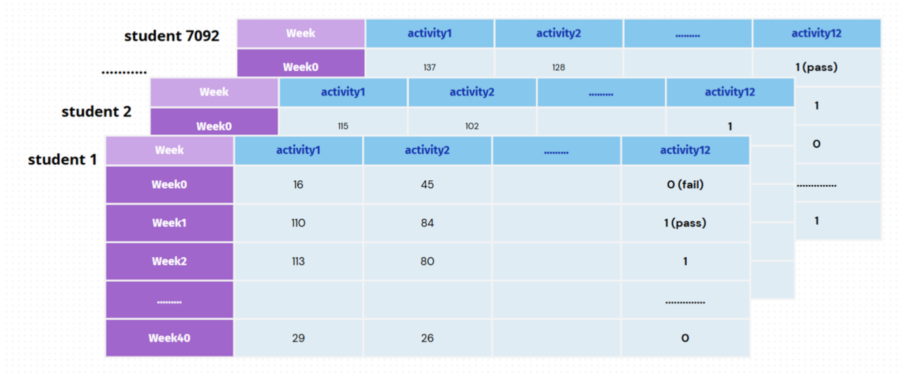

Figure 3.5: Two-Dimensional (2D) Model Representation

3.4.4 Handling missing values

After analyzing the OULAD data, we identified missing values, particularly for students with no recorded VLE interactions. A total of 670 students had no interaction data and were removed, reducing the dataset to 7,092 students. For the MOOCDataset, similar checks were performed, and wo don't found any students with no interaction data so the number of student as the same 6822,This approach ensured compatibility with the 1D and 2D model representations for robust analysis across both datasets.

3.4.5 Label Merging and Class Simplification

To simplify the classification task and align with the binary prediction objective (pass/fail), the OULAD Dataset original four outcome labels (Pass, Fail, Withdrawn, Distinction) were merged into two classes: "Withdrawn" was merged with "Fail," and "Distinction" was merged with "Pass." This resulted in a balanced dataset with 3,648 students labeled as "Pass" (51.44%) and 3,444 labeled as "Fail" (48.56%) after doing the preprocessing. For the MOOC Dataset, the truth labels were already binary (Pass as 1 and Fail as 0), requiring no further merging. This step reduced model complexity while maintaining dataset relevance for predicting student performance and identifying at-risk students across both datasets. To further address potential class imbalance introduced through this binary simplification, we also computed and applied class weights during model training. This step helped reduce bias toward the majority class and improved generalization across both datasets.

3.4.6 Imballanced data handling

We addressed potential class imbalance in the target variable during preprocessing. Since the distribution of pass/fail outcomes was not perfectly balanced , we computed class weights to be used during model training. These weights were calculated using the compute_class_weight function from scikit-learn with the 'balanced' mode, which assigns higher weights to underrepresented classes based on their frequency in the training set. The resulting class weights were then passed to the loss function during training to ensure the model did not become biased toward the majority class. This step was critical in improving generalization and ensuring fair evaluation of student performance prediction across both datasets.

3.4.7 Feature scaling

To ensure consistent feature scales and improve model performance, both standardization and normalization techniques were applied to the input data [15].

a. Standardization: This transformation adjusts the data to have a mean of zero and a standard deviation of one, where x is the original feature value, μ is the mean, and σ is the standard deviation of the feature across all samples. according to the formula:

$$z = \frac{x - \mu}{\sigma} \tag{3.1}$$

b. Normalization: This transformation was used to rescale the values into the range [0,1], where x is the original value, x_{min} and x_{max} and are the minimum and maximum values of the feature, respectively. as defined by:

$$x' = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \tag{3.2}$$

3.5 Evaluation metrics

- **Accuracy:** is a fundamental metric for evaluating the performance of a classification model, it's show the effectiveness of a model in classification tasks it's measures the proportion of correctly classified instances (positive and negative) among the all instances.

  $$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.3}$$

  The accuracy working great when each class has the same number of instances, but in case of imbalanced data can be misleading.

- **Precision:** measures the proportion of true positive predictions instance among all positive predictions instance made by the model.

  $$\text{Precision} = \frac{TP}{TP + FP} \tag{3.4}$$

- **Recall:** or sensitivity measures the proportion of true positive predictions instance among all actual positive instance:

  $$\text{Recall} = \frac{TP}{TP + FN} \tag{3.5}$$

  Where:
  TP (True Positive): Model correctly predicts the positive class.
  TN (True Negative): Model correctly predicts the negative class.
  FP (False Positive): Model predicts positive, but it's actually negative.
  FN (False Negative): Model predicts negative, but it's actually positive.

- **F1 score:** is the harmonic mean of precision and recall. It combines them into a single metric that balances the two. It shows the trade-off between precision and recall. It's especially useful when the class distribution is imbalanced and you want a balance between false positives and false negatives.

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.6}$$

- **AUC-ROC:** is an essential tool used for evaluating the performance of binary classification models. helps us understand how well a classification model distinguishes between the two classes (positive and negative). ROC-curve: plots TPR (true Positive rate or Recall) vs. FPR (false positive rate)

$$\text{FPR} = \frac{FP}{FP + TN} \tag{3.7}$$

  AUC measures the entire tow-dimensional area underneath the ROC curve ranging from 0 to 1. High AUC (close to 1) The model effectively distinguishes between positive and negative instances. Low AUC (close to 0) The model struggles to differentiate between the two classes. AUC around 0.5 The model doesn't learn any meaningful patterns i.e. it is doing random guessing. AUC-ROC is effective when the dataset is balanced and the model needs to be evaluated across all thresholds, False positives and false negatives are of similar importance.

3.6 Experiments

3.6.1 Model selection

The models selected for this study were chosen to address two key goals: capturing the time-based nature of student activity in Virtual Learning Environments (VLEs) and comparing deep learning approaches with traditional machine learning methods.

Recurrent models such as RNN, LSTM, and GRU were chosen over deep learning approaches due to their superior ability to model sequential data. while feedforward networks like ANN or spatially-focused models like CNN are not designed for time-dependent inputs, RNN, LSTM and GRU are built specifically to handle time-series data. They maintain memory of past events, allowing them to learn patterns over time, which is essential for accurately predicting student performance based on evolving behaviors.

Decision Tree and Random Forest represent traditional, interpretable methods. These models help identify which features are most important for predicting outcomes. Random Forest, in particular, helps reduce overfitting and handles high-dimensional data better than single decision trees. K-Nearest Neighbors (KNN) and Logistic Regression were included for their simplicity and to provide additional benchmarks and comparison.

3.6.2 Grid search

Grid Search is an exhaustive search strategy in which a machine learning algorithm is trained and evaluated using all possible combinations of hyperparameter values. The goal is to find the hyperparameter configuration that optimizes a chosen performance metric (e.g., accuracy, F1-score, RMSE) on a validation set or through cross-validation. For example, hyperparameters for the SVM algorithm (c, gamma, kernel), or for random forest (n_estimators, max_depth, min_samples_split, min_samples_leaf).

3.6.3 Cross validation

Cross-validation is a statistical method used to estimate the performance (generalization ability) of machine learning models.it involves splitting the dataset into multiple subsets (called folds), training the model on some folds, and testing it on the remaining fold(s). This process is repeated several times so that each data point gets to be in the test set once. The final performance is usually the average of all test performances. [1]

3.7 Conclusion

This chapter has outlined the comprehensive experimental methodology employed to preprocess and manage data for training predictive models. The research design integrates robust data preprocessing, feature engineering, and data structuring to ensure consistency and generalizability across both datasets useing same preprocessing structure to both OULAD and TsinghuaX MOOC dataset. By applying a unified preprocessing , handling missing values, simplifying class labels, and addressing class imbalance, the methodology establishes a solid foundation for training traditional machine learning and deep learning models. the next chapter presents the implementation of these models, detailing the architecture and configuration of each algorithm. It also evaluate and compares model performance across datasets.

