EXPLAINABLE MACHINE LEARNING FOR EDUCATIONAL DATA

A thesis submitted as partial fulfilment of the requirement of Doctor of Philosophy (Ph.D.)

by

Mashael Al Luhaybi

Department of Computer Science

College of Engineering, Design and Physical Sciences

Brunel University London

April 2020

Abstract

Educational repositories contain complex trajectories of students and university data. Being able to model this data would offer great value in being able to identify students' trajectories, predicting their likely future performance, and identifying those who require appropriate intervention as early as possible. However, understanding the nature of the correlations and the dependencies among the educational attributes (which can be time-dependant non-linear relationships) is fundamental for the learning of robust predictive classifiers. When predicting academic performance, many machine learning algorithms make decisions based on data that can be imbalanced, badly sampled, or biased based on historical societal prejudices.

In this thesis, I explore, implement, and evaluate temporal predictive classifiers that aim to overcome some of these issues. The approach combines time-series clustering in conjunction with probabilistic learning, resampling, feature subspace learning, and specialist deep learning methods to learn models that are simultaneously accurate and unbiased. A key technical objective in learning these classifiers is to incorporate different types of temporal performance data collected at different times (student admission to a higher education institution, and at Year 1 and 2 of a student's studies), for the explicit modelling of cognitive styles. A resampling method is applied with bootstrap aggregating to address the issue of the imbalanced time-series educational datasets, which is related to miss-classifying the minority-class of the high-risk or failing students. The evaluation of an unsupervised subspace learning approach using an Autoassociative Neural Network (Autoencoder) is also made, to reconstruct the educational data by maximising variance for improved performance prediction. In addition, the issues of modelling bias are explored such that the types are identified and whether they are accounting for inflated predictive accuracies is established. A graphical learning approach with a BN, that is transparent in how they make decisions, is compared with three forms of Deep Multi-label Convolutional Neural Network (CNNs) to investigate whether deep learning classifiers can be learned that maximise accuracy and minimise bias.

The evaluation of the experimental results reveals that identifying cognitive styles improves both explanation and accuracy; that rebalancing also improves accuracy, and that a combi-

nation of probabilistic modelling and deep 1D Multi-label CNN can successfully identify and eliminate many biases when predicting student's academic performance.

Dedication

This thesis is dedicated with gratitude to my parents Fahad and Ebtisam who have been always encouraging and supportive despite the distance. I also dedicate this thesis to my beloved husband Thamer, my sons Ahmed and Yazan, and my little daughter Lana. No words can express the love and gratitude that I owe for them, for the support and encouragement that they have been providing during all the stages of my Ph.D. There is no doubt that I could not have been able to conduct this research without their unlimited support, which made this journey possible. I am very grateful to all of you.

Acknowledgements

First and foremost, I would like to express my sincere gratitude and appreciation to my supervisory team, who provided constructive supervision during my Ph.D. journey at Brunel. In particular, my thanks go to my principle supervisor Dr Allan Tucker for his advice, insight and unconditional support throughout these years. Without his constant support and patience, this research would not have been completed. It was a privilege to be supervised by him. I would like also to extend my appreciation to my second supervisor Dr Steven Swift for his suggestions and comments throughout this research.

I also owe special thanks to Prof. Steve Counsell, my research development advisor, for his professional cooperation in the conducted research for this thesis as well as in giving me the opportunity to participate in the experimental works that have been partially conducted for Brunel's internal project on students' assessments and retention (STARS Project).

I would also like to thank the Saudi Arabian Cultural Bureau in the United Kingdom for the financial support and the excellence awards for my progress in participating at the academic conferences, which kept me motivated during all the ups and downs of my Ph.D. I also owe a special thanks to Umm Al-Qura University in Saudi Arabia (where I was based before coming to the UK and where I will be working at on my return) for providing this opportunity to pursue my Ph.D. at Brunel University, London.

Special thanks also go to all the staff members and my colleagues in the Department of Computer Science, especially Leila Yousefi, Roja Ahmadi, Anas Alkasasbeh, Nada Al-Subhi and Khalid Eltayef for their assistant and encouragement throughout the research process.

Last but not least, I cannot forget to thank my family back home, particularly my brothers and sister for their love and the kind wishes since I started my PhD journey. Also, special thanks go to my best friends in London, Abeer Alqahtani and Azizah Alyoubi, for sharing wonderful moments with me and my family which made my time in London memorable.

Publications

The following publications have resulted from the research conducted in this thesis:

- Al-Luhaybi, M., Swift, S., Counsell, S., Yousefi, L., and Tucker, A. Temporal Profiling of Online Self-Assessment Sequential Trajectories for Improved Performance Prediction (in review). International Journal of Artificial Intelligence in Education (IJAIED). Springer.
- Al-Luhaybi, M., Swift, S., Counsell, S., and Tucker, A. Exploring the Explicit Modelling of Bias in Machine Learning Classifiers: A Deep Multi-label Convolutional Neural Network Approach (in review). Submitted to the IEEE 2020 International Joint Conference on Neural Networks (IJCNN 2020).
- Al-Luhaybi, M., Yousefi, L., Swift, S., Counsell, S., and Tucker, A. (2019, June). Predicting Academic Performance: A Bootstrapping Approach for Learning Dynamic Bayesian Networks. In the 20th International Conference on Artificial Intelligence in Education (AIED 2019) (pp. 26-36). Springer International Publishing. (A double-blind accepted paper with an acceptance rate of 25% for the full papers).
- Al-Luhaybi, M., Tucker, A. and Yousefi, L. (2018, April). The Prediction of Student Failure using Classification Methods: A Case Study. In the 4th International Conference on Artificial Intelligence and Soft Computing (AIS 2018) (pp. 79-90).

Symposiums Participation

- Al-Luhaybi and Tucker, A. (2019, March). Temporal Profiling of Online Self-Assessment Trajectories for Performance Prediction. Paper presented at Brunel PhD Symposium 2019.
- Al-Luhaybi, M., Yousefi, L., Swift, S., Counsell, S., and Tucker, A. (2018, October). Identification of Student Types from Online Self-assessment Temporal Trajectories with Dynamic Time Warping for Performance Prediction. Poster presented at the 17th International Symposium on Intelligent Data Analysis (IDA 2018).

Contents

List of Figures

List of Tables

Acronyms

- **AHC** Agglomerative Hierarchical Clustering
- **AUS** Area Under Curve
- **BN** Bayesian Network
- **CI** Confidence Interval
- **ConvNet/CNN** Convolutional Neural Network
- **CPD** Conditional Probability Distribution
- **CPT** Conditional Probability Table
- **CV** Cross Validation
- **DAG** Directed Acyclic Graph
- **DBN** Dynamic Bayesian Network
- **DTW** Dynamic Time Warping
- **EDM** Educational Data Mining
- **EM** Expectation Maximization
- **MB** Markov Blanket
- **ML** Machine Learning
- **MMPC** MAX-MIN Parents and Children
- **MSE** Mean Squared Error
- **NLPCA** Non-linear Principle Component Analysis
- **PCA** Principle Component Analysis
- **ROC** Receiver Operating Characteristics Curve

- **SMOTE** Synthetic Minority Over-sampling Technique

Chapter 1

Introduction

1.1 Motivation

Predicting student performance is a major area of interest within the field of Educational Data Mining (EDM). For example, ascertaining accurately what final grades (Romero, Lopez, Luna and Ventura, 2013), will be achieved. However, it is a very complex task as it is influenced by social, environmental and behavioural factors (Bhardwaj and Pal, 2012) (Araque et al., 2009). Identifying these factors will enable predictive models to identify high risk students who may need intervention. Many educational datasets are complex as they often capture a temporal process with only a small number of observations. They are also often imbalanced with only a limited number of high-risk students (likely to fail). Learning from such data requires novel approaches and combinations of techniques to transform such data into useful knowledge (He and Garcia, 2009). As well as these technical challenges, data driven models are facing other challenges as many make decisions based on data that are biased and can therefore result in prejudiced decisions. For instance, in higher education, which this thesis focuses on, a student may be rejected from a course or an academic program based on historical decisions in the data that only exist due to historical biases in society. Furthermore, educational datasets can have missing data and biased samples. A predictive model may therefore predict a student as a high risk or failing student based on decisions in the data that only exist due to the skewed sampling of such data. These issues of course are not solely the preserve of educational data

mining but are also very common issues in many other applications involving sensitive variables and important decision making.

Ensuring trust in the AI data driven models is crucial for many of the real-world challenges. Indeed, the whilst AI algorithms have demonstrated abilities in making accurate predictions, there is no consensus in what algorithms constitute useful explanations as to how and why the predictions are made. Explainable AI (XAI) has emerged as a topic to explore explainable models while achieving as good or better prediction. This is to overcome the dominance of black-box algorithms which have become very powerful at mapping inputs to outputs (Jia et al., 2018). Some traditional approaches such as rule-based and graphical-based classifiers (e.g. Bayesian Networks (Pearl, 2011)) have become popular again as they can be used to generate explanations because knowledge is more explicitly represented. However, these methods do not ensure against unbiased predictions: models are only as good as the data that is used to train them.

In the research area of AI in education, a shift in thinking is required, from unexplainable decisions by machine learning classifiers to a new approaches where we explicitly consider the dependencies and correlations between factors to generate accurate and explainable models. In this thesis, a target will be the implementation of the state-of-the-art machine learning techniques based on Bayesian Networks and Deep learning to provide explainable predictive models or classifiers with higher educational temporal data.

In this connection, this chapter seeks to address the motivations, aim, objectives and contributions of the research conducted in this thesis.

1.2 Thesis Aim

The works conducted in this thesis are aimed at identifying accurate, reliable, and interpretable temporal predictive classifiers. The importance of exploiting temporal models based on students' progression time series trajectories is determined in this work. This allows for the capturing of students' dynamics so as to provide accurate and reliable predictions of academic performance in different time slots of the class modules. In addition, the structure learning

modelling and the deep learning technique were integrated with the aim of identifying explainable classifiers that are transparent in their reasoning and decisions for ethical decision making with academic performance prediction. By exploiting these modelling approaches, the target is to produce explainable models, while enabling researchers not only to understand the process, but also, to trust these AI models. If there is conflict between accurate, reliable, and explainable temporal predictive classifiers, the priority is setting the most suitable performance metric for evaluating each machine learning problem For example, if the target of the machine learning task is to understand the different dynamics of students and identify their relation to overall performance (as is the case for this thesis), then the accuracy is an important performance measurement as well as the reliability of the explained model.

1.3 Thesis Objectives

The objectives of this thesis are specified bellow:

- Identify the temporal cognitive styles of students in Year 1, Year 2 and 3 based on students tutoring engagement trajectories and other related data using time-series clustering approaches.
- Identify temporal profiles from the online self-assessment trajectories using distance-based similarity clustering methods on time-series data of student engagement.
- Implement a resampling approach using Bootstrap Aggregating on time-series educational datasets to deal with imbalance in educational datasets.
- Develop an approach for the explicit modelling of bias in machine learning classifiers with Bayesian Networks and Deep Multi-Label Convolutional Networks.

1.4 Thesis Contributions

The main contributions of this thesis are provided bellow:

- **Discovery of student's "Temporal Cognitive Styles" based on their engagement (attendance) throughout a module.** I explored clustering students' tutor group engagement trajectories using Dynamic Time Warping and Agglomerative Hierarchical clustering algorithms to group students to different clusters based on their engagement level. We then clustered this information in conjunction with students' admission and progression (final grades) for all Year 1 and Year 2 Modules to identify students' Cognitive Styles.

- **Identification of students "Temporal Profiles" from online self-assessment Progression trajectories of an online module.** A distance-based similarity clustering approach for time-series data has been discovered using Dynamic Time Warping (DTW). This approach was explored to identify different profiles of students' online self-assessment trajectories in order to improve the prediction of student's overall performance.

- **Improvement to performance prediction using an unsupervised subspace learning/dimensionality reduction approach.** I explored classifying students using nonlinear subspace learning (or NLPCA) via an auto-associative neural network (Autoencoder) to reconstruct the educational data by minimizing the squared error and maximizing variance. The nonlinear subspace learning was exploited to maximize variance to reduce bias and therefore, improve the classification.

- **Incorporation of student's cognitive styles and online self-assessment profiles into an ordered Bayesian Network for predicting student performance at different stages of their study.** I used a structure learning algorithm on students' time-series data at admission level, Year 1 and Year 2, respectively, for the early detection of students at risk of failing or dropping out at each stage of their study.

- **Improvements to the predictive model of the ordered Bayesian Network using resampling methods.** I extended the investigation of the probabilistic modelling of the time-series educational data to address the issue that is related to the imbalanced datasets, especially for classifying the minority-class high-risk students.

- **An exploration in transparent models for predicting student performance using linear and non-linear PCA, CNNs and BNs.** I carried out an exploration to compare transparent and explainable classifiers using Bayesian Networks and deep multilabel convolution networks. In particular, I consider modelling the biased features during classification to ensure decisions are not affected by sensitive features such as ethnicity or gender (including proxies in other features).

1.5 Thesis Outline

This section explains the structure of the thesis, which is composed of seven chapters including this introduction. Chapters 3 to 6 have independent research aims and experimental works in order to achieve the overall aim and objectives of this thesis.

Chapter 2 begins by providing background information related to the state-of-the-art methods that been explored in Educational Data Mining (EDM) research field especially for predicting student performance, which is the focus of this thesis. Furthermore, this chapter is considering background information of the machine learning methods that been used for learning unbiased classifiers such as: probabilistic modelling, dimensionality reduction/subspace learning and deep neural networks.

Chapter 3 introduces the initial analysis of the educational datasets to identify the key features affecting the prediction of academic performance as well as the cognitive styles detection. This chapter is also exploring the implementation of the subspace learning techniques, the linear PCA and non-linear PCA, to improve predicting performance.

Chapter 4 provides a novel approach for temporal profiling of online Self-Assessment trajectories using a distance-based similarity approach with Dynamic Time Warping (DTW) for improving performance prediction.

Chapter 5 describes the implementation of a structure learning approach with bootstrap aggregation for learning ordered Bayesian Networks from time-series data. This chapter also investigates the emphasis of a resampling method to deal with imbalanced classes in the educational datasets.

Chapter 6 explores a comparison of transparent machine learning models with Bayesian Networks (BNs) and Multi-labels Deep Convolutional Neural Networks (CNNs) for generating unbiased classifications. In this chapter the issues of modelling bias explicitly have been explored so that biased features can be identified that lead to inflated predictive accuracies.

Chapter 7 provides a conclusion of the experimental works conducted in this thesis, limitations of the research and future directions for the development of reliable and ethical machine learning classifiers.

Chapter 2

Backgrounds and Literature Review

2.1 Introduction

In recent years, there has been an increasing interest in applying machine learning algorithms and techniques in various fields such as medicine, marketing, education, engineering and so forth, due to its advances in transforming huge amount of such data into useful knowledge. machine learning (ML), or in other words Knowledge Discovery in Databases (KDD), can be defining as a multi-disciplinary field in which several computing paradigms converge: decision-trees, artificial neural networks, rule induction, instance-based learning, Bayesian learning, logic programming, statistical algorithms, etc. The most well-known ML techniques are Clustering, Classification, Association rule mining and Description and visualisation (Hand, 2007).

However, the Higher Education Statistics Agency (HESA) in the UK (HESA, 2019) has revealed that the drop-out rate among undergraduate students has increased in the last three years. The statistics published by the HESA in 2016 reported that a total of 26,000 students in England in 2015 dropped out from their enrolled academic programs after their first year. Also, the statistics show that the higher education (HE) qualifications obtained by students for all levels, including undergraduate and postgraduate levels, decreased from 788,355 in 2012/13 to 757,300 in 2016/17. The growing availability of such statistics provides new opportunities for researchers to investigate the issues associated with students' learning and overall achievements. For instance, several attempts have been exploited to analyse and evaluate student

data to enhance their education and provide solutions to failure issues using the state-of-the-art Artificial Intelligence (AI) methods. These attempts seek to analyse students learning and performance data to develop novel machine learning approaches that benefit the students and enhance their learning process.

However, a major issue with academic performance prediction is that most of the available applications do not consider the data-related issues when implementing the classifiers, such as imbalanced classes as well as noisy and biased data. Reliable educational data plays a crucial role in learning predictive and explainable models aimed at identifying and evaluating the most predictive educational attributes to students' end-of-course grade, and to identify the time-dependent attributes more precisely, thus allowing for appropriate interventions in real time. If the predictive model can characterise the students with high risk of failure prior the examination from sequence data (e.g. sequence mining), then the academics can focus on providing extra effort to improve such students' performance and thus, enhance their likelihood of passing the module and/or obtaining higher results. Hence, detecting the data-related issues is another important aspect for learning robust predictive models.

The explainability of the predictive models is another major challenge with student performance prediction, and the main impediment to this is implementing the classifiers using black box algorithms, such as neural networks that are unexplainable in the way they take their decisions. Despite the high accuracy results of the available predictive models, limited research has been conducted to consider the transparency of the predictive classifiers when using black box models. Such models are not reliable as they usually result in biased predictions.

This chapter seeks to remedy these learning and prediction issues by analysing the literature of educational data mining so as to identify the most predictive attributes and effective ML algorithms used for academic performance prediction. In addition, I present the research conducted far for the implementation of the probabilistic modelling approaches, such as BN, to determine the most reliable and explainable predictive classifiers from the educational data.

2.2 Educational Data Mining

The growing availability of data in educational databases attracts many researchers to analyse and evaluate such data to enhance education and provide optimal solutions for the associated issues. This emerging discipline is called Educational data mining (EDM) where we apply data mining (DM) techniques or develop new DM methods to explore educational data in order to understand student's learning process and their outcomes (Peterson et al., 2010). Educational Data Mining (EDM) seeks to analyse student learning by developing data mining approaches that merge student data and machine learning algorithms to benefit the students and enhance their learning outcomes. Therefore, the EDM process transforms raw learning and performance data into useful knowledge that benefits educational research. This process passes through a five step process, which is the same as the general DM process: data selection, data pre-processing, data transformation, application of machine learning algorithms and evaluating and deploying the discovered knowledge.

With the recent developments in education, therefore, there has been an increasing interest in EDM which has been emerged in related areas, for instance (Romero and Ventura, 2007):

- **Web based-educational systems:** applying data mining to data stored E-Learning and Learning Management Systems (LMS). These systems provide online tools for instruction, collaboration, communication and administration;
- **Offline education:** understanding the psychometric technique for student learning in face-to-face teaching environments;
- **Intelligent Tutoring System (ITS):** the adaption of teaching approach for each individual need of the students. This approach is another form of the just-put-it-on-the-web approach.

All the above mentioned EDM areas have different datasets and educational issues to be resolved using different data mining techniques. From a practical view point the web-based educational systems provide details on student interactions and learning process. For example, the log files contain data on how many times a student logs into the system or how many

times they visit specific content in the course. In addition, these systems record data for each student's submission to a specific assignment, such as the time spent answering the question and if their answer matches the correct answer or not (Romero and Ventura, 2007).

EDM also helps the academics/instructors to make correct decisions on choosing the learning environments to best meet student needs and course objectives (Romero and Ventura, 2007). Furthermore, analysing students' data generated from educational systems provides better educational processes as EDM allows for the discovery of information about what students need to improve and where they did well, and this can be used to identify good examples (Merceron and Yacef, 2005). However, the process of applying DM in educational systems is unlike applying it in other systems. When we apply data mining methods to data coming from the learning management systems (e-learning systems), it should enter a loop of the system, not just to turn data into knowledge but also to make decisions about the filtered data (Romero and Ventura, 2007) (see Figure 2.1).

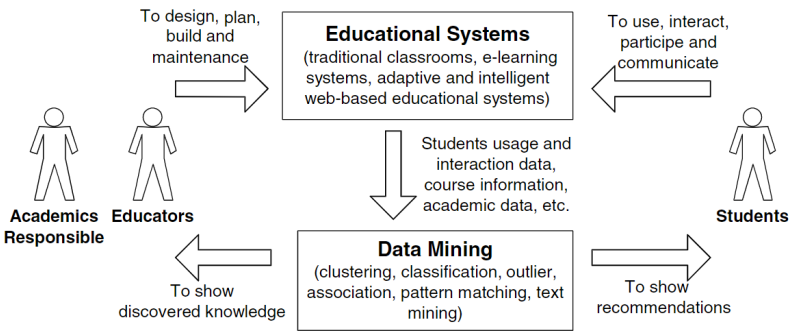


Figure 2.1: The cycle of DM in educational systems(Romero and Ventura, 2007)

As we notice in Figure 2.1, the academics are the main responsible for designing, planning and developing the online course in the educational systems. Then, the students can use the educational materials, interact and participate in the online activities and assessments. Starting from students participation and interactions, different machine learning algorithms can be applied to discover useful patterns to indicate students' development as well as the online course.

A serious weakness with the current data mining cycle displayed in Figure 2.1, however,

is that it does not take into account the transparency of the decisions made by the data mining algorithms. This will consequently lead to another critical issue, which is making unethical decisions based on black box algorithms. It is very important to explore the DM process in terms of providing better knowledge on the behaviour of the algorithm. This will produce ethical as well as explainable decisions to enable the instructors/educators not only to understand, but also, to trust these decisions.

2.2.1 Machine Learning Applications in Education

Machine Learning (ML) has been applied to resolve several issues in educational environments, including: the detection of students learning behaviours, prediction of academic performance, provision of recommendations, and feedback on students learning progress to both the academics and the students. On one hand according to (Baker et al., 2010) there are four main areas of applying ML methods in educational setting: improve students' models, improve domain models, study the pedagogical aspects of the learning software and investigate learning process. On the other hand, (Castro et al., 2007) identifies five applications of EDM: assessment of student performance, making learning recommendations based on students behaviour, evaluation of web-based educational courses and course contents, provision of feedback to instructors and students in web-based educational systems, and applications for detecting students behaviours. (Romero and Ventura, 2010) have classified therefore the tasks of applying data mining identified in the previous publications based on the main focus of researchers. The prediction of student academic performance was the main target, followed by analysing and visualizing data and detecting leaning behaviour. A summary of ML applications in education and their descriptions are presented in Table 2.1.

2.2.2 Machine Learning Tasks in Education

Classification

Classification, a form of supervised learning, is a very common ML technique that is applied to map datasets into sets of classes (Aher and Lobo, 2011). To develop such models, the data undergo a process that consists of learning and classification. In the learning process, the

Table 2.1: Machine learning applications in Education.

EDM Application	Description
Predicting student performance	To predict the unknown outcomes of student knowledge, scores or final grades.
Detecting students behaviours	To analyse students learning styles and learning preferences.
Providing feedback to instructors	To provide information to the instructors about students learning processes to help them improve their learning throughout the course.
Providing recommendations to students	To provide recommendations to students about their tasks and activities in order to overcome any associated issues.
Providing groups of students	To group the students according to their characteristics, learning data or learning performance.
Developing students modelling	To develop cognitive students models that reflect their skills and the obtained knowledge.
Constructing courseware	To provide guidance to the instructors and the developers of the course to better develop effective courseware.
Analysing social network	To study the relationship between individuals rather than individual features or attributes
Developing concept maps	To develop a graph that presents the relationship between the concepts and the hierarchal knowledge structure.
Planning and scheduling	To plan for future courses scheduling, resources development, and admission processes.
Analysing and visualizing data	To provide information with regard to educational process or outcomes as reports or statistics.

training set is analysed using classification algorithms to generate logical rules based on the relation between the selected attributes. Consequently, the classification process identifies the accuracy of the model by applying obtained rules on the test sets to evaluate the classifier (Bhardwaj and Pal, 2012). In educational data mining for instance, the student work grades in a particular course can predict the final results or overall performance in that course using classification algorithms.

Prediction

Prediction is a widely used data mining task to model continuous valued functions (Abu Tair and El-Halees, 2012). It can be applied through classification algorithms to predict the unknown class or through regression to predict the unknown or missing value. In educational data mining prediction can be used to predict student final grades, educational outcomes and identify the weak students among their classes, and so forth. More exploration about predicting

student performance or outcomes are predicted in the following chapters.

Clustering

Clustering is identifying groups of objects in which the objects of such groups are similar to one another in some aspects and different from the objects in the other groups (Romero and Ventura, 2010). Clustering is considered as one of the most applied unsupervised learning technique in machine learning. In educational data mining, clustering is applied to group the students according to their performance in the course into weak and strong students to help the weak students improve their studies (Hogo, 2010); (Perera et al., 2008). Also, clustering method used to identify the active and the non-active students based on their performance in course activities (Aher and Lobo, 2011).

Association analysis

Association analysis is a data mining task used to analyse the relationship between data objects to identify interesting correlation among these objects in a given dataset (Aher and Lobo, 2011). Thus, it generates rules of the form 'If antecedent then (likely) consequent' where antecedent and consequent are items sets consisting of one item or more. According to (Aher and Lobo, 2011), association rules mining is usually generate through two steps. First step is identifying the frequent item sets in the database. The second step is generating the rules through the item sets and the minimum confidence constraint. Association analysis has been applied in educational data mining for analysing student learning performance. For instance, if the student attendance is good, then we can estimate the final grade of the student as excellent or very good.

Outlier/novelty detection

Outlier detection is discovering data objects in which are significantly different from the other data objects in the database (Mansur et al., 2005). In educational data mining, outlier detection can identify the students with learning issues (Romero and Ventura, 2007). Also, it used to detect the outlier students behaviour. Therefore, applying outlier detection method the instructors can identify the reasons of the irregular behaviour of the students and finding solutions for the discovered students.

2.3 Modelling Classifiers for Academic Performance Prediction

Researchers have been increasingly attempting to analyse students' datasets with machine learning algorithms in order to understand how students learn and to ultimately increase the performance of students and the quality of learning. Thus, considerable amount of research has been published on predicting the performance of the students based on different factors and attributes. This section will review the classification methods used for learning the predictive classifiers. Also, the influential attributes on performance prediction will be explained.

2.3.1 Learning Predictive Classifiers

The most popular machine learning task to predict student academic performance is classification. There are several ML algorithms have been applied to classify students data and predict their performance. Among them are Decision Trees, Random forest, Bayesian Networks, Naïve Bayes, Neural Networks, K Nearest Neighbour and Support Vector Machines. Also, classification via clustering approach has been applied to predict student overall performance.

The ML algorithms applied in predicting student academic performance as presented as follows:

- **Decision Tree (DT)**

Decision tree is a tree-like structure that provides decisions for specific issues. These decisions provide rules for classifying the dataset (Bhardwaj and Pal, 2012). A decision tree consist of internal, child and leaf nodes. The internal node splits intro two or more child nodes depending on the learning algorithm. The internal and child nodes are linked by arcs, which are labelling the result of bsaed on the test cases in the datasets (Yadav and Pal, 2012). Consequently, the leaf nodes contain the classes of the class attribute.Thus, decision tree is one of the most commonly used approaches for predicting students' performance. Several researchers have applied this method because of its efficiency to predict the performance of the students even for the small structured datasets (Quadri and Kalyankar, 2010).

When predicting performance, the C4.5 decision tree is one of the most commonly used algorithms (Quadri and Kalyankar, 2010); (Kabakchieva, 2013). Several researchers have applied it owing to its efficiency in predicting student performance, even for small structured datasets (Merceron and Yacef, 2005). According to Romero et al., C4.5 models can easily be converted into sets of IF-THEN-ELSE Rules (Romero and Ventura, 2007). Analyzing decision trees rules is beneficial for educational researchers to extract useful knowledge about what the most effective factors "attributes" are and how they influence the prediction of student performance (Romero and Ventura, 2007). Also, Cortez and Silva (Cortez and Silva, 2008) conducted a study using a combination of three datasets of student attributes, these: demographics, social aspects and past grades. By means of classification algorithms, in particular, decision trees, the authors found that the performance of students is highly influenced by past grades. Furthermore, they achieved a very reasonable accuracy result of 76.70% in predicting the student future performance. In the same vein, Al-Radaideh et al. (Al-Radaideh et al., 2006) conducted a study to generate a predictive model of the performance of computer science students at Yarmouk University in Jordan using the same algorithm (the C4.5 decision tree algorithm). They used student demographics, high school grades and tutors' data, including their degree, affiliation and gender.

A broader perspective has been adopted by some authors to compare the performance of the C4.5 decision tree algorithm and other classification algorithms in predicting weaker students in the early stages of a course. Yadav and Pal (Yadav and Pal, 2012) applied different decision tree algorithms (ID3, C4.5 and CART) to an engineering student dataset to identify weaker students among their groups before the final exam. The student datasets included demographical, social, environmental and psychological attributes. They revealed that the C4.5 algorithm gave the best result, with an accuracy of 67.77% in predicting the weaker students in targeted courses. Similarly, Kaur and Singh (Kaur and Singh, 2016) compared decision tree algorithms and other classification methods regarding student demographics and psychological data to investigate their weaknesses, obtaining 61.53 % accuracy for the c4.5 decision tree model. In addition to previous studies, preliminary work on predicting slow learners was undertaken by Kaur et al. (Kaur et al., 2015). They analysed high school datasets, including student demographics and

admission data, to identify slow learning students, with the aim of improving their performance prior the examination.

However, the studies presented thus far, provide evidence that they have not achieved greatly accurate results, for 76.70% is the highest accuracy for the decision tree models. Hence, further investigations need to be undertaken to generate more accurate predictive models for modelling and predicting the academic performance of students using the C4.5 algorithm. Considered together, these studies outline that C4.5 produces reasonable accuracies, even though Support Vector Machines or Neural Net may provide better. We have decided to continue using C4.5 since we are interested in understanding the meaning of DTW clusters in the prediction results.

Table 2.2: Predictive models accuracy results based on decision trees.

ML Method	Attributes	Accuracy	Authors
Decision Tree	Students Interaction data & Web usages data (log files)	80.30%	(Minaei-Bidgoli et al., 2003)
	Past school grades (rst and second periods), demographic and social data	76.70%	(Cortez and Silva, 2008)
	High school dataset (Demographic, Personal Data and Admission data)	69.73%	(Kaur et al., 2015)
	Personal, social and psychological data	67.77%	(Yadav and Pal, 2012)
	Personal and pre-university data	65.94%	(Kabakchieva, 2013)
	Demographic and psychometric factors	65.93%	(Gray et al., 2014)
	Students Interaction data & Web usages data (log files)	65.86%	(Romero, Espejo, Zafra, Romero and Ventura, 2013)
	Demographic, personal and psychological data	61.53%	(Kaur and Singh, 2016)
	Demographic, personal and tutors related data	38.05 %	(Al-Radaideh et al., 2006)

A considerable number of studies have been conducted to predict the performance of the students based based on some extracted attributes from the educational datasets as shown in Table 2.2. The best perform decision tree model for predicting the academic performance of the student was based on the extracted data from students log files in web based educational systems (Minaei-Bidgoli et al., 2003). Also, past school grades and pre-university data are highly influencing student academic performance when have been modeled with decision tree (Cortez and Silva, 2008); (Kabakchieva, 2013). However, most of the researchers have involved

personal, social, psychometric and psychological characteristics of the students to develop the decision tree model and they obtained a very accurate results for the predictive model (Cortez and Silva, 2008); (Kaur et al., 2015); (Yadav and Pal, 2012); Kabakchieva (2013); (Kaur and Singh, 2016); (Al-Radaideh et al., 2006).

- **Artificial Neural Network (ANN)**

An artificial neural network is a computational approach that consists of groups of artificial neurons inter-connected to each other by process information. Neural network is one of the most widely used successful method for predicting academic performance because of its advantage as an adaptive model that could influence and change based on the internal and external information passed during the learning process (Romero and Ventura, 2010). Numerous neural network algorithms have been utilized for the purpose of predicting student performance, the well-known among them are: multilayer perceptron (MLPreceptron), a hybrid genetic neural network (GANN), neural network evolutionary programming (NNEP) and a radial basis function neural network (RBFN).

As illustrated in Table 2.3, ANN is one of the best classification methods in predicting the performance of the students as the models achieved high prediction accuracy results. Thus, most researchers have compared neural network approach with other classification approaches to predict student final grades and overall achievement based on some online aspects such as online assessment grades and students participation in the online dissection forum. It has been found that ANN performs better than the other classification techniques as the learned classifiers obtained very high accuracy percentages (Lykourentzou et al., 2009); (Romero, Espejo, Zafra, Romero and Ventura, 2013).

- **Naïve Bayes (NB)**

Naïve Bayes algorithm is a simple classification method based on probability theory (Witten and Frank, 2002), such probability predicts the membership of all attributes and the class attributes by assuming that the independency of the class attributes is based on the associated

Table 2.3: Predictive models accuracy results based on Artificial Neural Network (ANN).

ML Method	Attributes	Accuracy	Authors
Artificial Neural Network	Courses final grades for semester 1	97.74%	(Arsad et al., 2013)
	Online Assessment grades	95.21%	(Lykourentzou et al., 2009)
	Student demographic, social data and past school grades (first and second periods)	90.07%	(Cortez and Silva, 2008)
	Student participation in online forum	88.06%	(Romero, Espejo, Zafra, Romero and Ventura, 2013)
	Students demographic and CGPA	80%	(Ibrahim and Rusli, 2007)
	Student demographic and high school grades	74%	(Oladokun et al., 2008)
	Online assessment data (online quizzes and assignments)	65.95%	(Romero, Espejo, Zafra, Romero and Ventura, 2013)

values with the other attributes in the prediction model (Kabakchieva, 2013). Thus, the independent effect of the attributes in the classification model plays a crucial role in classifying the instances. Consequently, Naïve Bayes determines the accuracy of the classification model according to the classified instances. Thus, a large volume of published papers were examining Naïve Bayes prediction methods to predict student performance. As shown in Table 2.4, researchers have been attempting to analyse students demographic, social and assessment data to predict the slow learning students in order to improve their performance and reduce failure rate prior the exam (Mayilvaganan and Kalpanadevi, 2014); (Kaur et al., 2015). Also, there are several studies which compared Naive Bayes method with other classification methods to classify the students and identify their abilities, interests and weaknesses (Kabakchieva, 2013); (Kaur and Singh, 2016). These studies are summarised in Table 2.4 with the class accuracy results of the predictive models.