CHAPTER 4

IMPLEMENTATION AND EVALUATION

4.1 Introduction

This chapter presents the implementation details of the proposed models, including the development environment, training procedure, evaluation metrics, and performance comparison; for predicting student performance based on interaction data within a Virtual Learning Environment (VLE). It starts by describing the development environment and tools used, including Python, Anaconda, TensorFlow, and other useful libraries. Then, a detailed description of the models' architectures and hyperparameters is provided, covering sequentiel model (LSTM, GRU, RNN) and traditional machine learning methods (Random Forest, Decision Tree, Logistic Regression, and KNN). Two models are analyzed (Model 1 and Model 2), were trained and tested on two different datasets. Furthermore, The chapter also includes a section to early prediction, evaluating how well the models perform at different stages of the course timeline. This work compares the effectiveness of different methods and determines the most suitable techniques for early identification of at-risk students.

4.2 Development environment

The development and implementation of the proposed models were carried out using a combination of a personal computer and a cloud-based environment to ensure flexibility, efficiency, and sufficient computational power for training deep learning models.

Initially, data preprocessing, feature engineering, and preliminary experiments were conducted on a personal computer equipped with an Intel Core i5 processor, 16 GB of RAM, and a 512 GB SSD running Windows 10. This local setup provided an effective environment for dataset preparation, exploratory data analysis, and initial testing of machine learning algorithms.

For the training and evaluation of deep learning models, the Google Colab environment was employed. Google Colab offers a free cloud-based platform that integrates GPU acceleration, making it well suited for computationally intensive tasks such as training Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) models.

- **Anaconda:** Anaconda is a free and open-source software distribution that simplifies package management and deployment for scientific computing, data science, and machine learning. It primarily supports the Python and R programming languages and is designed to make it easier for users, especially researchers, analysts, and developers to work with large-scale data processing, statistical modeling, and complex computational tasks. Anaconda includes over of 250 popular data science and machine learning packages right out of the box. This eliminates the need to install each package separately and ensures compatibility across tools and libraries. Additionally, it includes tools such as Jupyter Notebook, Spyder, and RStudio, which provide interactive environments for writing and testing code [40].

- **Python:** is a high-level, interpreted programming language that emphasizes code readability, simplicity, and flexibility. Created by Guido van Rossum and first released in 1991, Python has become one of the most popular programming languages in the world due to its easy-to-understand syntax and powerful features. Python is widely used across various domains such as web development, data science, machine learning [41].

- **TensorFlow:** is an open-source machine learning library developed by Google that provides tools to build, train, and deploy machine learning and deep learning models. It supports computations on CPUs, GPUs, and TPUs and is widely used for tasks like image recognition, natural language processing, and neural network training [42].

- **Numpy:** (Numerical Python) is an open-source Python library used for numerical computing. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on them efficiently. It's a foundational library for scientific computing in Python [43].

- **Pandas:** is an open-source Python library used for data manipulation and analysis. It provides powerful data structures like DataFrame and Series to handle structured data easily, allowing for data cleaning, transformation, aggregation, and visualization. It is commonly used in data science and machine learning workflows [44].

- **Scikit-learn:** (or sklearn) is an open-source Python library for machine learning. It provides simple and efficient tools for data mining and data analysis, including algorithms for classification, regression, clustering, dimensionality reduction, model selection, and preprocessing. It is built on top of NumPy, SciPy, and matplotlib [45].

- **Matplotlib:** is an open-source Python library used for creating static, interactive, and animated visualizations. It is widely used for plotting graphs, such as line charts, bar charts, scatter plots, histograms, and more, making it a fundamental tool for data visualization in Python [46].

4.3 Model building

This section provides a comprehensive overview of the parameters and architecture of the models developed using the Grid Search technique.

4.3.1 Model 1

This section presents the parameters and architecture of the one-dimensional models.

a. Long Short-Term Memory (LSTM)

The table 4.1 presents the detailed architecture of the LSTM model. The learning rate was set to 0.0001 and optimized using the Adam optimizer. The number of epochs was set to 50, with a batch size of 64. An L2 regularization technique was applied with a value of 0.001. All these hyperparameters were fine-tuned using the Grid Search optimization method.

	Layer Type	Num neurons	activation fun	add info
1	input layer			
2	LSTM	16	tanh	
3	Dropout			value=0.3
4	LSTM	8	tanh	
4	Dropout			value=0.3
6	Dense	8	sigmoid	
7	Dropout			value=0.3
8	output layer		sigmoid	

Table 4.1: LSTM architecture

b. Gated Recurrent Unit:

The table 4.2 presents the detailed architecture of the GRU model. The learning rate was set to 0.0001 and optimized using the Adam optimizer. The number of epochs was set to 50, with a batch size of 64. An L2 regularization technique was applied with a value of 0.01. All these hyperparameters were fine-tuned using the Grid Search optimization method.

	Layer Type	Num neurons	activation fun	add info
1	input layer			
2	GRU	16	tanh	
3	Dropout			value=0.3
4	GRU	8	tanh	
4	Dropout			value=0.3
6	Dense	8	sigmoid	
7	Dropout			value=0.3
8	output layer		sigmoid	

Table 4.2: GRU architecture

c. Recurrent Neural Network

The table 4.3 presents the detailed architecture of the RNN model. The learning rate was set to 0.0001 and optimized using the Adam optimizer. The number of epochs was set to 50, with a batch size of 64. An L2 regularization technique was applied with a value of 0.01. All these hyperparameters were fine-tuned using the Grid Search optimization method.

	Layer Type	Num neurons	activation fun	add info
1	input layer			
2	RNN	16	tanh	
3	Dropout			value=0.3
4	RNN	8	tanh	
4	Dropout			value=0.3
6	Dense	8	sigmoid	
7	output layer		sigmoid	

Table 4.3: RNN architecture

d. Random Forest

The table 4.4 presents the parameters trained by the models for both Datasets. These parameters were selected through the Grid Search optimization technique.

	OULAD	MOOC
Max_depth	5	3
n_estimators	50	5
Min_samples_leaf	1	1
Min_samples_split	2	2
Max_leaf_nodes	NONE	3

Table 4.4: Random Forest parameters

e. Decision Tree

For decision tree, the OULAD dataset models used a max_depth of 3 and The criterion was set to entropy while TsinghuaX dataset, max_depth was set to 7 and criterion to gini to guide the split decisions. These parameters were selected using grid search.

f. Logistic Regression

For linear regression models, the solver was set to 'liblinear' for both datasets. These solvers were chosen using grid search to ensure optimal convergence and performance for each dataset.

g. k-Nearest Neighbors (KNN)

The table 4.5 represent the parameter tuning using grid search.

	OULAD	TsinghuaX
$N_neighbors$	9	3
metrics	Euclidean	Manhanttan
weights	uniform	distance

Table 4.5: KNN parameters

4.3.2 Model 2

This section presents the parameters and architecture of the two-dimensional models.

a. Long Short-Term Memory (LSTM)

The table 4.6 presents the detailed architecture of the LSTM model. The learning rate was set to 0.0001 and optimized using the Adam optimizer. The number of epochs was set to 50, with a batch size of 64. An L2 regularization technique was applied with a value of 0.001. All these hyperparameters were fine-tuned using the Grid Search optimization method.

	Layer Type	Num neurons	activation fun	add info
1	input layer			
2	LSTM	32	tanh	
3	Dropout			value=0.3
4	LSTM	16	tanh	
4	Dropout			value=0.3
6	Dense	8	sigmoid	
7	Dropout			value=0.3
8	output layer		sigmoid	

Table 4.6: LSTM architecture

b. Gated Recurrent Unit (GRU)

The table 4.7 and the Figure 4.1 presents the detailed architecture of the GRU model. The learning rate was set to 0.0001 and optimized using the Adam optimizer. The number of epochs was set to 50, with a batch size of 64. An L2 regularization technique was applied with a value of 0.01. All these hyperparameters were fine-tuned using the Grid Search optimization method. While the TsinghuaX dataset has the same layers with 16 neurons for the first GRU layer and 8 neuron for the second GRU layer.