- **K-Nearest Neighbour (KNN)**

K-Nearest Neighbour, or nearest neighbour method, is a very simple data mining method that classifies the instances in the datasets depend on the classes of the K- instances that are most close to them (Bhardwaj and Pal, 2012). As depicted in Table 2.5, several researchers have conducted comparative studies to classify student's performance using K-Nearest Neighbour method. According to Mayilvaganan and Kalpanadevi (Mayilvaganan and Kalpanadevi, 2014)

Table 2.4: Predictive models accuracy results based on Naïve Bayes classifiers.

ML Method	Attributes	Accuracy	Authors
Naïve Bayes	Socio-demographic, high school grades and entrance exam	76.65%	(Osmanbegovic and Suljic, 2012)
	Demographic, GPA and course assessments data	73%	(Mayilvaganan and Kalpanadevi, 2014)
	Demographic and psychometric data	68.03%	(Gray et al., 2014)
	Demographic, social data and past grades (first and second periods)	65.13%	(Kaur et al., 2015)
	Demographic, psychological and environmental data	63.59%	(Kaur and Singh, 2016)
	Demographic and pre-university data	58.10%	(Kabakchieva, 2013)

and Minaei-Bidgoli et al. (Minaei-Bidgoli et al., 2003), KNN approach can identify the slow learners among students to overcome their difficulties and improve their skills based on their assessments data. Also, KNN predictive models can identify students at risk of fail the course based on a combination of demographic and psychometric factors (Gray et al., 2014). Classifying student's data into bad, average, good or excellent performance taking into account their demographic and pre university data has been also explored by Kabakchieva (Kabakchieva, 2013).

Table 2.5: Predictive models accuracy results based on K-Nearest neighbour.

ML Method	Attributes	Accuracy	Authors
K-Nearest Neighbour	Demographic, CGPA and students assessments data	83%	(Mayilvaganan and Kalpanadevi, 2014)
	Online assessment and interaction data in LON-CAPA educational system	82.30%	(Minaei-Bidgoli et al., 2003)
	Demographic and psychometric data	69.43%	(Gray et al., 2014)
	Demographic and pre-university data	60%	(Kabakchieva, 2013)

- **Support Vector Machine (SVM)**

Support vector machine is another good classification method for predicting students' academic performance. It classifies the dataset by selecting support vectors (data points) in order to define wide linear margin between the classes (Hämäläinen and Vinni, 2006). In practice,

SVMs are best designed for the small size datasets as the learning process of the classifier is selecting only some data points not taking into consideration the dimension or the size ratio of the dataset.

As shown in Table 2.6, SVM models obtained excellent accuracy results and usually considered as one of the most optimal classification methods. A number of researchers have applied SVM method in their comparative studies to predict the performance of the students. Hamalainen et al. (Hämäläinen and Vinni, 2006) have mined students assessment data and CGPA to predict their success in the course. Similarly, Sembiring et al. (Sembiring et al., 2011) have analysed students' behavioural factors in order to predict students' success. In contracts, Cortez and Silva (Cortez and Silva, 2008) applied SVM method on secondary school students demographic, social data and past school grades to predict students' final grades.

Table 2.6: Predictive models accuracy results based on Support Vector Machine (SVM).

ML Method	**Attributes**	**Accuracy**	**Authors**
Support Vector Machines	Demographic, social data and past school grades (first and second periods)	86.3%	(Cortez and Silva, 2008)
	Psychometric data	83%	(Sembiring et al., 2011)
	Demographic, CGPA and students assessments data	80%	(Mayilvaganan and Kalpanadevi, 2014)
	GPA and students assessment data	80%	(Hämäläinen and Vinni, 2006)

- **Classification via Clustering**

Classification via clustering is a prediction approach that employs via using clustering algorithms to classify the instances in the dataset assuming that each cluster maps to a class (Romero, Lopez, Luna and Ventura, 2013). The mapping between the cluster and the class attribute in the training set is used then for predicting the class label of the instances in the testset. However, the obtained clusters do not generate classes but this approach utilizes to evaluate the generated clusters as classifiers. Thus, it is important to set the number of the clusters same to the number of the classes to create accurate model that each cluster relates to a class.

Classification via clustering has been explored by some researchers for classifying and mapping students' academic performance using students data (see Table 2.7). López et al. (Lopez et al., 2012) reveal that classification via clustering approach obtained similar result to classification algorithms in predicting student's final grades starting from their participation in online forum. Also, Romero et al. (Romero, Lopez, Luna and Ventura, 2013) have applied the same approach to predict the success and failure from students' participation trajectories in on-line forums. Meanwhile, analysing web log files using classification via clustering method to evaluate the e-learners performance reveals interesting results characterize weak and strong students to help the weak students improve their studies (Hogo, 2010); (Perera et al., 2009).

Table 2.7: Classification via clustering method.

Method	**Attributes**	**Authors**
Classification via Clustering	Demographic data, GPA, core courses grades, lab grades and attendance rates	(Alfiani et al., 2015)
	Student participation in online forum data	(Romero, Espejo, Zafra, Romero and Ventura, 2013)
	Student participation in online forum data	(Lopez et al., 2012)
	Course related data	Aher and L.M.R.J., (Aher and Lobo, 2011)
	Web access logs file	Hogo, (Hogo, 2010)
	Group work logs file	Perera et al., (Perera et al., 2009)

Most of the studies reviewed so far were able to predict the academic performance of the students with high accuracy results. However, the accuracy is not the only measurement for the performance of the classifier. Classification accuracy and model explainability both greatly influence the decision made when implementing a classifier. It is very essential to provide a transparent explanation as how the classification has been achieved and therefore, how the decision has been made by the classifier. A very common solution is observing the attributes to study the correlation between the most predictive educational attributes and performance prediction. However, far too little attention has been paid to exploring machine learning approaches to opening the black box of educational data mining and supporting ethical decision making.

2.3.2 Predictive Attributes on Performance Prediction

A primary concern in predicting the academic performance of the students is the important features that could enhance the learning process of the classifiers. Extensive researches have been therefore carried out on predicting the performance of the students based on several students' attributes and features. These features were identified as one of the most influencing factors of the prediction process. For instance, student-related attributes (i.e. demographic and personal data) were involved in several studies to exploit predictive models for students performance (Bekele and Menzel, 2005); (Al-Radaideh et al., 2006) (Cortez and Silva, 2008); (Yadav and Pal, 2012); (Bhardwaj and Pal, 2012); (Kabakchieva, 2013); (Alfiani et al., 2015). This includes student's category, gender, age, origin and address. Also social and socio-economic data of students were mostly used by researchers for predicting the academic performance of the student such as family size, annual income, parent's status, parent's qualification and occupation (Bekele and Menzel, 2005); (Cortez and Silva, 2008); (Yadav and Pal, 2012); (Bhardwaj and Pal, 2012); (Poh and Smythe, 2014).

In contrast, some studies were involving course and tutor related data as predictive attributes for predicting the student's performance. Course related data identify information about students attendance, core lectures grade and lab grade (Alfiani et al., 2015). While, tutor related data includes information about the tutor of the course such as tutor name, qualification, gender and the affiliated department (Al-Radaideh et al., 2006).

Pre-univierty data was also used in predicting performance such as: past school grades and examination grades (Cortez and Silva, 2008); (Aher and Lobo, 2011); (Poh and Smythe, 2014), and students grades in high and senior secondary schools (Yadav and Pal, 2012). These attributes were rarely used by data mining researchers as they did not obtain very high accuracy for the predictive model compared to other attributes. Study progress was also investigated as part of the attributes used to predict the performance of the students (Bhardwaj and Pal, 2012). Study progress identifies information about students feeling about success or failure, learning difficulties and study skills.

In addition, student interactions and web usage data that recorded in online databases such as the log files are identified as a good predictor for predicting the performance of the

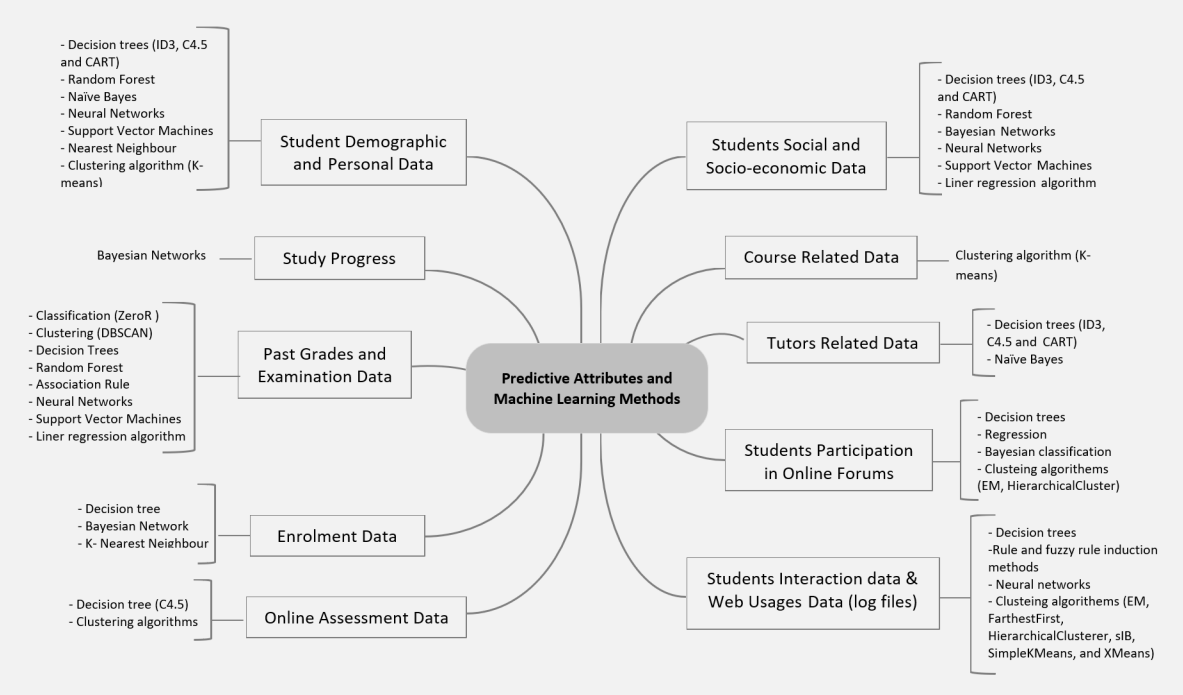


Figure 2.2: Predictive attributes and machine learning methods exploited for predicting the academic performance of the students.

student. Student interaction in the form of their participation in the online forum is analysed by López et al (Lopez et al., 2012) and Romero et al (Romero, Espejo, Zafra, Romero and Ventura, 2013) to predict the academic performance of the students. Whereas, students log files is used to predict the final grades of the students who used Moodle learning management system (Minaei-Bidgoli et al., 2003); (Romero, Espejo, Zafra, Romero and Ventura, 2013). It has been proven that mining student data in the educational systems can improves student overall performance (Aher and Lobo, 2011).

The most predictive attributes and ML methods exploited for predicting the academic performance of the students according to the literature are summarised in Figure 2.2 for reference. As can be seen, student demographical data, social and socio-economic data, past grades and students' interaction data have been widely analysed for predicting the academic performance using different ML methods, such as decision trees, naive Bayes, Bayesian Networks, Support Vector Machines and Neural Networks. Each method was used according to the target of the study, whether to capture learning behaviour, patterns or ascertaining the correlation between

the factors and students' progress. Therefore, the selection of student attributes and machine learning approaches that used in the following chapters was mainly influenced by the literature. The experimental works of the following chapters examine combining educational factors collected at different times with the most influential factors identified in the literature, including student demographical data, social and socio-economic data as well as past grades, so as to provide robust systems.

2.4 Probabilistic Modelling of Educational Data

Probabilistic modelling methods such as Bayesian Networks (BNs) have been recently involved in numerous learning tasks in education due to the capability of such method to infer a graphical probabilistic-based structure. Learning a probabilistic structure enhances the classifiers as the learned models usually capture the dynamic and the nonlinear correlations between the attributes (Heckerman et al., 1995). Characterizing such complex correlations in predicting the academic performance requires exploiting a probabilistic learning approach to infer the predictive models from the educational performance data.

A considerable number of studies have been conducted therefore to predict the academic performance of the students based on the Bayesian approach. For instance, (Kabakchieva, 2013) compared the Bayesian approach with other classification approaches to identify useful patterns that could be extracted from students' personal and pre-university data in order to predict students' performance at the university. Similarly, authors of (Kaur and Singh, 2016) used the same approach for a comparative study of classification algorithms but the aim was to classify the students and identify the most influencing attributes on students' failure.

Bayesian classification method is also applied by (Bekele and Menzel, 2005) to predict students' performance based on values of social and personal attributes. The empirical result revealed that Bayesian network classifier is a valuable method for predicting the students having satisfactory, or above/bellow satisfactory performance. Another Bayesian classification method (in particular Naïve Bayes) modelled by (Bhardwaj and Pal, 2012) to predict the slow and the high learner's students. The study conducted on 300 student records for BCA module (Bachelor

of Computer Applications). The attributes included in this investigation were demographic, academic and socio-economic that obtained from students questionnaire and the database of the university. By means of Naïve classification approach, it was stated that student's performance in university level is dependent on secondary examination grades, students living location, teaching mode and other potential factors such as parents qualifications, students habit, family annual income and family status.

Essentially, Bayesian networks used as a method to characterize the relationships between educational attributes and knowledge predictor attributes such as students end-of-course performance. The probabilistic estimation of the attributes in the BN provides insights for the uncertainty related to the predictors. In modelling students interactions data from a tutoring system, uncertainty involves lack of understanding how knowledgeable is the student and the achieved aims by the student (Conati et al., 2002). This is mainly a result of involving the students in various activities without showing their reasoning for performing such activities. For example, (Conati et al., 2002) explored managing uncertainty in modeling students knowledge assessment data from 'Andes' (tutoring system) using BN. By means of modling students' knowedlge and use this information for predicting students' responses, the obtained results provide that the greater granularity of the predictive model enhances the prediction of students' performance in online systems.

Although some researchers were attempting to use the BNs for characterizing performance and exploring the correlation between the students' performance and the attributes, others were interested in detecting and modeling students' learning styles (García et al., 2007) (Carmona et al., 2008). For example, (Kaur et al., 2015) conducted a study to analyze demographic, social and assessment data to predict the slow learning students in order to improve their performance and reduce failure rate prior to the exam.

However, the work conducted far for the implementation of the Bayesian methods on the educational data has targeted extracting useful patterns for detecting the learning styles of the students or predicting their overall performance. Limited experimental studies have considered the issues that were associated with the educational datasets before learning the predictive classifiers. These issues are including: missing, noisy, imbalanced data and so forth. For instance,

the imbalanced educational data was investigated through exploiting the bootstrap approach (Beal and Cohen, 2005) (McLaren et al., 2004). Feng and co-authors (Feng et al., 2009) utilized and validated their statistical results by using bootstrapping with logistic regression to evaluate students' learning based on different educational interventions. Similarly, a study has been conducted by (Pfannkuch et al., 2013) to evaluate students' understanding of statistical inference with a bootstrapping approach while did not consider time. To the best of our knowledge, there is no previous work in this field that applied BNs on the bootstrapped "time series" dataset of students progression data as this thesis explores. Hence, the work presented in chapter 5 is a first attempt to use them, with the aim of achieving an improvement in student performance overall when exploiting BN.

2.5 Clustering Educational Data

A growing body of literature has investigated clustering students based on their extracted features from educational databases to identify students with similar learning behaviours and patterns (Romero and Ventura, 2010). For this purpose, different clustering algorithms have been applied, such as: K-Means, agglomerative hierarchical and model based clustering algorithms (Romero and Ventura, 2010). All of these algorithms have been implemented to model student learning styles or patterns. For instance, the K-Means clustering algorithm was applied by (Hogo, 2010) and (Perera et al., 2009) on student log files to discover the unknown behaviour patterns of each. Interesting results were obtained, whereby they distinguished weak from strong students, with the aim of helping the former to improve their studies prior to the final exam. The K-means algorithm has also been effectively applied to a group of students with similar learning portfolios for their examination and assignment results (Chen et al., 2007). In another study, which set out to determine the effectiveness of the hierarchical agglomerative algorithm on detecting student learning preferences, Zakrzewska (Zakrzewska, 2008) clustered students based on extracted features from an eLearning system. He proposed a method to group students to determine their preferred learning styles from the information and performance results gathered from an eLearning system.

The notion of clustering students has also been investigated to support collaborative learning. (Talavera and Gaudioso, 2004) clustered students' interaction data in a learning management system to identify their profiles and learning behaviours. The main goal of the study was to evaluate the collaborative activities of an online forum, chat and email. They extracted useful knowledge showing different collaborative profile patterns of each student based on collaboration data. Another study was conducted by (Lopez et al., 2012) to predict the final grades of the students based on their participation in the online forum using Weka mining tool. By means of Clustering algorithms (EM, Farthest First, Hierarchical Clusterer, sIB, Simple KMeans, and X Means) they found that students participation in the course forum is a predictive factor for predicting student final grade in a module. Similarly, clustering students' profiles in an online discussion forum has been also explored by (Cobo et al., 2011) to identify Students with similar behaviour profiles. The clustering approach was performed using the Agglomerative hierarchical clustering algorithm. Hence, to date, clustering students research has been mainly underpinned by the goal of identifying similar learning behaviours and preferences.