	Layer Type	Num neurons	activation fun	add info
1	input layer			
2	GRU	32	tanh	
3	Dropout			value=0.3
4	GRU	16	tanh	
4	Dropout			value=0.3
6	Dense	8	sigmoid	
7	Dropout			value=0.3
8	output layer		sigmoid	

Table 4.7: GRU architecture

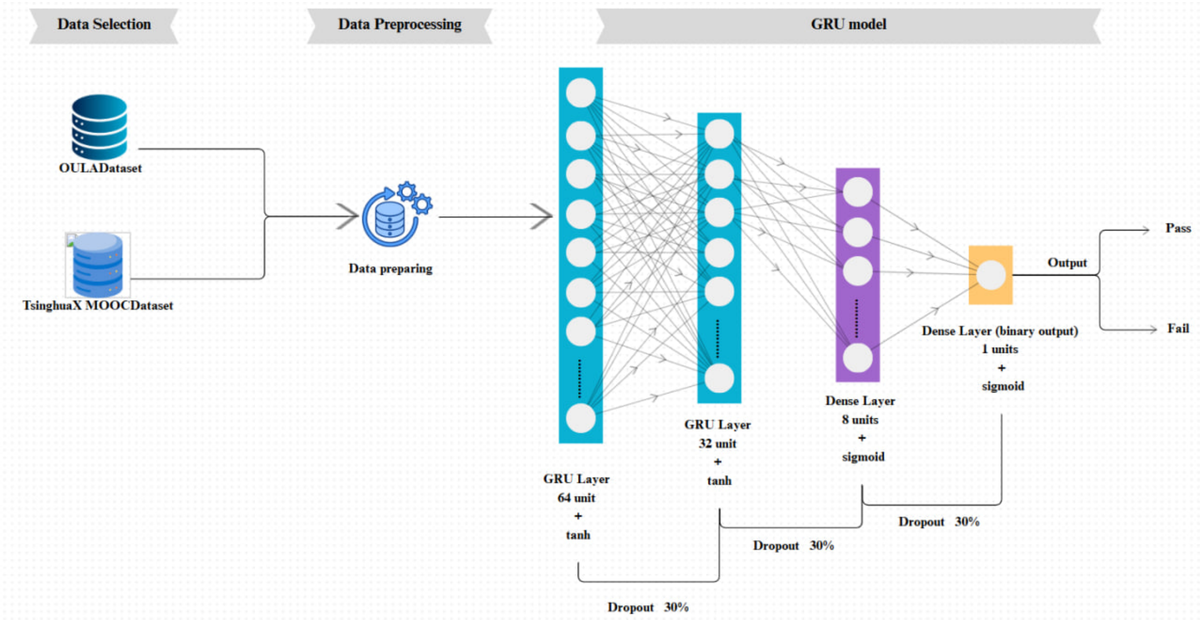

Figure 4.1: Architecture of GRU model

c. Recurrent Neural Network

The table 4.8 presents the detailed architecture of the RNN model. The learning rate was set to 0.0001 and optimized using the Adam optimizer. The number of epochs was set to 50, with a batch size of 64. An L2 regularization technique was applied with a value of 0.01. All these hyperparameters were fine-tuned using the Grid Search optimization method.

	Layer Type	Num neurons	activation fun	add info
1	input layer			
2	RNN	16	tanh	
3	Dropout			value=0.3
4	RNN	8	tanh	
4	Dropout			value=0.3
6	Dense	8	sigmoid	
7	Dropout			value=0.3
8	output layer		sigmoid	

Table 4.8: RNN architecture

d. Random Forest

The table 4.9 presents the parameters trained by the models for both Datasets. These parameters were selected through the Grid Search optimization technique.

	OULAD	TsinghuaX
Max_depth	5	3
n_estimators	50	5
Min_samples_leaf	1	1
Min_samples_split	2	2
Max_leaf_nodes	NONE	3

Table 4.9: Random Forest parameters

e. Decision Tree

For both datasets, the decision tree models used a max_depth of 5. The criterion was set to 'entropy' for OULAD and 'gini' for TsinghuaX dataset to guide the split decisions. These parameters were selected using grid search.

f. Logistic Regression

For linear regression models, the solver was set to 'liblinear' for OULAD dataset and 'lbfgs' for TsinghuaX dataset. These solvers were chosen using grid search to ensure optimal convergence and performance for each dataset.

g. k-Nearest Neighbors (KNN)

The table 4.10 represent the parameter tuning using grid search.

	OULAD	TsinghuaX
N_neighbors	9	3
metrics	Euclidean	Manhanttan
weights	uniform	uniform

Table 4.10: KNN parameters

4.3.3 Early prediction

In the early prediction models of the experimental evaluation, the model architecture and hyperparameter configurations were maintained in alignment with those defined for TsinghuaX dataset under Model 1. This consistency was applied across all selected methodologies, including Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Random Forest classifiers.

4.4 Model Evaluation

4.4.1 Model 1

Table 4.11 and 4.12 present the complete result of Model 1.

a. Long Short-Term Memory (LSTM)

OULAD dataset: The LSTM model delivers strong performance with 93.13% accuracy, high recall (98.96%), and solid precision (88.97%). The F1 score (93.69%) reflects a well-balanced detection of passing students. The AUC-ROC of 96.67% confirms excellent class separation.

TsinghuaX dataset: While slightly weaker, performance remains robust with 85.36% accuracy, precision (94.76%), and recall (87.82%), resulting in an F1 score of 91.16%. AUC-ROC (86.25%) confirms decent reliability.

b. Gated Recurrent Unit (GRU)

OULAD Dataset: GRU slightly outperforms LSTM with 93.15% accuracy. It balances recall (99.05%) and precision (88.93%) to produce an excellent F1 score of 93.71%. AUC-ROC (96.61%) reinforces strong class separation.

TsinghuaX dataset: Solid metrics with 85.84% accuracy, precision (94.64%), and recall (88.55%), resulting in a balanced F1 score (91.48%). AUC-ROC (86.11%) reflects reliable separation despite some trade-off.

c. Recurrent Neural Network (RNN)

OULAD Dataset: RNN posts a strong showing with 93.09% accuracy, high recall (98.66%), and precision (89.11%), yielding an F1 score of 93.63%. The AUC-ROC of 95.96% confirms its effectiveness.

TsinghuaX dataset: the model achieve acceptable performance with Accuracy at 84.98%, with precision (94.71%) and recall (87.41%) figure 4.3. F1 score (90.91%) shows solid performance, As illustrated in figure 4.2, both accuracy and loss curves show stable convergence. AUC-ROC (84.4%) is acceptable but weaker than OULAD Dataset.

d. Random Forest (RF)

OULAD Dataset: RF performs well with 92.31% accuracy. Recall is high (98.91%), but precision drops slightly (87.74%), leading to a strong F1 score (92.99%). AUC-ROC (96.53%) confirms effective classification.

TsinghuaX dataset: Performance drops with 82.04% accuracy, and F1 score (88.97%). Lower AUC-ROC (78.87%) indicates weaker class separation.

e. Decision Tree (DT)

OULAD Dataset: DT achieves 91.65% accuracy, with good balance: precision (88.07%), recall (96.92%), and F1 score (92.27%). AUC-ROC of 95.24% confirms strong performance.

TsinghuaX dataset: Slight dip with 83.26% accuracy, but strong precision (93.42%) and recall (86.63%) lead to a good F1 score (89.9%). AUC-ROC (73.67%) points to weaker class discrimination.

f. K-Nearest Neighbors (KNN)

OULAD Dataset: KNN underperforms here. Accuracy (89.62%) is acceptable, but recall (90.74%) is lowest among top models. F1 score (89.99%) is decent, yet AUC-ROC

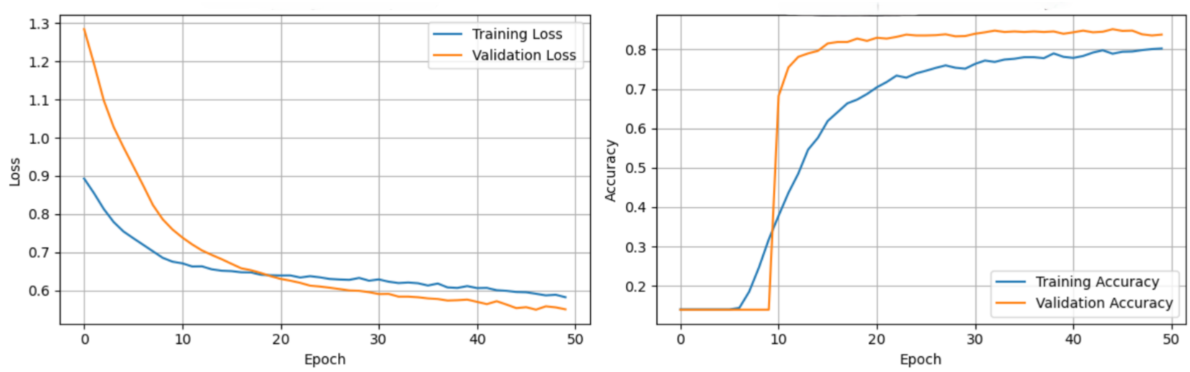

Figure 4.2: GRU Model accuracy and loss

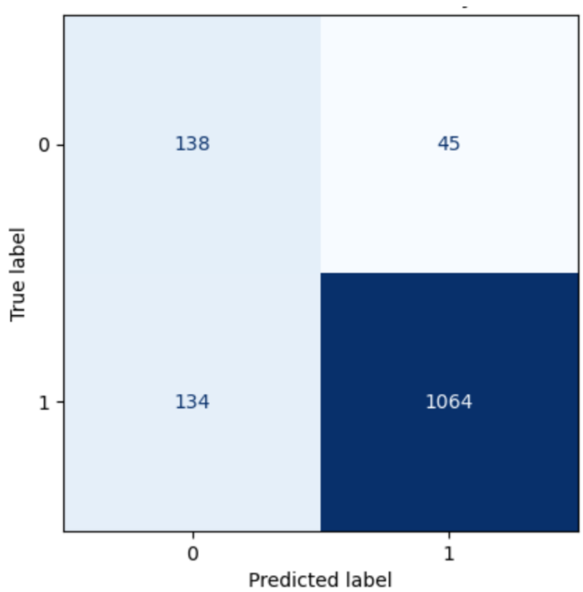

Figure 4.3: GRU Model confusion matrix

(94.93%) is weaker compared to others.

TsinghuaX dataset: Performs well on recall (93.26%) and maintains accuracy (83.89%), precision (88.6%), and F1 score (90.87%). However, AUC-ROC plummets to 59.86%, highlighting poor boundary definition.

g. Logistic Regression (LR)

OULAD Dataset: LR shows consistent results with 90.26% accuracy, balanced precision (90.07%) and recall (91.16%), and F1 score (90.59%). AUC-ROC (95.5%) confirms strong, stable performance.

TsinghuaX dataset: Results remain good: accuracy (84.07%), precision (91.61%), recall (89.68%), and F1 score (90.64%). AUC-ROC (71.08%) is on the lower end, suggesting limitations in distinguishing edge cases.

h. Discussion

The results clearly show that deep learning models designed for sequences (like LSTM, GRU, and RNN) perform better than traditional machine learning methods on both datasets. On OULAD Dataset, these models reach high accuracy (about 93%) and strong F1 scores (around 93.6–93.7%). This shows that they're better at understanding the time-based patterns in how students interact each week.

In comparison, traditional models like Random Forest, Decision Tree, Logistic Regression, and KNN don't perform as well. For example, Random Forest hits 92.31% accuracy and high recall (98.91%) on OULAD Dataset, but its lower precision (87.74%) shows it makes more false positives. KNN and Logistic Regression do okay, but they can't handle time-based patterns well—especially visible in KNN's weaker AUC-ROC on TsinghuaX dataset.

All models do worse on TsinghuaX dataset, especially in terms of AUC-ROC. For example, LSTM drops from 96.67% to 86.25%, and Random Forest from 96.53% to 78.87%. This suggests the TsinghuaX dataset is messier or less organized.

Model	Accuracy	Precision	Recall	F1 Score	AUC-ROC
LSTM	0.9313	0.8897	0.9896	0.9369	0.9667
GRU	0.9315	0.8893	0.9905	0.9371	0.9661
RNN	0.9309	0.8911	0.9866	0.9363	0.9596
RF	0.9231	0.8774	0.9891	0.9299	0.9653
DT	0.9165	0.8807	0.9692	0.9227	0.9524
KNN	0.8962	0.8928	0.9074	0.8999	0.9493
LR	0.9026	0.9007	0.9116	0.9059	0.9550

Table 4.11: Model 1 performance on OULAD Dataset

Model	Accuracy	Precision	Recall	F1 Score	AUC-ROC
LSTM	0.8536	0.9476	0.8782	0.9116	0.8625
GRU	0.8584	0.9464	0.8855	0.9148	0.8611
RNN	0.8498	0.9471	0.8741	0.9091	0.8440
RF	0.8204	0.9424	0.8427	0.8897	0.7887
DT	0.8326	0.9342	0.8663	0.8990	0.7367
KNN	0.8389	0.8860	0.9326	0.9087	0.5986
LR	0.8407	0.9161	0.8968	0.9064	0.7108

Table 4.12: Model 1 performance on TsinghuaX dataset

4.4.2 Model 2

Table 4.13 and 4.14 present the complete result of Model 2.

a. Long Short-Term Memory (LSTM)

OULAD Dataset: The model achieve high accuracy with 92.29%, and precision (89.23%), recall (96.71%), indicate the model is highly effective at correctly identifying passing students, while f1 score indicate strong balance between precision and recall.

TsinghuaX dataset: the model has good performance with an accuracy 85.43%, The balanced and high F1 score (91.2%) confirms overall reliability.

b. Gated Recurrent Unit (GRU)

OULAD Dataset: The GRU model demonstrates excellent performance with an accuracy of 93.32%, As illustrated in figure 4.4, both accuracy and loss curves show stable convergence, The F1 score of 93.78% balances recall 97.84% and precision 90.04% metrics well, showing strong reliability in both detecting and correctly classifying the majority class figure 4.5.

TsinghuaX dataset: the model record good result with accuracy 86.11%, despite the difference value between recall 88.56% and precision 94.95%, The F1 score of 91.64% reflects this trade-off between precision and recall.

c. Recurrent Neural Network

OULAD Dataset: The RNN performs strongly with accuracy 93.09%, also high recall 97.57% and precision 89.88%, leading to a strong F1 score of 93.57%. The AUC-ROC of 0.9653 confirms excellent class separation.

TsinghuaX dataset: The model realize good performance with 85.59% of accuracy, precision 94.7%, and recall 88.19%. The F1 score of 91.32% shows a decent balance.

d. Random Forest

OULAD Dataset: The model perform very well with accuracy of 93.08%, and an exceptionally high recall of 98.97%, Precision is slightly lower at 88.85%, The F1 score of 93.64% confirms strong overall performance.

TsinghuaX dataset: the model obtain a good results with 84.45% of accuracy and f1 score 90.74%. AUC-ROC with 74.97% shows weaker class distinction.

e. Decision Tree

OULAD Dataset: model performs well on OULAD dataset, with high accuracy (91.54%) and a balanced F1 score (91.93%), indicating solid overall performance. Precision (90.32%) and recall (93.63%) are both strong.

TsinghuaX dataset: the model achieve acceptable results, Accuracy is 85.25%, precision (92.65%), recall (89.98%), a decent F1 score (91.29%). and slow AUC-ROC 72.08% pointing to weaker class separation.

f. Logistic regression

OULAD Dataset: Logistic regression performs strongly on OULAD Dataset, with high accuracy (91.86%) and well-balanced precision (92.03%) and recall (92.19%). The F1 score (92.1%) confirms consistent performance across both classes.

TsinghuaX dataset: the model achieve good results with Accuracy is 83.4%, precision (92.21%), recall (88.14%), a decent F1 score (91.29%). and low AUC-ROC 70.31% indicating limited ability to distinguish pass/fail cases.

g. k-Nearest Neighbors (KNN)

OULAD Dataset: The model shows weak performance with low accuracy (77.86%) and a poor recall of 62.93%. Despite a high precision (91.35%) and decent AUC-ROC (91%), the low recall drags down the F1 score to 74.5%.