However, too little attention has been paid to clustering students based on their time series trajectories for performance prediction purposes. Moreover, most of the previous research on education that considered clustering students has used a selection of extracted features of activity data (not the entire performance sequential trajectory). For instance, (Młynarska et al., 2016) explored students' activity attempts in Moodle, a Virtual Learning Environment (VLE), to extract patterns among students to characterize the struggles and experts. They clustered activity counts of students completed tasks (counts of: courses, assignments, activities) using Dynamic Time Warping, which have been considered as time-series data. Seven behaviour profiles have been identified in this study to help understanding how a large group of student behave when learning and interacting through a VLE such as Moodle. Furthermore, Shen and Chi (Shen and Chi, 2017) grouped students using the Dynamic Time Warping (DTW) algorithm. Their main objective was to capture student learning behaviour patterns so as to be able to offer personalised learning. They investigated the differential impact of DTW, Normalised DTW and the Euclidean distance on clustering moment-to-moment student sequential

trajectories, proving that clustering these trajectories was highly effective for detecting learning behaviour. According to Shen and Chi (Shen and Chi, 2017), the proposed framework involved the first implementation of DTW in the educational field for the detection of student behaviours.

Predicting the academic performance based on students-related attributes as well as some extracted profiles was determined in a recent study by (Al-Luhaybi et al., 2019). A bootstrapping resampling approach was investigated by the authors to accurately predict university students' performance using their time-series data with dynamic Bayesian networks. However, a perspective has been adopted in this paper such that it extends our previous works and argue that clustering students' online assessment trajectories in Year1 using a DTW distance-based clustering algorithm can improve the prediction performance across all modules. Furthermore, we proved that using the entire self-assessment progression trajectories is better than some students selected attributes. Therefore, the application of machine learning algorithms to online time series progression trajectories provide early insights for student outcomes and outperforms existing solutions.

2.6 Unbiased Classification

Ensemble-based methods have been executed in several studies (Street and Kim, 2001) (Tsoumakas and Vlahavas, 2007) Xia et al. (2011) (Tan and Gilbert, 2003) in attempts to provide unbiased and improved classification. For instance, (Tsoumakas and Vlahavas, 2007) provided an example of constructing an algorithm for large-scale classification of streaming data. They obtained a fast algorithm with very accurate classification results by using a replacement strategy. Similarly, the RAndom k-label sets (RAKEL) algorithm was used in (Tsoumakas and Vlahavas, 2007) as an ensemble approach for constructing the classifiers for multi-label classification with consideration of the label correlations. Moreover, (Tan and Gilbert, 2003) investigated another ensemble method by using boosted decision trees with compassion with a single decision tree for cancer classification. Interesting classification results were obtained when using bagged decision trees. A broader perspective was achieved in a study conducted by (Martis et al., 2013)

using an ensemble bootstrapping approach with time-series educational data for unbiased classification with a dynamic Bayesian network. However, adapting the ensemble methodology in classification models may remove some of the good patterns and correlations of the original data. Hence, such models can be characterised new synthetic correlations that might affect their predictive generalisability.

Whilst many studies have involved determining approaches associated with data, others have been focused on considering feature-based approaches, such as dimensionality reduction and feature extraction. For example, principal components analysis (PCA), linear discriminant analysis (LDA) and independent components analysis (ICA) were exploited by (Subasi and Gursoy, 2010) (Martis et al., 2013) as a dimensionality reduction technique to perform signals classification. In particular, the reduced dimension was considered to learn a classifier using a support vector machine algorithm. Apart from these studies as a dimensionality reduction technique with the PCA method (the original linear mapping approach) to reduce the bias with many classification algorithms, other studies were investigating further the nonlinear PCA. A study was conducted by (Hoffmann, 2007), who extracted the principal components with Kernel principal component analysis (kernel PCA) to detect novelty using breast-cancer cytology and handwritten digits data. Another study was conducted by (Kim et al., 2001) which involved using the same Kernel PCA to learn a feature subspace for texture pattern classification. The studies presented thus far, proved the validity of the dimensionality reduction approaches to provide accurate classification results. However, the lack of transparency in the linear and non-linear dimensionality reduction techniques affects the efficiency of the learned classifiers. In particular, it was observed that there were redundant correlations and features. To address these shortcomings, in chapter 6, the advances of deep learning approaches with a BN are examined explicitly to determine the unwanted dependencies for performing unbiased classification. In essence, the aim is contribute to the debate as to when building trust in AI models with outputs that are explainable, reliable and accurate.

2.7 Summary

In this chapter, I presented the machine learning applications and tasks that have been explored in EDM. In particular, I focused of the prediction of students academic performance using learning approaches with different characteristics of student data to construct predictive classifiers. Additionally, I explained probabilistic modelling of educational data with BN as a potential graphical learning method to capture the relationships among the educational attributes and overall achievements of the students. Hence, in this thesis the aim is to exploit robust and explainable machine learning classifiers for the prediction of students academic performance in higher education. This will be achieved via identifying the key issues associated with educational data, learning predictive models with explainable machine learning methods as well as understanding the students underlying dynamics for predicting the performance.

In the following chapter, I exploit some machine learning approaches to model the predictive classifiers of students academic performance. In addition, I explore the key educational attributes and investigate a clustering approach to identify the temporal cognitive styles from students' engagement time-series to enhance the prediction process.

Chapter 3

Preliminaries and Identification of Cognitive Styles

In this chapter, the significance of some ML methods, such as classification, clustering and feature subspace learning approaches, aimed at learning transparent and reliable classifiers for students' performance prediction is examined. This includes initial analysis of the educational datasets to identify the key features affecting the prediction of a student's academic performance using different classification methods. This chapter also provides the method of identification of the temporal cognitive styles from students' engagement time-series aimed at enhancing the prediction process. There is also exploration of feature-subspace learning (dimensionality reduction) techniques, including linear PCA and non-linear PCA with Autoencoder, for improving prediction.

3.1 Predicting Academic Risk of Failure with Classification Methods

Machine learning classification can be used to predict student's performance as well as identify the key features influencing the prediction process. This chapter explores classification algorithms to predict academic failure based on a combination of three major attributes categories.

These are:

i) admission information

ii) module-related data

iii) first year final grades

For this purpose, the J48 (C4.5) decision tree and Naïve Bayes classification algorithms are applied to predict grades of computer science 2nd year students at Brunel University London for the academic year 2015/16. The outcome of the predictive model identifies the low, medium and high risk of failure of students. This prediction will help instructors to characterize and then assist high-risk students by making appropriate interventions. In this connection, this study seeks to address the following:

- Factors affecting the prediction of the high risk of failure of students in higher education institutions and universities,
- Predictive data mining models using classification algorithms based on first year student final grades, modules related data and students' admission datasets.

3.1.1 Data Selection and Pre-processing

This chapter considers student and module data obtained from the Admission and Department of Computer Science databases at Brunel University London, UK. The integrated data considered in this investigation are categorised into three categories, are as follows:

1. Admission Data: relating to students' information when they register at the university such as Student Enrolment Status, Student Route name, Fee Status, Student Mode of studying, Qualification on Entry, Location of Study, previous institution ... etc (see Table 2);
2. First Year Grades: the overall grades for all first year modules that were taken by Computer Science Students:

- Information Systems and Organisations
- Logic and Computation

- Level 1 Group Project Reflection
- Data and Information Assessment
- Software Design
- Software Implementation Event
- Fundamental Programming Assessment;

3. Module-Related Data: the data for the predicted module such as Module teaching mode, Tutor Code, Tutor Name, Student study mode, Assessment type and Absences.

The attributes and the domain values for the selected attributes for the current study are defined in Table 3.1 for reference. As can be observed from this table, the educational attributes for the three categories have been merged into one dataset to exploit the classifiers. The dataset, therefore, includes the full trajectories of admission data, Level 1 final grades and module relate data for the same group of students. These pre-processing steps were fundamental to modelling the classifiers in a way that identifies the key features affecting the prediction of a student's academic performance. As can be seen from Table 3.1, the original educational features include discrete domain values, which can be handled and inferred for the probabilistic reasoning of the predictive models.

A total of 129 student records (instances) for the year 2015/16 are involved in this investigation to develop the predictive model for the prediction of the students at high risk of failure in year 2 modules:

- Algorithms and their Applications
- Usability Engineering
- Software Development and Management
- Year 2 Group Project

The predicted class attribute is Overall Grade, which is the final grade obtained by the student in the targeted module. It has five possible values A: Excellent, B:very Good, C: Good,

D: Acceptable and F: Unacceptable or Fail, which have been merged to Low risk, Medium risk and High risk of failure (see later in Table 3.1) as these were seen to be the most useful distinction between students and also led to most consistently strong results.

I performed steps for the implementation of the classification and clustering algorithms to predict the academic performance of the students for the year 2 computer science core modules using Java API and Weka Mining tool. Students overall grades were merged to reduce the number of classes for the targeted Modules into three classes: low risk, medium risk and high risk of failure based on discussions with academic staff as to what is the most useful distinction between students of making informed interventions. The low risk class is for students who have obtained A and B in the targeted module. Medium risk class is for students who have obtained a C in the module. Whereas, the high risk class represents students who obtained a D - F (see Table 3.2).

Data augmentation techniques would be a useful solution to increase the sample size of the educational data, especially having the datasets include confidential information, such as student personal data. However, this solution might affect the reasoning process when implementing the classifiers, especially given that the decisions made using this type of data will be highly influenced by the synthetic data. The issue of the limited sample is addressed in later chapters by including more labelled data based on student progression trajectories and other educational factors collected in different time slots.

3.1.2 Learning Predictive Classifiers

Classification, a form of supervised learning, is a very common data mining technique that is applied to map datasets into sets of classes (Aher and Lobo, 2011). To develop such models, one or more training and testing phases are required. In the training process, a training subset of data is used to infer a classifier model based on the relation between the selected attributes and the class label. Consequently, the testing process identifies the accuracy of the model by applying the obtained classifier on the test subset (Bhardwaj and Pal, 2012). The machine learning algorithms explored in this study were C4.5 decision trees and Naïve Bayes. This is because of the way that they model the data in a transparent manner using clear tree structured

Table 3.1: Description of the educational attributes and the domain values.

	Attribute	Description	Domain Values
Category 1: Admission Data	**Enrolment Status**	Students enrolment status	{EE}
	Programme Name	Student program name	{UG Computer Science}
	Route Name	The student chosen route	{Computer Science, Computer Science (AI), Computer Science (SE), Computer Science (Digital Media And Games), Computer Science (Network Computing)}
	Route Code	The code of the student chosen route	Based on Rout Code at the University
	Through Clearing	Whether the student enrolled in the same course as the course she/he has applied for	{Y, N}
	Fee Status	Tuition fee status	{Home/EU, Overseas}
	Student MOA	Students study mode	{FT, FSK, FT120, PT80, PT20}
	Detailed Fee Status	Tuition fee status	{Home, European, Overseas}
	Fee	The amount of paid fees	Based on the amount of paid fees
	Gender	Student gender	{M, F}
	Country of Domicile	Student country	Based on Student country
	Age on Entry	The student's age when he/she enrolled at the university	Based on Student age
	Qualification on Entry	Students previous qualification	{Foundation degree, Foundation course at level J, Higher education (HE) access course, A/AS level, Level 3 quals, all are subject to UCAS Tariff, Other qualification at level 2, International Baccalaureate (IB) Diploma, Non-UK degree}
	CRS Code indicates LBIC	Payment method for the course	{Y, N}
	Location of Study	Campus name	Based on Campus name
	Admissions - Core Grades Flag	Indicates admissions decision for registering the student in the course	{Achieved, Predicted}
	Previous Institution	Student previous school or institution	{UK State School, UK Independent School, Any Non-UK Institution, UK Higher Education Institution}
Category 2: Level 1 (1st Year) Final Grades	**Information Systems and Organisations_Grade**	Module Final Grade	{A – Excellent, B - very Good, C - Good, D - Acceptable, F – Unacceptable}
	Logic and Computation_Grade	Module Final Grade	
	Level 1 Group Project Reflection_Grade	Module Final Grade	
	Data and Information Assessment_Grade	Module Final Grade	
	Software Design_Grade	Module Final Grade	
	Software Implementation Event_Grade	Module Final Grade	
	Fundamental Programming Assessment_Grade	Module Final Grade	
Category3: Module-Related Data	**Course MOA**	Module teaching mode	{FT, FSK}
	Tutor 1 Code	The code of the tutor at the university	Based on tutor code
	Tutor 1	The name of the tutor of the Module	Based on tutor name
	Module	Module Code at Brunel University	Based on module code in the university
	MAB_SEQ	Assessment code	{1, 2}
	MAB_NAME	Assessment type	{Unseen Examination, Assessment, Post-Mortem Style Group Review, Assessment of ethical behaviour, Open book in-class Programming Test, Group submission of a design document, Individual viva voce, Programming Ass., Coursework (Practical Assignment)}
	MOA	Student study mode	{full time, part time}
	Supervisor	Student supervisor's name	Based on supervisor name
	Absences	The total number of absences during the semester	Based on Module attendance count
Overall Grade	**Overall Grade**	Student overall grade in the Module	{A – Excellent, B - very Good, C - Good, D - Acceptable, F – Unacceptable}
Overall Grade	**Overall Grade**	Student overall grade in the Module after merging	{Low risk, Medium risk , High risk}

Table 3.2: Class Attribute regarding to student final grades.

Class	Grade Band
Low risk	A, B
Medium risk	C
High risk	D, F

decisions or conditional probabilities. The explanation of these algorithms and how they have been constructed to characterise the students at risk of failure is presented in the following subsections:

- **Learning C4.5 Decision Tree Classifiers**

The C4.5 decision tree algorithm was first implemented by Ross Quinlan (*Is C5.0 Better Than C4.5*, n.d.). It uses a supervised learning approach to classify the data based on measuring a gain ratio. The gain ratio measurement is important for evaluating the splitting process of the attributes based on the sorted data at each node. This process is defined as follows:

$$\text{GainRatio}(\ \boldsymbol{p}, \boldsymbol{T}) = \frac{\boldsymbol{Gain}(\boldsymbol{p}, \boldsymbol{T})}{\text{Splitlnfo}\ (\boldsymbol{p}, \boldsymbol{T})} \tag{3.1}$$

Where SplitInfo is the amount of components at the position p, considering the test value j as follows:

$$\text{Splitlnfo}\ (p,\ \text{test}\ ) = -\sum_{j=1}^{n} P'\left(\frac{j}{p}\right) \times \log\left(P'\left(\frac{j}{p}\right)\right) \tag{3.2}$$

The C4.5 decision trees are then generated using the training datasets (Hssina et al., 2014). Every node of the tree considers one attribute only in which it is best splitting the cases into subsets that fit into one or more classes. The result of the decision tree is then obtained by the gained information from splitting the data. Thus, the educational attribute with the highest result of information gain is considered to be the decision node in that tree.

- **Learning Naïve Bayes Classifiers**

Naïve Bayes is a simple probabilistic learning method, which constructs the classifier based on the assumption that the attributes of a particular dataset are independent (Zhang, 2004) (no correlations between the attributes) given the class attribute value (see Figure 3.1). In Naïve Bayes classification the target is to learn a classifier based on the training dataset that includes class labels. In this experiment, I assume that three labels/classes are included in the class attribute: Low risk, Medium risk and High risk of failing students. The values of an attribute (E) are considered as $(x_1, x_2, , \cdots, x_n)$. Let (C) be considered as the class attribute and the value of this class be (c).

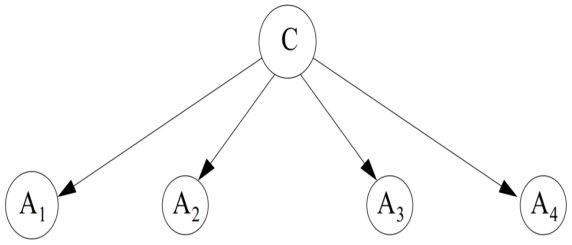


Figure 3.1: An example of a Naïve bayes classifier (Zhang, 2004).

From the probability assumption of the conditional independence of a classifier, the probability of (E) = $(x_1, x_2, \cdots, x_n)$ given a particular class (c) is calculated as:

$$p(c|E) = \frac{p(E|c)p(c)}{p(E)} \tag{3.3}$$

The assumption of the classifier that all the educational attributes are independents for learning the probabilities of a particular class attribute is calculated as:

$$p(E|c) = p\left(x_1, x_2, \cdots, x_n|c\right) = \prod_{i=1}^{n} p\left(x_i|c\right) \tag{3.4}$$

Then the resulting function of Naïve bayes classifier $f_{nb}(E)$ is:

$$f_{nb}(E) = \frac{p(C=+)}{p(C=-)} \prod_{i=1}^{n} \frac{p\left(x_i|C=+\right)}{p\left(x_i|C=-\right)} \tag{3.5}$$

- **Evaluation Measures for the Predictive Classifiers**

The predictive models were based upon level 2 student data from one academic year (2015/16), which resulted in a limited sample of 129 student records. The evaluation results of the predictive models, were obtained using the 10-fold cross-validation (CV) evaluation approach to obtain more robust classifiers. With this learning approach, the educational dataset was randomly divided into 10 approximately equal groups/sets. The first set (in Fold 1) was considered as a validation/testing dataset, whereas the sets 2-10 were used to learn the classifiers. This approach was repeated 10 times on the available data, taking a different validation set in each fold to learn and test the classifiers. The 10 accuracy results obtained from the folds 1 to 10 were then averaged to obtain a single estimation of the overall accuracy.