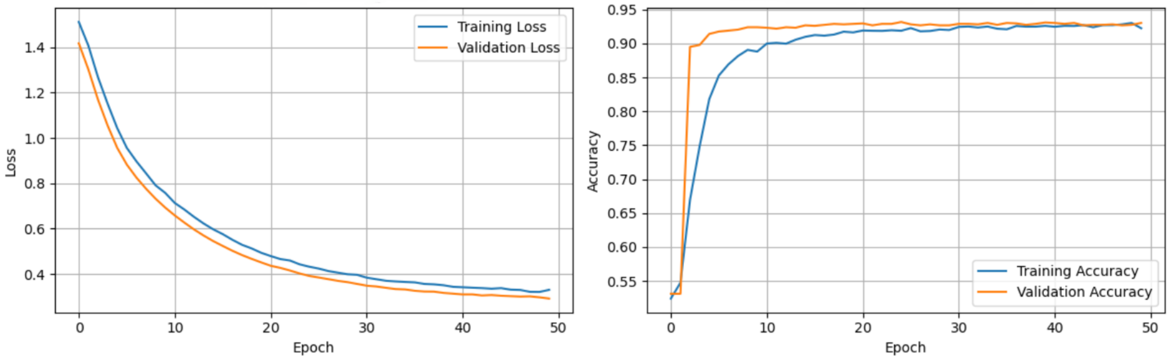

Figure 4.4: GRU Model accuracy and loss

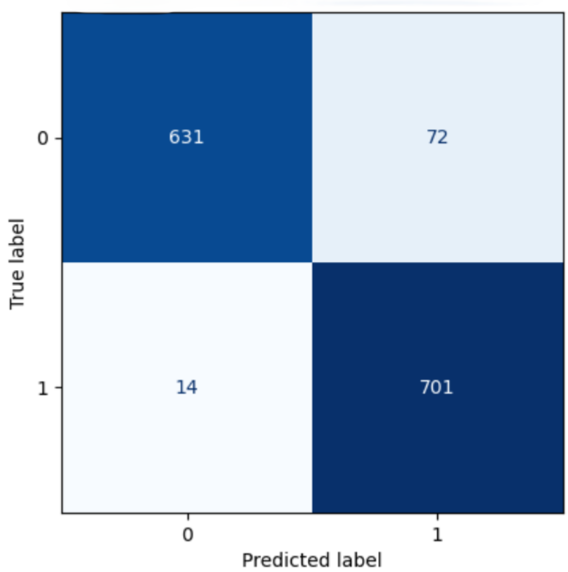

Figure 4.5: GRU Model confusion matrix

TsinghuaX dataset: the model performs better in terms of recall (93.79%) and F1 score (90.32%), indicating improved balance. Accuracy (82.71%) and precision (87.1%) are also solid. However, the AUC-ROC is very low (54.31%), pointing to poor class separation.

h. Discussion

The results clearly show that sequential models such as LSTM, GRU, and RNN perform better than traditional machine learning methods on both datasets. On OULAD Dataset, these models reached high precision (92–93%) and strong F1 scores (over 93%), proving they're good at handling student activity data over time. They can pick up patterns in weekly clicks that regular models miss. LSTM and GRU also had a very high recall (96.71% and 97.84%), which means they're great at correctly identifying students who are likely to succeed and are important for providing early support.

Traditional models like Random Forest and Logistic Regression did okay, but had clear limits. Random Forest had solid accuracy (93.08%) on OULAD Dataset, mostly because of very high recall (98.97%), but it came with lower precision. Logistic Regression and

Decision Tree were steady but not as good at spotting complex patterns. KNN didn't do well on OULAD Dataset it had low recall and a weak F1 score, showing it struggles with time-based and unbalanced data.

All models performed worse on TsinghuaX dataset, with accuracy dropping by 5–10%. This suggests this Dataset may be messier or have less reliable patterns, maybe due to inconsistent student activity or poor data recording. Still, LSTM, GRU, and RNN held up well, with F1 scores around 91%, showing they're resilient even with lower-quality data. Traditional models dropped off more, especially in AUC-ROC scores, which means they couldn't separate successful and unsuccessful students as well.

In short, deep learning models that handle time-based data are better for predicting student behavior. They're more accurate and work better overall, especially when the data has clear patterns. The lower performance on TsinghuaX dataset also shows how important it is to have clean, consistent data in educational machine learning projects.

Model	Accuracy	Precision	Recall	F1 Score	AUC-ROC
LSTM	0.9229	0.8923	0.9671	0.9281	0.9630
GRU	0.9332	0.9004	0.9784	0.9378	0.9657
RNN	0.9309	0.8988	0.9757	0.9357	0.9653
RF	0.9308	0.8885	0.9897	0.9364	0.9689
DT	0.9154	0.9032	0.9363	0.9193	0.9499
LR	0.9186	0.9203	0.9219	0.9210	0.9668
KNN	0.7786	0.9135	0.6293	0.7450	0.9100

Table 4.13: Model 1 performance on OULAD Dataset

Model	Accuracy	Precision	Recall	F1 Score	AUC-ROC
LSTM	0.8543	0.9478	0.8791	0.9120	0.8674
GRU	0.8611	0.9495	0.8856	0.9164	0.8595
RNN	0.8559	0.9470	0.8819	0.9132	0.8535
RF	0.8445	0.9296	0.8863	0.9074	0.7497
DT	0.8525	0.9265	0.8998	0.9129	0.7208
LR	0.8340	0.9221	0.8814	0.9012	0.7031
KNN	0.8271	0.8710	0.9379	0.9032	0.5431

Table 4.14: Model 1 performance on TsinghuaX dataset

4.4.3 Early prediction model

a. Random Forest

Random Forest, a non-sequential ensemble method, demonstrates surprisingly strong performance in the early weeks. With an F1 score of 0.7209 and accuracy of 0.7005 in Week 1 (table 4.15), it shows that early click behavior even without modeling temporal dependencies is informative. The model continues to improve steadily as more weekly data becomes available, reaching an F1 score of 0.9376 by Week 10. However, because Random Forest treats each input as static, its ability to capture progression or behavioral trends over time is inherently limited. Its performance begins to plateau around Week 7, implying diminishing returns from just adding more time-sliced click data without modeling the sequential nature of the input figure 4.6.

course stage	**Accuracy**	**F1 Score**	**Precision**	**Recall**	**ROC-AUC**
10%	0.7005	0.7209	0.6923	0.7521	0.6990
20%	0.7470	0.7691	0.7248	0.8192	0.7449
30%	0.7738	0.7873	0.7625	0.8137	0.7726
40%	0.8154	0.8296	0.7896	0.8740	0.8136
50%	0.8682	0.8797	0.8291	0.9370	0.8662
60%	0.8901	0.8987	0.8543	0.9479	0.8883
70%	0.9056	0.9125	0.8716	0.9575	0.9040
80%	0.9260	0.9309	0.8961	0.9685	0.9247
90%	0.9309	0.9362	0.8921	0.9849	0.9293
100%	0.9323	0.9376	0.8923	0.9877	0.9307

Table 4.15: RF performance at different course stages

b. Long Short-Term Memory (LSTM)

LSTM, a sequence-based deep learning model, starts with a similar results to early-week performance to Random Forest but shows stronger growth as the sequence length increases. Its F1 score rises from 0.7211 in Week 1 to 0.9456 by Week 10 table 4.16 and figure 4.7, outpacing Random Forest after Week 4. This model is better suited for identifying patterns in the progression of student behavior—such as delayed engagement or sudden drops in activity—due to its memory mechanism. The steep improvement between Weeks 3 and 5 suggests that the temporal structure of the data becomes highly predictive during this phase, highlighting the value of capturing not just how much interaction occurred, but when and how it evolved.

course stage	**Accuracy**	**F1 Score**	**Precision**	**Recall**	**ROC-AUC**
10%	0.6970	0.7211	0.6847	0.7616	0.7476
20%	0.7393	0.7473	0.7452	0.7493	0.8167
30%	0.7569	0.7612	0.7692	0.7534	0.8372
40%	0.8323	0.8476	0.7957	0.9068	0.8959
50%	0.8647	0.8743	0.8371	0.9151	0.9254
60%	0.8830	0.8911	0.8552	0.9301	0.9373
70%	0.9006	0.9069	0.8752	0.9411	0.9499
80%	0.9253	0.9295	0.9031	0.9575	0.9603
90%	0.9387	0.9424	0.9117	0.9753	0.9695
100%	0.9422	0.9456	0.9165	0.9767	0.9721

Table 4.16: LSTM performance at different course stage

c. Gated Recurrent Unit (GRU)

GRU, a simpler version of LSTM with fewer parameters, performs strongly and slightly better than LSTM in most weeks. With an F1 score of 0.7088 in Week 1 and 0.9431 by Week 10 table4.17 and figure 4.8, it balances speed and accuracy effectively. Notably, GRU achieves superior ROC-AUC scores in the later weeks, indicating better generalization and class separation in risk prediction. It also shows a sharper rise in precision and recall from Week 3 onward, suggesting that it captures early warning signals in student activity patterns more efficiently than both LSTM and Random Forest.

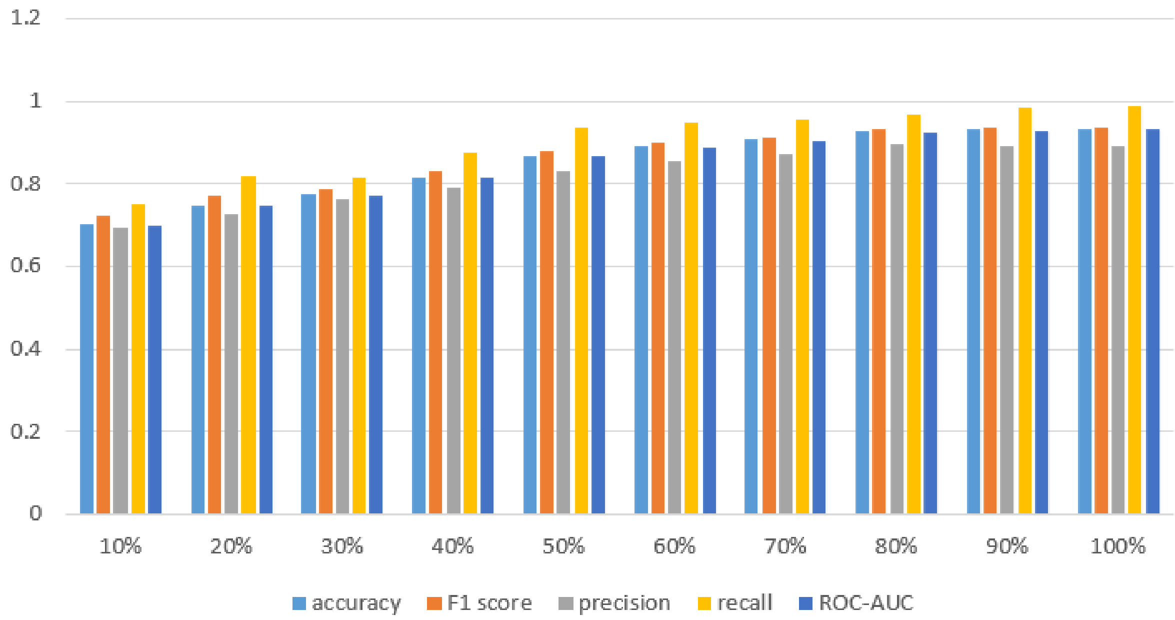

Figure 4.6: Random Forest Graph

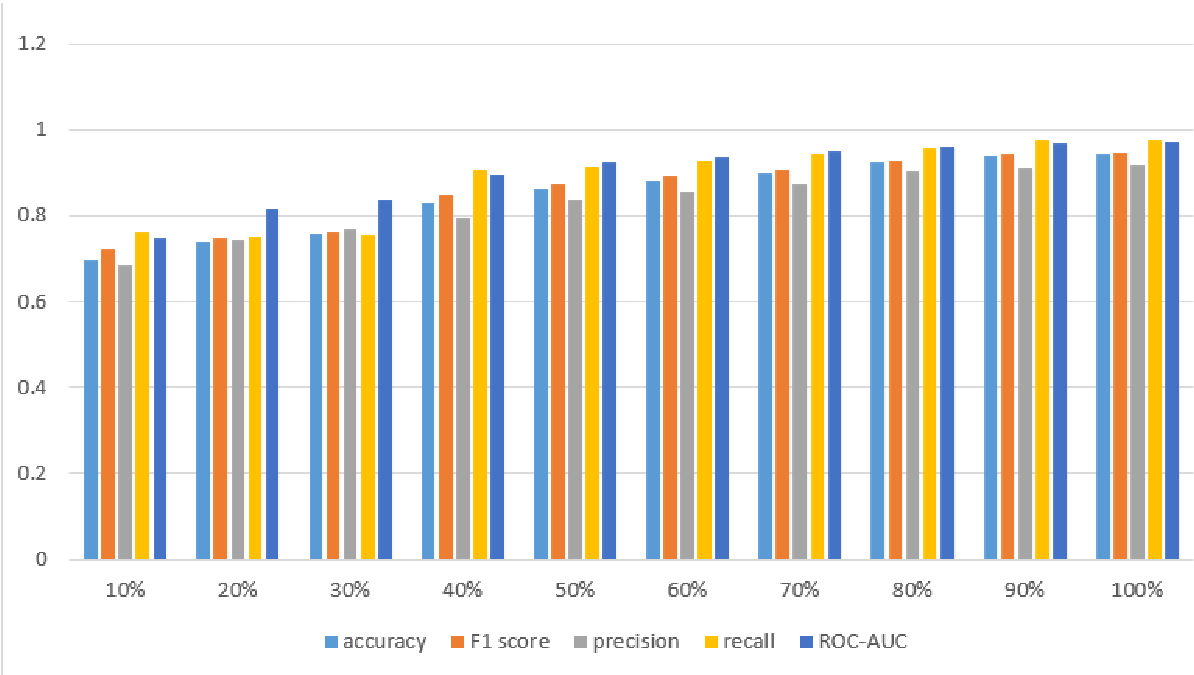

Figure 4.7: LSTM Graph

course Stage	**Accuracy**	**F1 Score**	**Precision**	**Recall**	**ROC-AUC**
10%	0.7012	0.7088	0.7107	0.7068	0.7714
20%	0.7625	0.7836	0.7376	0.8356	0.8298
30%	0.7703	0.7782	0.7730	0.7836	0.8438
40%	0.8365	0.8554	0.7849	0.9397	0.8999
50%	0.8682	0.8783	0.8364	0.9247	0.9295
60%	0.8929	0.9012	0.8577	0.9493	0.9408
70%	0.9063	0.9135	0.8699	0.9616	0.9525
80%	0.9253	0.9288	0.9116	0.9466	0.9653
90%	0.9387	0.9422	0.9148	0.9712	0.9787
100%	0.9401	0.9431	0.9227	0.9644	0.9775

Table 4.17: GRU performance at different course stage

d. Discussion and Summary

In summary, while Random Forest performs adequately in the early weeks and is computationally inexpensive, it lacks the temporal modeling power needed for optimal performance in mid-to-late stages. LSTM and GRU, by contrast, leverage the sequence structure of the data to achieve superior predictive accuracy, with GRU offering the best overall trade-off between model complexity and performance. For early identification of at-risk students—particularly between Weeks 3 and 6—GRU stands out as the most effective approach.

4.5 Conclusion

This chapter divided into three main part developement environment, Model building, Model evaluation. it demonstrated that sequentiel model(LSTM, GRU, RNN) outperform traditional machine learning algorithms in predicting student performance. GRU proved as the most effective model in early prediction tasks, While traditional models such as Random Forest and Logistic Regression showed reasonable performance, they struggled with temporal patterns, These results have provided valuable insights and perceptions for further analysis.