After each validation process, the mean squared error (MSE) was calculated for the data of the validation fold. This testing process computed MSE1, MSE2, ..., MSEk, the average of which estimated the testing error of the classifier (James et al., 2013).

$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\mathrm{MSE}_i \tag{3.6}$$

Thus, the 10-fold CV, as resampling approach (James et al., 2013) was utilised in this experimental work to improve the skill of the predictive classifiers as well as improving the confidence in the accuracy of the results. The predictive models 'resulting from the learning process' illustrate ways to identify whether the performance of the student is high, medium or low. To evaluate the skills of the predictive classifiers of the educational datasets, three evaluation measures were obtained for the classification performance of each class. These evaluation measurements include model accuracy (ACC), sensitivity (SN) and specificity (SP), which are commonly used in the literature for testing the classifiers' performance (Labatut and Cherifi, 2012). They are defined as follows:

$$ACC = \frac{(TP + FN)}{(TN + TP + FN + FP)} \tag{3.7}$$

$$SN = \frac{TP}{(TP + FN)} \tag{3.8}$$

$$SP = \frac{TN}{(TN + FP)} \tag{3.9}$$

where, TP, FP, TN, and FN represent the cases that are classified as True Positive, False Positive, True Negative and False Negative, respectively. For classifying the performance of the students, for example the high-risk class, TP indicates the number of students that were classified correctly by the classifier and they were actually in the high risk class. FP indicates the number of high-risk students that were miss-classified by the classifier but they were originally in this class (as positive cases). TN represents the number of students in the other classes that were classified correctly, whilst FN indicates the number of miss classified students from the other classes and they were originally labelled from these classes.

3.1.3 Experimental Results

A comparison of accuracy of the selected classification algorithms is provided in Figure 3.2. It can be seen that "Algorithms and their Applications" Module obtained the highest accuracy result in both Naïve Bayes and C4.5 decision tree comparing to other Modules. However, all the predictive models produced very good accuracy results in terms of (69%- 84%).

The analysis of the predictive models is summarized in Table 3.3 and illustrates the comparison of sensitivity (SN) and specificity (SP) of the applied algorithms on different modules. The highlighted probabilities in the following table indicate the high risk class for each specific module. In particular, the probability of correctly detection of high risk failure in "Algorithms and their application" module is identified by the highest sensitivity of 0.969 and 0.938 for Naïve Bayes and C4.5 Decision tree, respectively.

Figure 3.3 presents the best preform C4.5 decision tree model that predicts the students at high risk of failure in Algorithms and their Applications module. Student Overall Grade is the predicted feature in this classification model, and only several features were considered (8 of 33). Interestingly, a remarkable result to emerge from the predictive model is that student

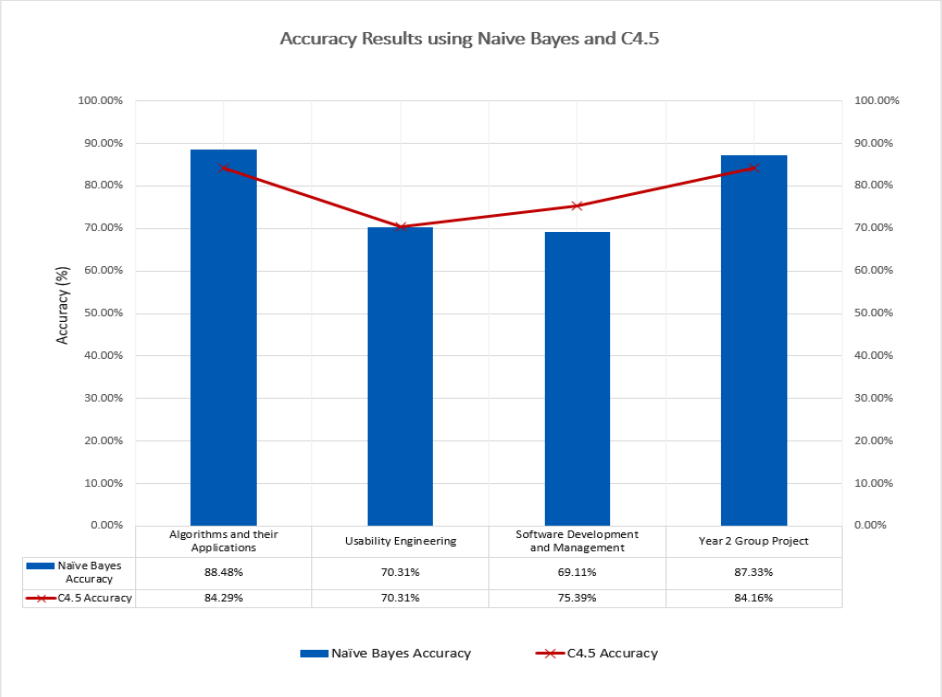


Figure 3.2: Accuracy comparison of the predictive models.

Table 3.3: Sensitivity (SN) and Specificity (SP) comparison of the predicted modules.

Module Title	Class	Naïve Bayes		C4.5 Decision Tree	
		SN	SP	SN	SP
Algorithms and their Applications	Low risk	0.821	0.926	0.776	0.942
	Medium risk	0.867	0.957	0.817	0.933
	High risk	**0.969**	**0.909**	**0.938**	**0.842**
Usability Engineering	Low risk	0.671	0.779	0.714	0.766
	Medium risk	0.192	0.915	0.192	0.915
	High risk	**0.865**	**0.712**	**0.833**	**0.733**
Software Development and Management	Low risk	0.806	0.541	0.921	0.381
	Medium risk	0.379	0.823	0.517	0.872
	High risk	**0.391**	**0,885**	**0.043**	**0.986**
Year 2 Group Project	Low risk	0.732	0.919	0.610	0.879
	Medium risk	0.882	0.921	0.882	0.918
	High risk	**0.920**	**0.982**	**0.902**	**0.953**

qualification has a high impact on the prediction of the high risk of failure students. Furthermore, some of the first year module final grades highly influence the prediction result. These

Modules are Information Systems and Organisations, Logic and Computation, and Software Implementation Event which are the core Modules of year 1 of the computer science program and often involve more technical skills in terms of coding.

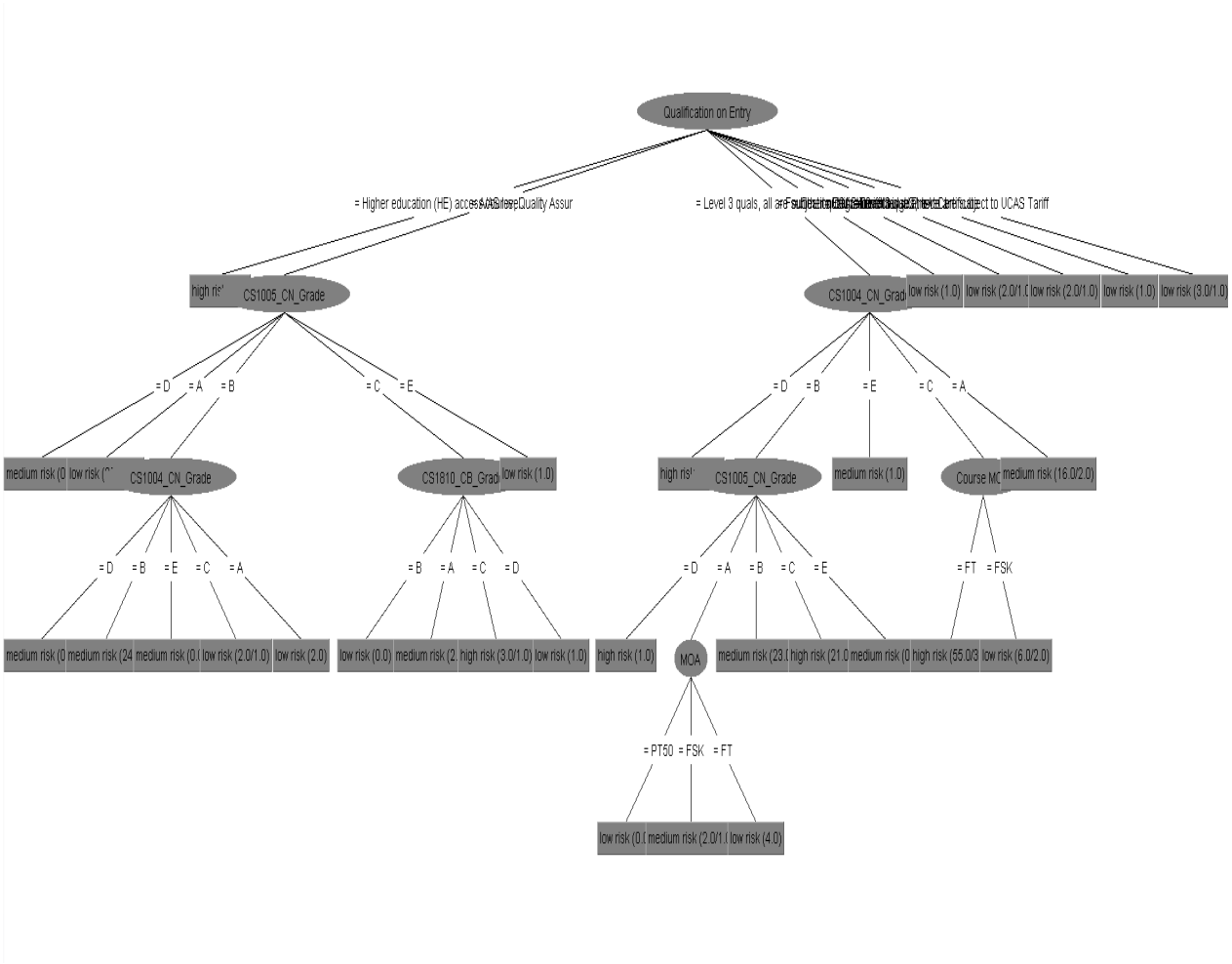


Figure 3.3: Algorithms and their applications C4.5 decision tree output.

Since the C4.5 decision tree algorithm was modeled the educational data in transparent structure with clear decisions as shown in Figure 3.4, some interesting rules can be extracted to predict the high risk students. These rules indicate the influence of student's qualification on their academic performance in Algorithms and their Applications Module, for example:

1. if Qualification on entry = Higher Education (HE) access course then high risk;

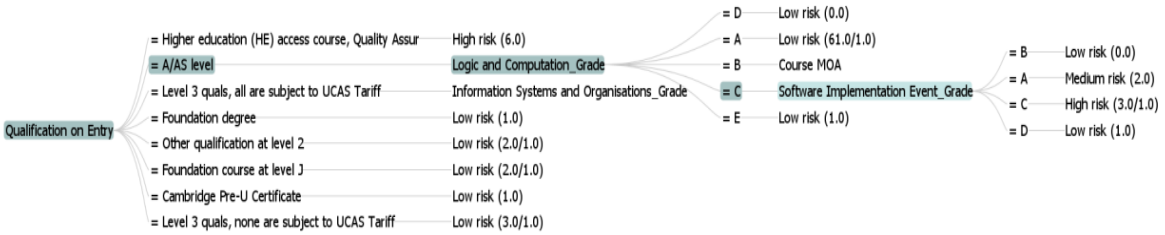


Figure 3.4: Algorithms and their applications C4.5 Prefuse tree output.

2. if Qualification on entry = A/AS Level Λ Logic and ComputationGrade = C Λ Software Implementation EventGrade = C then high risk;

The main goal of this investigation was to explore existing machine learning models for student academic performance to identify the high risk of failure students. The second aim was to identify the key features affecting the predictive model. By applying C4.5 and Naïve Bayes algorithms we revealed that both Naïve Bayes and C4.5 decision trees are promising methods for predicting the students at high risk of failing the module with an accuracy result of 88.48% for Naïve Bayes and 84.29% for C4.5. It also shows that both methods can identify that qualifications on entry have a high impact on students' academic performance. Moreover, some of the first year modules final grades are influencing the results of the students in the second year modules. These findings provide useful insights for the prediction of students' performance which could be influenced by other factors or features. An investigation of other student's features that may influence the prediction process is provided in the following chapters to obtain more accurate and robust classifiers. Moreover, the correlations between the educational features and academic progress will be identified to characterize the most influential features. Thus, the next part of this chapter focuses on extracting temporal cognitive styles from engagement data to improve prediction results.

3.2 Temporal Cognitive Styles Identification

To date various machine learning methods have been introduced to detect students' engagement. The majority focus on behavioural and engagement aspects. Temporal cognitive engagement has not been researched in terms of students' assessments and participation from their online temporal trajectories. The recent increase in using educational systems provides significant benefits to researchers to extract a huge amount of data for students' development, progression, engagement, and learning processes. Analyzing such data can provide useful knowledge on student's styles of perceiving and ways of thinking.

In this section, a distance-based similarity clustering approach was exploited to identify university students' temporal cognitive styles. The proposed method in this section is a temporal clustering method which focuses on clustering students' time-series engagement trajectories based on their attendance at lectures and online tests as well as their progression results for Year 1 and Year 2 courses so early intervention can be provided.

3.2.1 Explanation of the Temporal Engagement Datasets

The main target of the identification of the temporal cognitive styles of a student is to capture students' cognitive patterns that could improve the prediction process. Because of this, time-series educational datasets were used in this experiment to identify students' temporal cognitive styles at different stages of their study, the initial cognitive style at admission level, then at the end of Year 1 and lastly at the end of Year 2. Three different datasets have been included in this exploration, each has 377 students records (for the same group of students) for their achievement during three academic years 2014, 2015 and 2016 as follows:

- The first dataset consists of tutor groups engagement trajectories, which is a track of computer science (CS) students' attendance in a group project module over the 24 week academic term, considered as time-series data.
- The second dataset includes CS students' admission data that has been explained earlier in this chapter when predicting student risk of failure.

- The last dataset contains all Year 1 and Year 2 CS modules final grades. More explanation on the module titles and the domain values is presented in Appendix A.

All these datasets were merged into one dataset to consider the time dependant order of students' progress throughout three academic terms (from 2014 to 2016). Each record in the dataset considers one student's data included based on the time for collecting such data. For example, the first record considers student1 admission data collected at time t, then Year 1 final grades and then, Year 2 grades (as time t+1, t+2, respectively). The datasets were reshaped using this approach to allow for the identification of student temporal cognitive styles at three different stages of their studies.

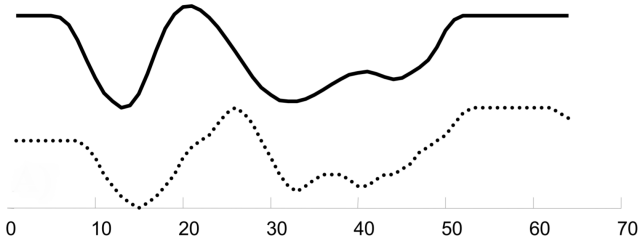
3.2.2 Distance-based Similarity Clustering Methods

For the temporal cognitive style identification, two steps were exploited to generate the clusters. Firstly, there was clustering of students' tutor group engagement trajectories to group students to different clusters based on their engagement. Secondly, students' cognitive styles at three stages were identified at three stages: admission, Year 1 and Year 2.

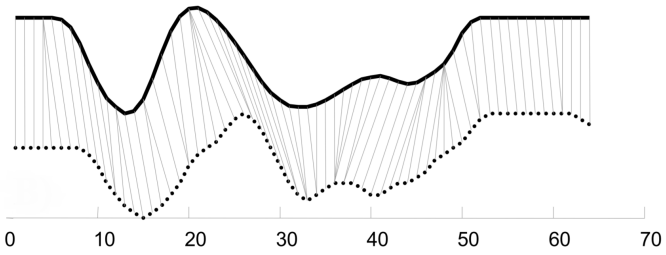
Step 1: Clustering Time-series Engagement (Attendance) Trajectories

The main objective of this step was to cluster students based on their time-series engagement trajectories with the Dynamic Time Warping (DTW) and Agglomerative Hierarchical clustering algorithms. Hence, students who had similar engagement patterns concerning their attendance attitudes were grouped together. This was achieved by measuring the distance between two students' attendance trajectories with DTW to align the engagement trajectories so that the distance was minimised. Figure 3.5 explains the process of finding similarity between two time-series trajectories with DTW, so that a better alignment "warping" for the two trajectories can be achieved, which can be seen in Figure 3.5(B)).

The DTW distance between two students engagement trajectories $x = [x_1, ..., x_M]$ and $y = [y_1, ..., y_N]$ of $M \times N$ dimensions is $D(M, N)$, which it calculates using a dynamic approach, as provided in the following equation:



(a) Two time-series trajectories with a similar trajectory length



(b) Alignment found between the two trajectories after calculating the distances with DTW

Figure 3.5: Utilisation of Dynamic Time Warping (DTW): Figure (a) indicates two time-series trajectories with similar overall sequence shapes, whilst Figure (b) presents a sophisticated alignment after measuring the similarity between the distances with DTW (Keogh and Pazzani, 2001).

$$D(i,j) = min \left\{ \begin{array}{c} D(i-1,j) \\ D(i-1,j-1) \\ D(i,j-1) \end{array} \right\} + d(xi,yi) \tag{3.10}$$

Where, $D(i,j)$ is the local constraint for a given (i,j) node, whereas $D(i-1,j)$, $D(i-1,j-1)$ and $D(i,j-1)$ determine the values of the restricted node. With this calculation, the final DTW distance matrix was generated.