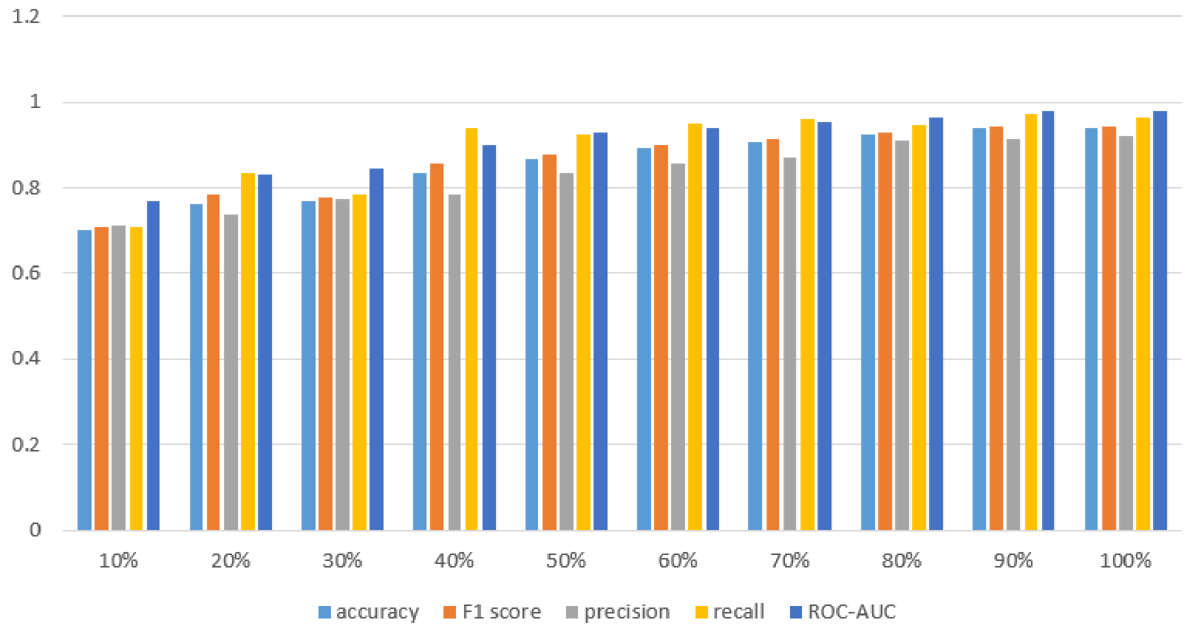

Figure 4.8: GRU Graph

GENERAL CONCLUSION

This study focuses on an important and critical challenge in modern education: the early prediction of at-risk students using artificial intelligence, with a particular focus on temporal modeling of learner behavior over time within Virtual Learning Environments (VLEs). By comparing traditional machine learning models (e.g., Random Forest, KNN, Logistic Regression) with deep learning approaches (e.g., LSTM, GRU, RNN), we demonstrated capability of the temporal models in capturing sequential patterns and improving prediction accuracy. In addition, the study builds early prediction models capable of providing actionable insights at various stages of the course timeline, thereby enabling timely interventions to support students.

Our experiments on the OULAD and TsinghuaX datasets confirmed that time-series-based models significantly outperform static models, especially in early-stage prediction. GRU and LSTM architectures achieved the highest accuracy, proving their strength in processing sequential educational data. Additionally, our preprocessing steps and dual-format data representation (1D and 2D) offered a strong framework for training and evaluating various algorithms under consistent conditions.

While this work has made valuable contributions, several areas still need improvement and future research. Firstly, although temporal models obtain high performance, their interpretability is still limited. Future work should explore explainable AI (XAI) techniques to make the model decisions more transparent and explainable for teachers and stakeholders. Secondly, the datasets used in this study, while valuable, are relatively outdated. It is necessary to extend the analysis to include new and more diverse datasets from different institutions and learning platforms would help evaluate the generalizability of the models across multiple contexts. Third, although we looked at several models, the study could be improved by comparing more advanced time-based models like Informer and TimesNet.

Moreover, integrating additional features such as sentiment analysis from discussion forums, real-time feedback, or social interactions could help the behavioral representation of learners. Finally, developing prediction systems that can be integrated directly into learning platforms would significantly enhance the practical impact of this research, can allow instructors and learning platforms to offer timely, personalized support to at-risk students. For example, an LMS could send automated feedback, recommend supplementary materials, or schedule tutoring sessions for students flagged as at-risk. Moreover, instructors can access dashboards summarizing predictive insights, enabling them to take proactive steps such as modifying teaching strategies or providing personalized support.

In summary, this study has shown that combining deep learning with educational time-series data provides a powerful approach to improving student outcomes. By continuing to refine these models and broaden their application, we can move closer to data-driven, adaptive learning environments that proactively support every learner's success.

BIBLIOGRAPHY

[1] R. K. Doharey, A. Verma, K. Verma, and V. R. Yadav, “Education: Meaning, definition & types,” in Agriculture Extension Education, S.R. Scientific Publication, Jul. 2023, pp. 1–6.

[2] Unity Environmental University, “Why is education important? The power of an educated society,” Unity Environmental University, Nov. 1, 2022. [Online].

[3] R. F. Smith, “Importance of education: Top 10 benefits,” Robert F. Smith, Jun. 27, 2023. [Online].

[4] H. Liu, Z. Wang, P. Benachour, and P. Tubman, “A time series classification method for behaviour-based dropout prediction,” in Proc. 2018 IEEE 18th Int. Conf. Advanced Learning Technologies (ICALT), Mumbai, India, Jul. 2018, pp. 31–35. doi: 10.1109/ICALT.2018.00052.

[5] Y. Liu, S. Fan, S. Xu, A. Sajjanhar, S. Yeom, and Y. Wei, “Predicting student performance using clickstream data and machine learning,” Education Sciences, vol. 13, no. 1, p. 17, Dec. 2022. doi: 10.3390/educsci13010017.

[6] Society for Learning Analytics Research (SoLAR), “What is learning analytics?,” Society for Learning Analytics Research, [Online].

[7] T. G. K. Calders and M. Pechenizkiy, “Introduction to the special section on educational data mining,” SIGKDD Explor. Newsl., vol. 13, no. 2, pp. 3–6, 2012. doi: 10.1145/2207243.2207245.

[8] F. Qiu, G. Zhang, X. Sheng, L. Jiang, L. Zhu, Q. Xiang, B. Jiang, and P.-K. Chen, “Predicting students’ performance in e-learning using learning process and behaviour data,” Scientific Reports, vol. 12, Art. no. 453, Jan. 2022. doi: 10.1038/s41598-021-03867-8.

[9] S. Hilbert et al., “Machine learning for the educational sciences,” Rev. Educ., vol. 9, no. 3, p. e3310, Nov. 2021. doi: 10.1002/rev3.3310.

[10] J. M. Hilbe, Practical Guide to Logistic Regression, 1st ed. Boca Raton, FL, USA: Chapman and Hall/CRC, 2015.

[11] S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Hoboken, NJ, USA: Pearson, 2020.

[12] R. Vishwakarma and J. Kumar, “KNN: K-Nearest Neighbours,” lecture notes, CS456: Machine Learning, National Institute of Science Education and Research, Bhubaneswar, India, Spring 2023.

[13] A. Gao, “Lecture 7: Decision Trees,” CS486/686: Introduction to Artificial Intelligence, University of Toronto, Toronto, ON, Canada, Nov. 2, 2021.

[14] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, May 2015.

[15] S. A. Naser, I. Zaqout, M. A. Ghosh, R. Atallah, and E. Alajrami, “Predicting student performance using artificial neural network,” *Int.J.Hybrid Inf. Technol.*, vol.8, no.2, pp.221–228, 2015. doi:10.14257/ijhit.2015.8.2.20

[16] A. Mondal, “An approach to predict a student’s academic performance using recurrent neural network (RNN),” Int. J. Comput. Appl., no. 6, Jul. 2018.

[17] A. Mikami, "Long Short-Term Memory Recurrent Neural Network Architectures for Generating Music and Japanese Lyrics", Honors thesis, Dept. Comput. Sci., Boston College, Chestnut Hill, MA, USA, 2016.

[18] N. Jha, I. Ghergulescu, and A.-N. Moldovan, “OULAD MOOC dropout and result prediction using ensemble, deep learning and regression techniques,” in Proc. 11th Int. Conf. Comput. Supported Educ. (CSEDU), vol. 2, Heraklion, Greece, May 2019, pp. 154–164. doi: 10.5220/0007767901540164.

[19] H. Chavez, B. Chavez-Arias, S. Contreras-Rosas, J. M. Alvarez-Rodríguez, and C. Raymundo, “Artificial neural network model to predict student performance using nonpersonal information,” Frontiers in Education, vol. 8, Art. no. 1106679, Feb. 2023. doi: 10.3389/feduc.2023.1106679.

[20] Z. Ersozlu, S. Taheri, and I. Koch, “A review of machine learning methods used for educational data,” Education and Information Technologies, vol. 29, pp. 22125–22145, May 2024. doi: 10.1007/s10639-024-12704-0.

[22] C. Wang, L. Chang, and T. Liu, “Predicting student performance in online learning using a highly efficient gradient boosting decision tree,” in Intelligent Information Processing XI, Z. Shi, J.-D. Zucker, and B. An, Eds., IFIP Advances in Information and Communication Technology, vol. 643, Cham, Switzerland: Springer, 2022, pp. 508–521. doi: 10.1007/978-3-031-03948-5_41.

[23] G. Al-Tameemi, J. Xue, S. Ajit, T. Kanakis, I. Hadi, T. Baker, M. Al-Khafajiy, and R. Al-Jumeily, “A deep neural network-based prediction model for students’ academic performance,” in Proc. 14th Int. Conf. Developments in eSystems Engineering (DeSE), Sharjah, UAE, Dec. 2021, pp. 155–160. doi: 10.1109/DeSE54285.2021.9719552.

[24] E. Drousiotis, L. Shi, and S. Maskell, "Early predictor for student success based on behavioural and demographical indicators," in Intelligent Tutoring Systems, vol. 12677, Lecture Notes in Computer Science, Cham, Switzerland: Springer, 2021, pp. 161–172. doi: 10.1007/978-3-030-80421-3_19.

[25] F. Ouatik, M. Erritali, F. Ouatik, and M. Jourhmane, "Predicting student success using big data and machine learning algorithms," Int. J. Emerging Technologies Learn., vol. 17, no. 12, pp. 236–251, Jun. 2022. doi: 10.3991/ijet.v17i12.30259.

[26] Y. Chen et al., "A longitudinal study of student engagement in MOOCs," in *Proc. 13th Int. Conf. Educational Data Mining (EDM)*, 2020, pp. 54–63.

[27] MOOCData.cn, "MOOCData.cn official website." [Online]. Available: `http://moocdata.cn/`

[28] S. Park et al., "EdNet: A large-scale hierarchical dataset in education," *arXiv preprint arXiv:2002.03075*, 2020. [Online]. Available: https://arxiv.org/abs/2002.03075.

[29] TsinghuaX, "Course structure documentation," 2016. [Online]. Available via moocdata.cn dataset description.

[30] MOOCData.cn, "MOOC dataset page." [Online]. Available: `http://moocdata.cn/dataset.html`

[31] M. Adnan, A. Habib, J. Ashraf, S. Mussadiq, A. A. Raza, M. Abid, M. Bashir, and S. U. Khan, "Predicting at-risk students at different percentages of course length for early intervention using machine learning models," IEEE Access, vol. 9, pp. 1–1, Jan. 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3049446

[32] A. Gumaei, M. M. Hassan, A. Alelaiwi, and H. Alsalman, "A hybrid deep learning model for human activity recognition using multimodal body sensing data," IEEE Access, vol. 7, pp. 99152–99160, 2019, doi: 10.1109/ACCESS.2019.2927134.

[33] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in Proc. SSST-8, Doha, Qatar, 2014, pp. 103–111.

[34] R. Dou, X. Wang, and D. Yang, "Time-series classification for predicting dropout in MOOCs using recurrent neural networks," *IEEE Transactions on Learning Technologies*, vol. 13, no. 1, pp. 124–137, 2020. doi: 10.1109/TLT.2019.2905898.

[35] J. Kuzilek, M. Hlosta, and Z. Zdrahal, "Open University Learning Analytics dataset," Scientific Data, vol. 4, Article no. 170171, pp. 1–8, Nov. 2017. doi: 10.1038/sdata.2017.171.

[36] S. A. A. Balabied and H. F. Eid, "Utilizing random forest algorithm for early detection of academic underperformance in open learning environments, PeerJ Computer Science, vol. 9, Art. no. e1708, pp. 1–16, Nov. 2023. doi: 10.7717/peerj-cs.1708.

[37] Iqbal H. Sarker. "Machine Learning: Algorithms, Real-World Applications and Research Directions." In: Sn Computer Science 2.3 (2021), p. 160. issn: 2662-995X. doi: 10.1007/s42979- 021- 00592- x.

[38] H. Waheed, S.-U. Hassan, R. Nawaz, N. R. Aljohani, G. Chen, and D. Gasevic, "Early prediction of learners at risk in self-paced education: A neural network approach," Expert Systems with Applications, vol. 213, Art. no. 118868, Mar. 2023. doi: 10.1016/j.eswa.2022.118868

[39] Nguyen, A., Gardner, L., & Sheridan, D. (2018). A review of learning analytics in higher education: Opportunities, challenges and future directions. Computers & Education, 126, 98-113. https://doi.org/10.1016/j.compedu.2018.07.017

[40] Anaconda, Inc. (n.d.). Anaconda Distribution. Retrieved from https://www.anaconda.com/products/distribution

[41] Python Software Foundation. (n.d.). Welcome to Python.org. Retrieved from https://www.python.org/

[42] Google Developers. (n.d.). TensorFlow. Retrieved from https://www.tensorflow.org/

[43] NumPy Developers. (n.d.). NumPy. Retrieved from https://numpy.org/

[44] The Pandas Development Team. (n.d.). Pandas Documentation. Retrieved from https://pandas.pydata.org/

[45] Scikit-learn Developers. (n.d.). scikit-learn: Machine Learning in Python. Retrieved from https://scikit-learn.org/stable/

[46] Matplotlib Development Team. (n.d.). Matplotlib: Visualization with Python. Retrieved from https://matplotlib.org/

WEB GRAPHY

- Dataset 1: OULAD dataset
- Dataset 2: TsinghuaX MOOC dataset

الجمهورية الجزائرية الديمقراطية الشعبية وزارة التعليم العالي والبحث العلمي

جامعة غرداية كلية العلوم والتكنولوجيا

Université de Ghardaïa

Faculté des Sciences et de la Technologie

قسم: Mathématiques et Informatique

شعبة: Informatique

تخصص: Systèmes intelligents pour l'extraction de connaissances

غرداية في : 09/11/2025

شهادة ترخيص بالتصحيح والايداع:

انا الاستاذ(ة). BRAHIM Nacéra.

بصفتي المشرف المسؤول عن تصحيح مذكرة تخرج (ليسانس/ماستر/دكتورا) المعنونة بـ:

Validating AI Predictions in Education: A Comparative Study of Methods and Datasets

من انجاز الطالب)الطلبة:(

Toufik BAZEMLAL && Messaoud MOSBAH

التي نوقشة/قييمت بتاريخ: 02 جويلية 2025

اشهد ان الطالب/الطلبة قد قام /قاموا بالتعديلات والتصحيحات المطلوبة من طرف لجنة المناقشة وقد تم التحقق من ذلك من طرفنا وقد استوفت جميع الشروط المطلوبة .

امضاء المسؤول عن التصحيح

Brh.N

مصادقت رئيس القسم

جامعة غرداية

قسم الرياضيات و الاعلام الآلي

كلية العلوم والتكنولوجيا

رئيس قسم الرياضيات والإعلام الآلي

الحاج موسى ياسين