Clustering the DTW distance matrix into five clusters was achieved with the Agglomerative Hierarchical Clustering (AHC) algorithm. This was determined to identify which student

belongs to which cluster and then to observe their temporal cognitive styles.

Step 2: Identification of the Temporal Cognitive Styles

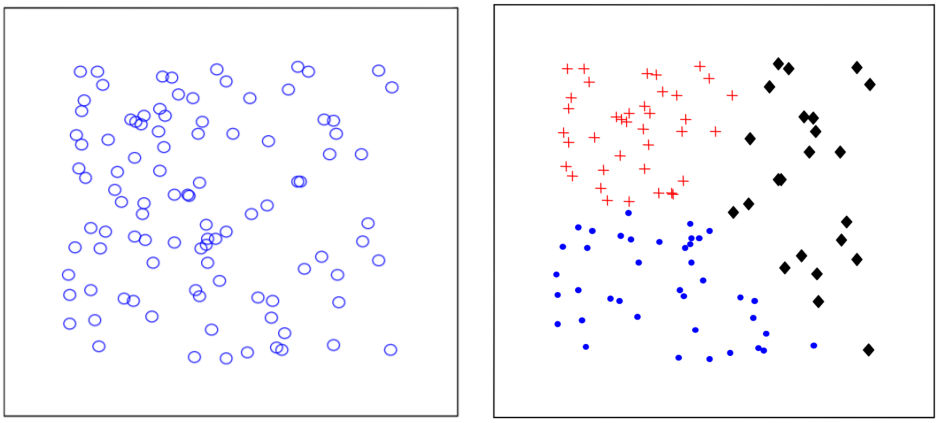
This step was implemented to characterise the different cognitive styles of students. To perform this, the K-Means clustering method (MacQueen et al., 1967) was applied to the DTW engagement clusters identified in the previous step in conjunction with the admissions data along with the year 1 and year 2 grades. This allowed for the identification of students' cognitive style at three different stages: at admission, in year 1 and then, in year 2. These were termed the Initial Cognitive style, Year 1 Cognitive Style and Year 2 Cognitive style. To identify these temporal cognitive styles, firstly, the student admissions dataset was clustered to determine the Initial Cognitive style of the students. Secondly, the clustering method was applied to the admissions data, year 1 final grades and the time-series engagement trajectories (DTW cluster) to identify Year 1 Cognitive Style. Finally, the K-means clustering method was implemented on the admissions data, year 2 final grades and the time-series engagement trajectories (DTW cluster) to partition the students into k groups, thereby identifying the Year 2 Cognitive Style.

The K-means proceeds by setting the k cluster centres and then refining those iteratively to obtain the clusters. This process is conducted as follows (Wagstaff et al., 2001):

1. Assigning each instance (d_i) to the close cluster centre;

2. Updating each cluster centre (C_j) to the mean of instances in that cluster.

An example of the K-means clusters' identification based on real data can be seen in Figure 3.6. Figure 3.6(a) indicates the data points when no clustering was performed. However, in Figure (b), the K-means algorithm tended to assign each data point to the nearest cluster centre with K=3 and hence three clusters were validated based on this data. Setting the value for the optimal number of K was based on the average silhouette statistical testing method using the "NbClust" package in R. Using different values of K, the average silhouette procedure computed the average silhouette of data points to set the best value of K for partitioning the educational datasets. The optimal number of clusters therefore was the number in which it maximised the average silhouette over a variety of k-values, in this case being three clusters.



(a) Dataset with no partition

(b) K-mean partition for the dataset with K=3

Figure 3.6: K-means cluster identification (Jain, 2010).

For performing the k-Means clustering approach on the educational data, five cluster centres were selected for students' data and then, each student was assigned to the closest centre. The statistical testing methods used for estimating the optimal k value (in this study five clusters) can be found in Chapter 4. The centre of each cluster was then updated based on the mean of the students in that cluster. Moreover, the initialisation of the clusters of the temporal cognitive styles was determined randomly, using students' records from the merged datasets. The dataset was discretised into ordinal data (see Appendix A) and the Euclidean distance metric was used to calculate the Hamming distance. Sankey diagrams were used for visualising the flow of students' temporal cognitive styles at a different level. They provide a quantitative explanation about network flows and their relations (Riehmann et al., 2005).

3.2.3 Temporal Cognitive Styles Results

The time-series clustering approach for students' engagement trajectories grouped the students into five clusters. Figure 3.7 illustrates the mean value for students with similar engagements in each cluster. For example, cluster 1 (C1) shows the least engaged group of students in turning up the course during the academic term. Similarly, C2 identifies an average engagement of the students. In contrast, the other clusters 'C3, C4, C5' in this figure provide fluctuated higher engagement attitudes with peaks and troughs at different points in the academic year.

The results of the time-series engagement data showed reliable partitioning for the students' engagement attitude as the DTW metric was based on computing the similarities between the trajectories.

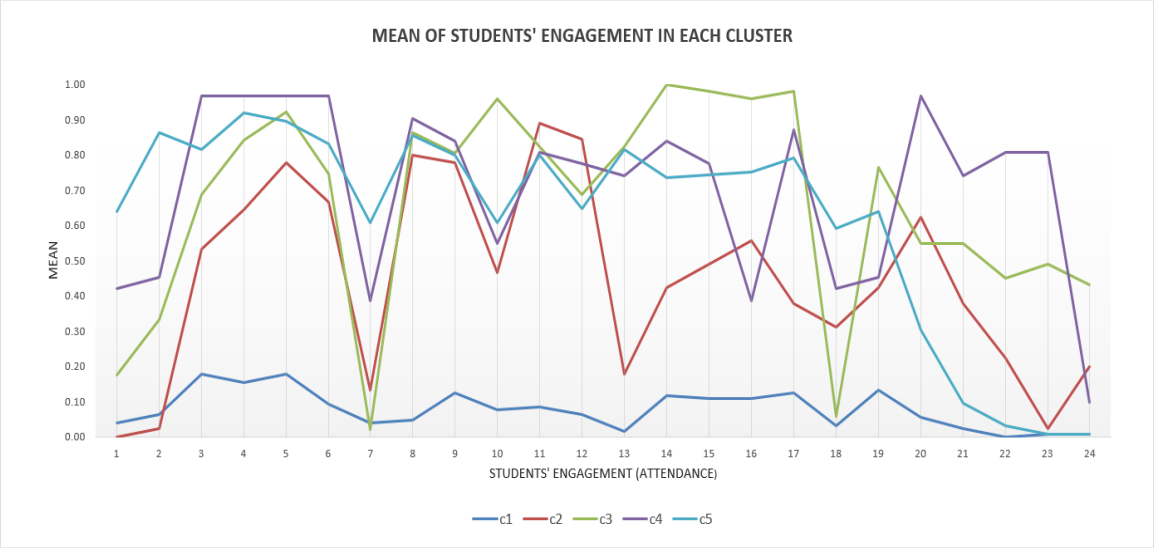


Figure 3.7: Clusters' mean based on student engagement trajectories that been obtained using the DTW and the Hierarchical Clustering methods.

To project the temporal cognitive styles, K-Means was exploited on the identified engagement clusters with DTW in conjunction with year 1 and year 2 final grades. Projecting the clusters with data at different time slots will allow for investigating the temporal changes of students' cognitive attitudes over time. Figure 3.8 provides a visualization of clusters flows that been obtained for the cognitive styles of the students with Sankey diagrams. The figure also indicates the changes in students' cognitive attitudes between the clusters at three levels. These were the initial level when the student enters the university, year 1 and then year 2. These patterns were considered as students' cognitive styles as they were highly impacted by the way the students perceived the information and developed their knowledge in each level of their study through the tutoring and assessments. All these factors (e.g. engagement data, final grades) were considered during the implementation process of the temporal clusters to capture the temporal cognitive patterns of students that could be utilised to improve the prediction of students' performance. For example, the 'Initial cog. style' of the students in 'cluster1' has

been changed in year 1 to another cognitive style based on other factors that were affecting students attitudes such as the engagement trajectories and year 1 grades (see Figure 3.8). In addition, the 'Year1 cog. style' for the same group of students has been changed further to another style called 'Year2 cog. style' based on students' progress in year 2.

The interpretation of the clusters at different academic stages is very important for the investigation of student cognitive style development and how it changes over the study period. The discovered clusters in each year are significant and meaningful and otherwise cannot be discovered easily. They demonstrate changing cognitive patterns for different subgroups.

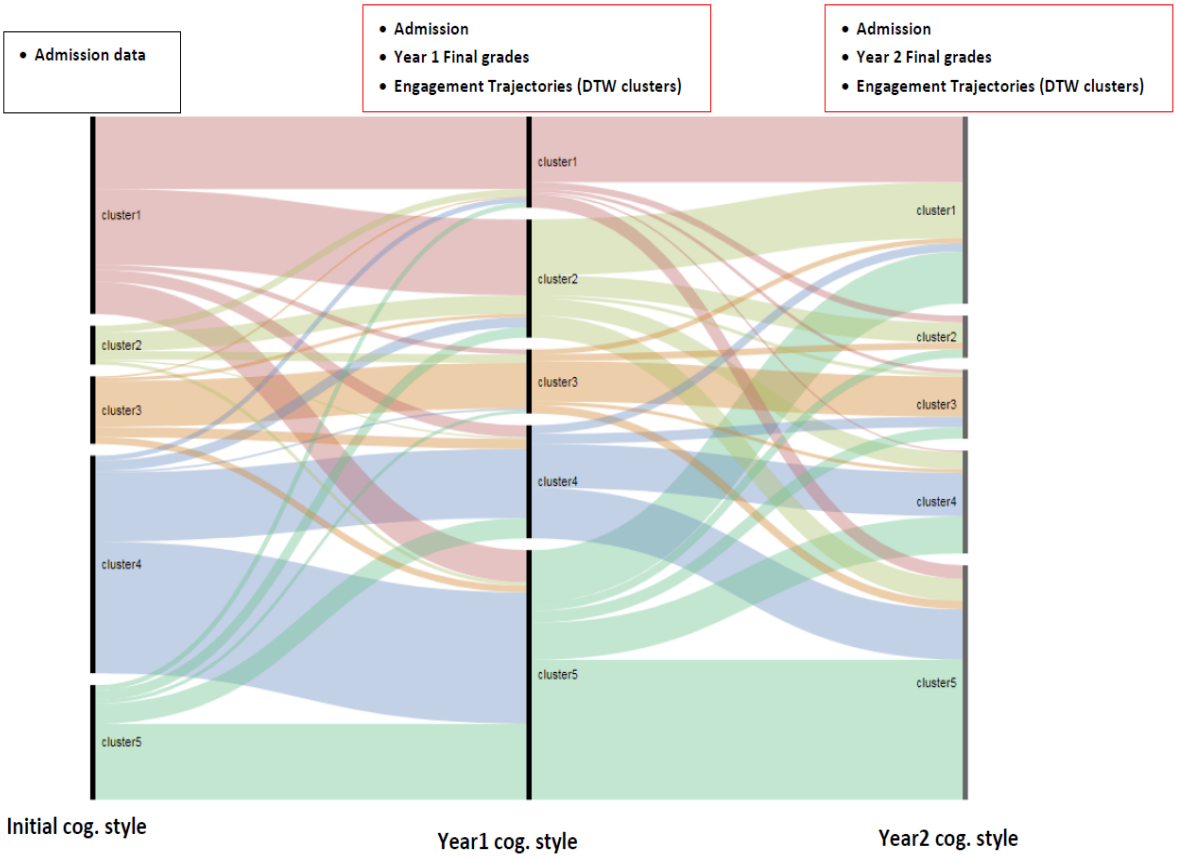


Figure 3.8: Temporal cognitive styles of students.

When discovering the cognitive patterns at different stages, the academic performance of the students in the clusters were also evaluated further based on the mean value of students'

grades in each cluster. In Table 3.4 the clusters were observed to show the mean values of the clusters in year 1 and year 2 based on student grades. Thus, observing these values can explain the performance contrast of the students between the clusters, so we can explain the clusters that include students with low performance. For instance, the mean values for students' grades of the clusters in year 1 were all B, which indicates that these clusters are including students with high academic performance. Interestingly, the mean values of students' grades in Cluster 3 and Cluster 4 in year 2 have been decreased from B to C, this potentially determines the changes in students' performance in year 2 modules. Though the students' distribution for Cluster 3 and Cluster 4 has been slightly reduced in year 2 they capture some useful information about the students' performance. The cognitive styles of these groups of students and their academic performance have worsened in year 2 to medium performance. However, the cognitive styles of the other clusters (e.g. Cluster 1, 2 and 5) in year 1 and year 2 were stable in terms of the performance of the students.

Table 3.4: Cluster mean based on student grades in year 1 and year 2 and students' distribution obtained by the simple K-Means clustering algorithm.

Cluster	**Year 1 Cognitive Style**		**Year 2 Cognitive Style**	
	Year 1 Grades' Mean	**Stu. Distribution (%)**	**Year 2 Grades' mean**	**Stu. Distribution(%)**
Cluster 1	5.76 = B	10%	6.38 = B	26%
Cluster 2	5.91 = B	21%	5.81 = B	9%
Cluster 3	5.94 = B	14%	4.72 = C	12%
Cluster 4	5.98 = B	20%	4.79 = C	19%
Cluster 5	6.15 = B	35%	6.46 = B	34%

The experimental work described in this section has demonstrated how the identification of the temporal cognitive style of students (and how it changes over time) can be insightful. It also indicates that the predicting of students' performance can be improved as will be demonstrated in the following chapters. The findings make an important contribution to the field of Educational Data Mining by demonstrating the influence of the dynamic clustering approaches of students' temporal engagement data for detecting the cognitive style at different stages of their studies. The next chapter, therefore, moves on to show the influence of the student cognitive styles identification on improving their performance prediction. Chapters

5 and 6 examine the identified cognitive styles as independent attributes of overall model performance using probabilistic graphical modelling and deep learning approaches.

3.3 Feature Subspace Learning "Dimensionality Reduction" for Unbiased Classification

In many real-world data, high dimensional data is difficult to interpret as it is in a large-dimension space. Such data impacts learning accurate classification models which typically causes low classification results. One possible solution to tackle these issues is transforming the data into a lower-dimensional subspace using a dimensionality reduction technique (Van Der Maaten et al., 2009). This approach should preserve the proprieties of the original feature-space but in a lower dimension (Fukunaga, 2013). Usually, dimensionality reduction is performed using an unsupervised linear mapping technique such as Factor Analysis (FA) (Spearman, 1904) or Principle Component Analysis (PCA) (Pearson, 1901). For example, in time series data, dimensionality reduction was exploited via a linear mapping in (Yang and Shahabi, 2004)(Keogh and Pazzani, 2000) (Megalooikonomou et al., 2004) (Wang and Megalooikonomou, 2008) (Verleysen and François, 2005) as a measure for detecting similarity. However, the linear mapping of the PCA cannot handle complex data representation as the PCA components possibly lose the nonlinear relations of the original data in the pre-processing step.

Explicitly detecting the nonlinear structure of data is a significant task especially for mining time series data (Scholz, 2012). Nonlinear mapping has been explored to capture diverse processes dynamics of time-series data as for example in (Scholz and Fraunholz, 2008) (Geng and Zhu, 2005) (Herman, 2007). Nonlinear PCA was exploited using Kernel PCA as a feature extraction approach for several purposes such as nonlinear regression (Rosipal et al., 2001), face recognition (Kim et al., 2002), multivariate time series analysis (Yang and Shahabi, 2005), time series prediction (Verleysen and François, 2005) and and classifying video motions (Chan and Vasconcelos, 2007). However, validating the NLPCA models is a very hard task as the over-fitting is usually caused due to the limited number of data trajectories as well as the

main structure of the data. An efficient approach to validate the NLPCA is by comparing the classification results of the NLPCA with the other data representation approaches. For this purpose, a dimensionality reduction technique was investigated in this thesis using the linear PCA and the nonlinear principal component analysis (NLPCA) as pre-processing approaches to learning low dimension subspaces. The nonlinear subspace was learned via an auto-associative neural network or Autoencoder to improve the classification of students' performance.

Firstly, the NLPCA methodology is described for learning a nonlinear subspace using an auto-associative neural network with maximum variance and low square error. Secondly, the experimental results for classifying students using data that has been processed with PCA and NLPCA will be compared.

3.3.1 Subspace Learning with Auto-associative Neural Network

Nonlinear Principal Component Analysis (NLPCA) is a generalization of principal component analysis (PCA). It aims to find a number of components which together explain the variance structure for a dataset (Kramer, 1991). It performs this through a nonlinear 'curve' mapping between the principal components instead of a straight mapping as in the standard PCA (Scholz et al., 2008). Therefore, the learned PC subspace is also adjusted with the nonlinear components and it is curved. The NLPCA can be exploited using a neural network approach with an associative layout called the Auto-associative neural network, Autoencoder or bottleneck (Scholz et al., 2008). This uses a multi-layer perceptron (MLP) (Bishop et al., 1995) to exploit a neural network with identity mapping between the input and output. Consequently, the learned network will be with an identical number of inputs x and output x and output $\hat{x}$ (see Figure 3.9). This achieved by the following equation to minimize the squared error:

$$E = \frac{1}{2}\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2 \tag{3.11}$$

The Autoencoder network learns a layer in the middle called 'bottleneck' with a fewer number of principle components than in the input and output spaces. This layer enforces the reduction of the inputs into a lower dimension space Z_1 ... Z_n, which will then provide

the nonlinear components. The Autoencoder network in Fig 1 represents two parts of two functions: the extraction $\Phi extr : X \rightarrow Z$ and the generation function $\Phi gen : Z \rightarrow X$. Each part includes a hidden layer to perform the nonlinear mapping. For instance, the Auto-associative neural network in Figure 3.9 has a 3-4-1-4-3 architecture with five layers having two hidden and the three units (features) of the input, output and the PCs. The learned autoencoder neural network can obtain more components by specifying additional units in the bottleneck 'component' layer.

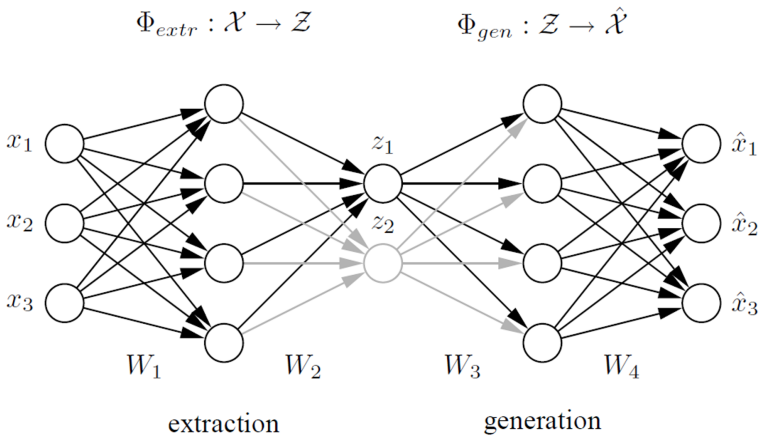


Figure 3.9: Auto-associative neural network (Autoencoder)(Scholz et al., 2008)

For the aim to learn a feature subspace for unbiased prediction of student academic performance. A nonlinear subspace learning with Auto-associative neural network was proposed in this section as well as an explanation of the datasets used for exploiting the experiments.

- **Datasets**

A real-world educational time series dataset was analyzed for evaluating the proposed approach in this section. This dataset includes 377 students progression data for the academic year 2012, 2013,. . . ,2016, respectively. It was collected from Brunel University London admissions and computer science department at the university. Each progression trajectory determines one student achieved grades in Year1, Year2, and Year3 at the university, as well as students' application (admission) data considered as time series data. The total number of

features in the feature space was 34. Here, a three-class problem was determined to predict the overall performance of a student (Low Perfrmance:0, Medium Performance:1 or High Performance:2). These classes were set to explicitly detect the low-performance students based on their overall grades so early interventions could be provided. A detailed explanation of the features and the domain values used in these experiments is presented in Appendix A.

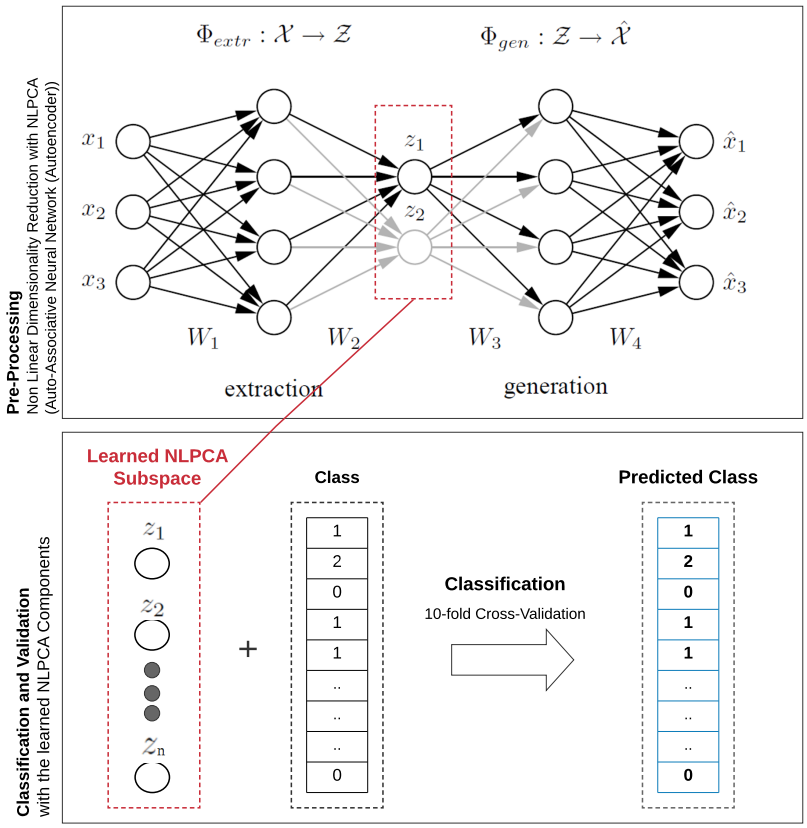


Figure 3.10: Unbiased Classification with NLPCA.

- **Subspace Learning with NLPCA**

To explicitly represent the dependencies between the features, the dimensionality reduction technique was constructed by using the Nonlinear PCA toolbox for MATLAB (HESA, 2019) to extract the principal components in a curved subspace without losing any information from the feature space with the maximize variance for improved classification. Figure 3.10 represents the proposed approach for subspace learning for performing the unbiased classification with NLPCA. This was achieved in two steps: a pre-processing step was undertaken to learn the nonlinear subspace with the auto-associative neural network, followed by the classification and validation step. The NLPCA extracts the non-linear components by determining a random value in each run. However, this approach sometime may miss some of the good dynamics in the data and therefore fails to find an accurate result among them. To avoid this, a pre-processing step was implemented using the linear PCA for better optimizing the extracted NLPCA components. After this pre-processing step, the optimization of the NLPCA will be considered from the linear PCA.

3.3.2 Parameters and K Components Estimation

Determining the optimal number of K components (Z1, ..., Zn in Figure 3.10) in dimensionality reduction is subjective as it is usually influenced by the method used in projecting the learned subspace, as well as the explained variance, to be achieved in the learning process. One possible solution to determine the K components in NLPCA is by testing a different number of K components to find the optimal K for the issue. Another recommended solution is testing different values for the parameter (weight-decay) (*Nonlinear PCA by Matthias Scholz*, n.d.) which can control the complexity of the curve around the leaned components to avoid overfitting and achieve better results. If the datasets have a high dimension input for example 20 or more, it is recommended to pre-process the data with the linear PCA to reduce the dimension for better NLPCA results. Thus, the number of nonlinear K components was determined via performing NLPCA with Pre-PCA and different weight-decay values (0.01 and 0.001). A different number of K was projected to achieve an explained variance of 93% for better comparison with the linear PCA approach.

3.3.3 Subspace Learning Results

In this section, a comparison between the learned subspace results with the linear PCA and the NLPCA with Autoencoder will be presented. To verify the effect of the proposed approach, the classification results of the learned feature subspaces with 'PCA' and 'NLPCA' were evaluated to classify the academic performance of university students. The subspace explained variance for the PCA and NLPCA methods are presented in Figure 3.11. It presented the learned subspaces variances in conjunction with the number of the projected components. From this figure, we can notice that PCA subspace achieved higher explained variance for most of the projected components than the NLPCA. However, the NLPCA obtained better-explained variance results than the linear PCA at dimension 16, with an explained variance of around 93%. Because of this, 93% explained variance has chosen as a measure to test the efficiency of the two approaches for performing unbiased classification. Therefore, the PCA subspace for the 18 principal components (PCs) and the NLPCA via PCA for 16 (PCs) were evaluated to better compare the two methods.

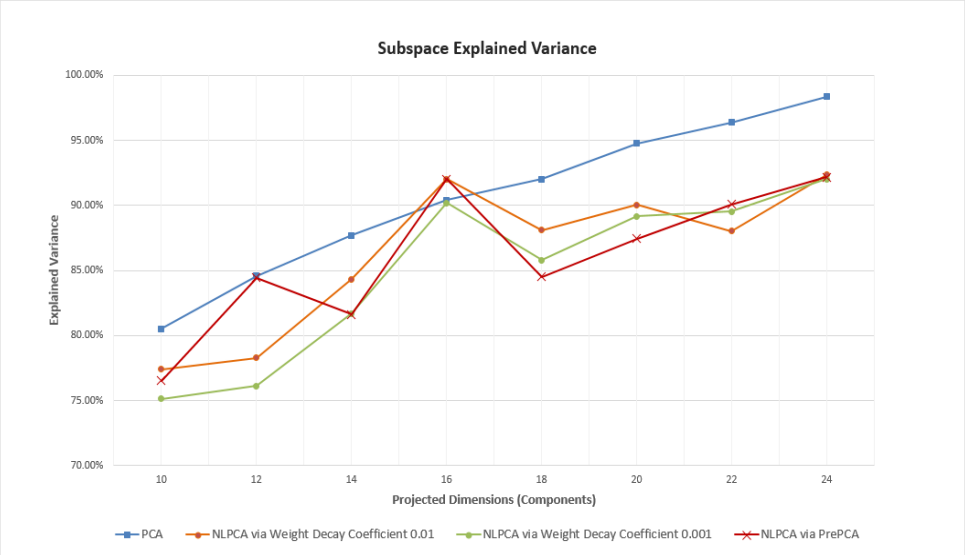


Figure 3.11: The explained variance for the subspaces learned with PCA and NLPCA with Autoencoder.

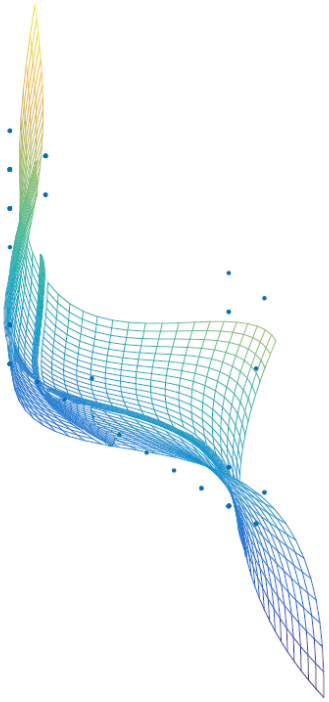
The learned nonlinear subspaces with NLPCA using different initialization is presented in Figure 3.12. These subspaces were obtained according to an explained variance of 93% using

different initialization methods. For obtaining the optimal NLPCA initialization method for unbiased classification, three different methods were examined in this experiment. Firstly, a NLPCA subspace was exploited via random initialization (see Figure 3.12 (a)). After that, the NLPCA subspace in Figure 3.12 (b) was constructed with a weight-decay of 0.001 to control the curve around the non-linear components and avoid overfitting. Lastly, NLPCA was obtained via Pre-PCA with a weight Decay of 0.001 as shown in Figure 3.12 (c). For all previous methods, the learned non-linear subspaces were examined for learning the unbiased classifiers, however, the later classification results show that the best initialization method for learning the NLPCA was via Pre-PCA with a Weight Decay 0.001.

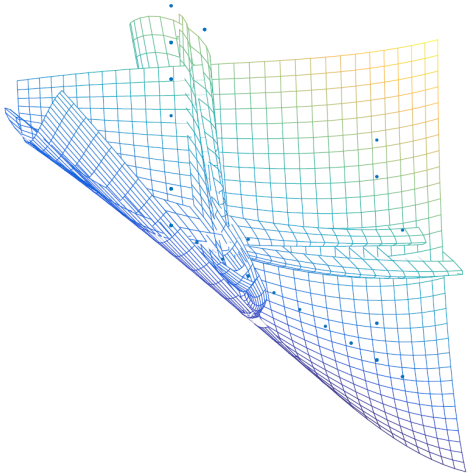
In the linear PCA we can simply explicit the most corresponding features with the PCs by observing the ranked eigenvectors or loadings. Thus, the features with low values can be excluded due to the low dependencies between their loadings and PCs. The loadings of the first 19 PCs (93% Variance) is available in Appendix B for reference. However, as the nonlinear PCs are curved in the NLPCA learned subspace, it is not possible to observe the rank of the features. As the curve between the PCs maps the time-series features and therefore, some features may have a high impact at a specific time point on the curve whereas the others not. In other words, the order of the features is depending on the time on a particular time point on the curve.

3.3.4 Classification and Validation Results

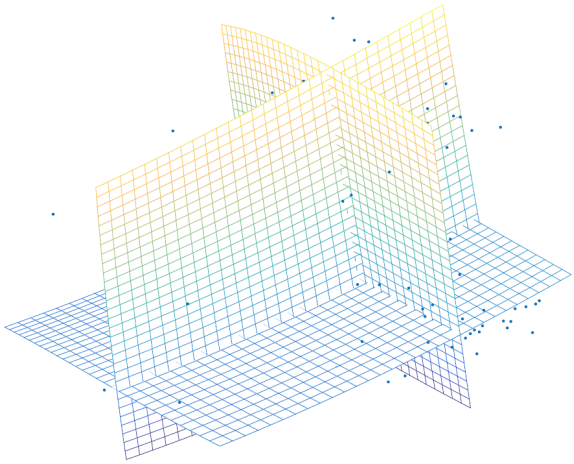
The main issue addressed in this experiment is the biased classification with time series educational data. An intuitive solution was proposed here to explicitly characterize the unwanted correlations between components with NLPCA. Therefore, several experiments have been conducted to learn unbiased classifiers with PCA and NLPCA using different machine learning algorithms. The classification accuracies were determined as a measure for the validation of the proposed approach in this work. Figure 3.13 compared the classification accuracy for classifying the class attribute with the original data and the learned subspaces form the Linear PCA and the NLPCA via Autoencoder. As shown in this figure the best accuracy results were obtained when learning all classifiers from the NLPCA subspace with all the classification algorithms.



(a) NLPCA via random initialization



(b) NLPCA via Weight Decay 0.001



(c) NLPCA via Pre-PCA

Figure 3.12: Learned nonlinear subspaces with NLPCA using different initialization.

Furthermore, there was a significant improvement for classifying the students especially with Bayesian Networks (BNs) and J48. With BNs the accuracy was increased dramatically from

75% to around 90%. This improvement was achieved due to the maximization of the variance of the dataset with the Autoencoder.

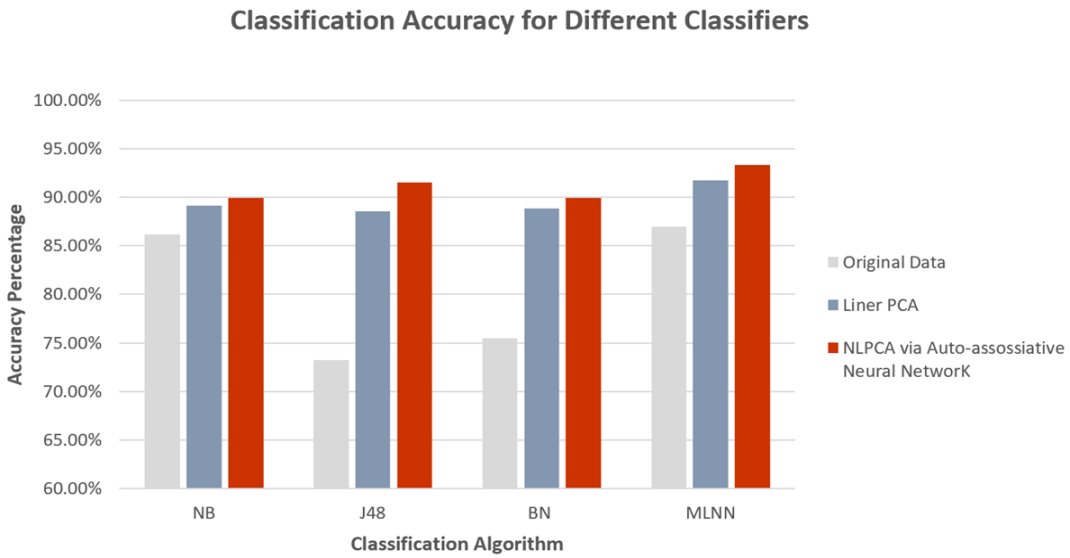


Figure 3.13: A comparison of the classification accuracy for classifying the class attribute with the original data and the learned subspaces form the Linear PCA and the NLPCA via Autoencoder.

The classification results of the learned 'NLPCA' subspace using different classification algorithms is provided in Table 3.6. It includes the classification accuracy as well as a detailed evaluation matrix of each class 'Low risk, Medium risk, High risk'. As we can observe from this table, the TP rate shows significant classification outcomes where the predictive models correctly identify the positive cases in each class. In particular, the High risk classes obtained significant TP values in terms of 0.94 with NB to 0.97 with MLNN. Further evaluations were considered in this table to evaluate the classifiers' performances with precision, recall and F1 measure for the three classes. The precision values indicated the proportion of the students where the classifier consider them in the class and they were actually within the class. All classifiers obtained very high precision values in terms of 0.85 and over. Also, the recall values show the ability of the classifiers in identifying the relevant students in each class.

To conclude, learning feature-subspaces with NLPCA is a very flexible approach. It can be implemented for any real-world problem to improve the classification results. Here, a nonlinear subspace learning approach was investigated using the auto-associative neural network. The

Table 3.5: Classification results obtained by the learned 'NLPCA' subspace with Autoencoder.

Classification Algorithm	Accuracy (%)	Class 'Student Performance'	TP Rate	FP Rate	Precision	Recall	F-Measure
Naïve Bayes (NB)	**89.92**	**Low risk**	0.836	0.013	0.938	0.836	0.884
		Medium risk	0.881	0.083	0.856	0.881	0.869
		High risk	0.941	0.067	0.919	0.941	0.930
J48-Decision Tree	**91.51**	**Low risk**	0.836	0.013	0.938	0.836	0.884
		Medium risk	0.881	0.054	0.902	0.881	0.891
		High risk	0.976	0.072	0.917	0.976	0.946
Bayesian Network (BN)	**89.92**	**Low risk**	0.836	0.013	0.938	0.836	0.884
		Medium risk	0.881	0.083	0.856	0.881	0.869
		High risk	0.941	0.067	0.919	0.941	0.930
Multi-Layer Neural Network (MLNN)	**93.36**	**Low risk**	0.808	0.010	0.952	0.808	0.874
		Medium risk	0.956	0.062	0.896	0.956	0.925
		High risk	0.970	0.034	0.959	0.970	0.965

main idea of the NLPCA is that the learned subspace is adjusted through a curve mapping as an alternative approach of a straight mapping as in the linear PCA. Thus, the NLPCA maximizes the variance between the learned principle components and reduces the squared error. In the area of predicting academic performance, the learned classifier can miss some of the good dynamics between the students which will usually cause biased classification. Thus, a subspace learning approach with the NLPCA was exploited to explicitly represent the dependencies between the space features for improving performance prediction. Although the classification results have been improved with the Autoencoder, it is difficult to explicitly capture the influential educational features. In other words, the Autoencoder does not enhance the explanation of outputs. Further research works are conducted in later chapters to provide transparent and explainable classifiers with graphical modeling and Deep Learning approaches.

Chapter 4

Temporal Profiling of Online Self-Assessment Performance Trajectories

This chapter provides a novel approach for temporal profiling of online self-assessment trajectories using a distance-based similarity clustering approach. This approach was aimed at identifying factors that influence the prediction of students' performance, in particular, the high-risk group of students as well as learning accurate and reliable classifiers for the prediction of student's overall performance.

4.1 Introduction

Predicting students' performance is an increasingly important task within the field of education. However, many aspects have to be considered to produce more accurate predictive models and thus, be able to provide the appropriate form of intervention so as to meet the needs of each student. Traditionally, the performance of a student is measured using final course grades. These final grades are usually based on assessment grades, course activities and final exam results (bin Mat et al., 2013). However, new developments in educational technologies and

Web-based educational systems have been emerging, which have the potential to record data that can be analysed to help students and enhance their learning outcomes. For instance, education-based systems can generate large amounts of data regarding students, instructors, courses, departments, and so forth (Baker et al., 2010). Moreover, electronic educational systems (eLearning systems), such as Blackboard Learn, D2L and Moodle, can help to analyse a student's performance from log files containing data on how many times a student logs into the system or visits a specific content area of the course. These systems also have the potential to record detailed data for each student submission to a specific assessment, such as the time spent answering the question and whether or not their answer matches the correct one (Michalski and Michalski, 2004).

The assessment of a student's performance is strongly reflected in many educational systems, where the student is given a chance to take exams online in order to assess his/her understanding for each module. This particular type of assessment is called online self-assessment. The term "self-assessment" is widely used in education and involves students evaluating their performance in provided questions or tests "with answers being either right or wrong" (Boud and Brew, 1995). It is an important part of the learning process and many online tools are now available for conducting this type of assessment. Perhaps the clearest advantage of online self-assessments is that they are marked automatically allowing instructors to monitor the performance of the students based on their results, thus not burdening the assessor (Ibabe and Jauregizar, 2010). Another advantage of self-assessment in the learning process of the students is that it can determine students' prior knowledge and their understanding of the course (Challis, 2005). These properties can provide information on how a particular student is performing on a course over time. However, capturing signs on the high-risk group of students in the online systems is not available in an appropriate form for the instructors, so the low-performance students cannot be categorized and compared among their class to enable appropriate interventions. As such, the online self-assessment tools have a real challenge in categorizing the high-risk students as well as providing an intelligent representation tool that stating each student's current assessment profile type among his/her class. Thus, predicting student performance based on data generated from educational systems enables understanding

of what students may need to improve, based on appropriate predictive models (Merceron and Yacef, 2005).

Educational data mining (EDM) research has consistently shown that data mining algorithms and techniques are important components in the prediction process (Hämäläinen and Vinni, 2011), as they play key roles in providing accurate predictive models. Extensive research has been carried out on identifying student performance using different classification algorithms (Romero, Lopez, Luna and Ventura, 2013); however, most studies in the field of predicting students' performance have only focused on classifying students based on specific selected attributes, not entire performance trajectories. This was a major reason for investigating this new research area on Educational data mining: applying machine learning algorithms on time-series data means that early insights can be provided for students at risk.

In this research chapter, the major goal is the clustering of online self-assessment trajectories as identifying factors that influence the prediction of students' performance in particular the high-risk group of students. To this end, a novel approach was presented for predicting students' performance by profiling students from temporal online self-assessment trajectories. Specifically, students' online self-assessment trajectories were clustered using a Dynamic Time Warping algorithm (DTW) and use this as input to classification algorithms for predicting student outcomes. This investigation makes a major contribution to research on EDM by identifying and validating the different categories of student performance profile from the online self-assessment trajectories based on a DTW distance-based similarity clustering approach. Also, the generalisability of such approach can be determined to be applied to different class modules.

The overall structure of the chapter takes the form of four sections, including this introduction. The proposed predictive approach with a detailed description of the temporal clustering method used for grouping students' trajectories is provided in section 2. The experimental results are then presented and discussed in the following section in which it includes the resulting online assessment profiles with an evaluation and validation of the profiles on students' performance. In the summary, the important findings of the temporal profiling of students' online self-assessment trajectories using DTW and Hierarchical Clustering algorithms are provided,

particularly for indicating the influence of learned online profiles on performance prediction.

4.2 Methodology

The proposed approach for improving prediction of students' overall performance is based on temporal profiling of their online self-assessment trajectories using the DTW and Hierarchical Clustering Algorithms. Subsequently, I involve the students' identified assessment profiles (the DTW clusters) in conjunction with student admission data to improve prediction of end-of course grades for the students.

For the purpose of the temporal profiling of students online self-assessments, two processes are applied to produce and then validate the resulting profiles (see Figure 4.1), as follows:

- **Temporal Profiling Process.** This determines the profiling of students based on online self-assessment trajectories using the DTW algorithm and Hierarchical Clustering Algorithms;

- **Prediction and Validation Process.** For this, the C4.5 Decision Tree algorithm is applied to student datasets to observe the influence of profiling students' temporal assessments (DTW Clusters) on improving their performance prediction. The obtained predictive model is then tested to predict the performance of the students in other modules. To do this, their attributes in the resulting decision tree are analysed so as to predict performance.

For the implementation of the proposed approach, MATLAB was used to calculate the DTW (the distance-based matrix) and then, generating the clusters using the Hierarchical Clustering algorithm. Subsequently, the GraphViz Package on Java API was used for visualizing the resultant C4.5 decision tree.

4.2.1 Data Preparation and Collection

The datasets used in this investigation were collected from two databases: the online database of the Learning Management System of Brunel University, London, namely, Blackboard Learn

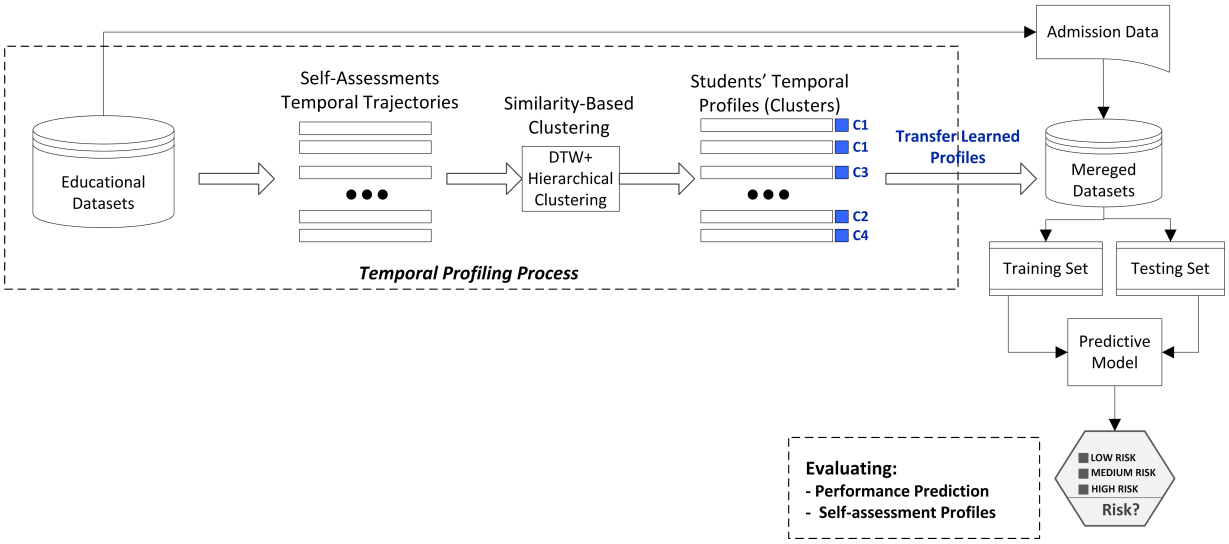


Figure 4.1: Temporal profiling of students' online self-assessment trajectories

and Brunel University admissions information. All the collected data were merged into one dataset.

The first dataset consisted of 329 student records for the online self-assessment exercises for a Logic and Computation Module (Module code. CS1005), taken by students in the 2013/14 and 2014/15 academic years. The online self- assessment dataset contains temporal data that represents students' overall grade for each assessment over time throughout the academic term. During each term, the students were asked to take an online assessment after each lecture to enhance their learning experience and to evaluate themselves. All student grades were recorded in Blackboard Learn. There were a total of 23 self-assessment attributes considered in this investigation, which were used to cluster student trajectories throughout the term to observe student academic performance. The attributes that were used in this set were: Self-Ass.1, Self-Ass.2, ... , and Self-Ass.23. Each contained a student grade on that specific self-assessment.

The second dataset included student application data when they registered at the University, including such as: student demographics, previous educational institution, parent education level etc. The selected attributes and the domain values for the current investigation are provided in Table 4.1 for reference. A total of 329 student instances of two years datasets (for

the academic years 2013 and 2014), for tracking the same students of the students of the online self-assessments dataset, were included in this study to develop the predictive model for the academic performance of the students in all Year 1 modules.

Table 4.1: Student related attributes.

Attribute	**Description**	**Domain Values**
Title	Student title	{MR, MISS}
Route Code	The code of student chosen route	{G400USCMPSC1 for Computer Science, G400USOFENG1 for Computer Science (Software Engineering), ..., etc}
Nationality	Student nationality	{British national, Belgian, Bulgarian, ..., etc}
Ethnicity	Student ethnicity	{Arab, Asian, White, White-British,....., etc} based on Student's ethnicity
Country of Domicile	Student country of domicile	{England, Greece, Spain, ... , etc} based on student's country of domicile
Country of Birth	Student country of birth	{England, Greece, Spain, ... , etc} based on student's country of birth
Level	Student level	{1 for Undergraduate, 2 for Postgraduate}
Socio Economic Class	Student socio economic class	{1-Higher managerial and professional occupations, 2-Lower managerial and professional occupations, -Not Classified, ... , etc}
Been In Care	Whether the student has spent any time in care	{DATA UNAVAILABLE, No, Unanswered}
Parents Been In HE	Whether the student's parents have been to any higher education institution	{Yes, No, Do not Know, Prefer Not To Say}
Sum of Self-Assessments	Sum of grades across all self-assessments that were attempted	{Numerical values}
Temporal Profiles	The resulting clusters from the DTW and hierarchical clustering process	{C1, C2, C3, C4, C5}
CS1005_Grade	Logic and Computation – final grade	{A: Excellent, B: Very Good, C: Good, D: Acceptable, F: Unacceptable}

The predicted class attribute was CS1005 Grade, which refers to the final grade obtained by the student in the targeted module. It had five possible values, A: Excellent, B: Very Good, C: Good, D: Acceptable and F: Unacceptable or Fail, which were later merged to low risk, medium risk and high risk of failure (see Table 4.2) to improve the classification results. More information on the class integration process is provided in the pre-processing section.

The Logic and Computation Module (CS1005) was chosen for the identification of students' performance for the following reasons. This module is a foundation module for year 1 computer science (CS) students, which provides the fundamental concepts of computation

and programming. By the end of this module, CS students should have a specific aptitude in programming to be prepared for the next levels. Also, CS1005 is a blended module that combines online activities and assessments through an eLearning system, Blackboard Learn, with traditional class lectures. Thus, analysing students' grades in a fundamental module provides valuable insights for students' profile and their future performance. However, when modeling a predictive classifier for predicting the performance of the students in year 1, the university database is only including the students' admission/application attributes. Because of this, a decision was made to learn the predictive classifier based on these attributes in conjunction with the learned temporal profiles from CS1005 (the proposed approach in this chapter).

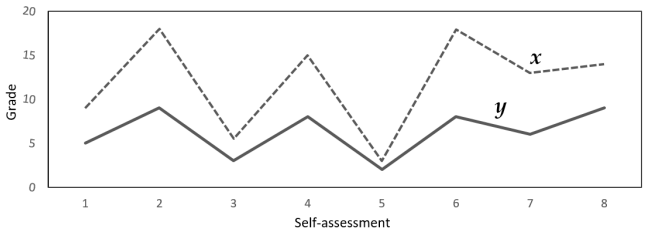
4.2.2 Temporal Profiling of Students' Online Self-Assessments

The main objective of this process was to cluster students based on their trajectories in the online self-assessment to profile students who performed similarly in respect to their online self-assessment results. To generate these profiles or clusters of students, a two-step process was implemented as follows:

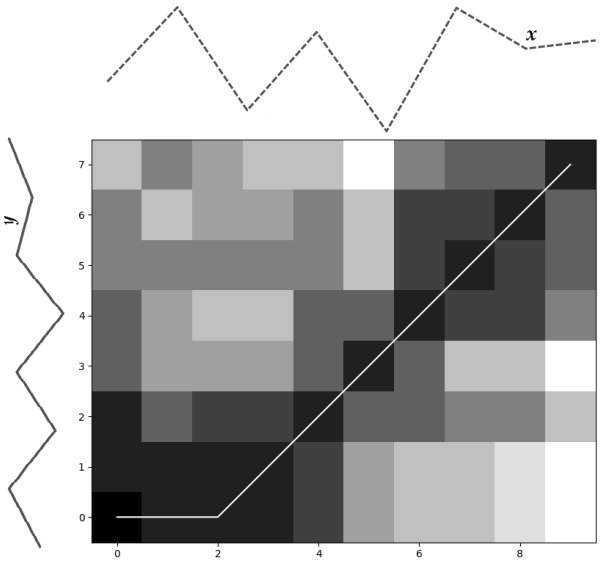
Step1. Calculate the DTW Distance

The first step was to calculate the distance matrix by applying the DTW method (Keogh and Ratanamahatana, 2005) to measure the similarity between each pair of students' self-assessment results over the duration of the module. Suppose that we have two student trajectories x and y of length M and N for the self-assessments, respectively, where $x = [x_1, ..., x_M]$ and $y = [y_1, ..., y_N]$ (see Figure 4.2(a)). The dimension of the DTW distance matrix between these two trajectories is $D(M, N)$ which is shown in the matrix in Figure 4.2 (b).

For measuring the similarity between two trajectories, the distance matrix is calculated using a dynamic approach to align the trajectories to a wrapping path, as provided in the following equation:



(a)



(b)

Figure 4.2: Utilisation of Dynamic Time Warping (DTW) on the self-assessment temporal trajectories: (a) two temporal self-assessment trajectories x and y , (b) the resulting DTW matrix with a "warping" path (the highlighted white line) of the alignment between the two trajectories with DTW.

$$D(i,j) = min \left\{ \begin{array}{c} D(i-1,j) \\ D(i-1,j-1) \\ D(i,j-1) \end{array} \right\} + d(xi,yi) \tag{4.1}$$

where, $D(i,j)$ is the local constraint for a given (i,j) node, whereas $D(i-1,j)$, $D(i-1,j-1)$

and $D(i, j-1)$ determine the values of the restricted node. Using this calculation, we obtained sets of distance values for each pair of student trajectories, which are then been computed to generate the final DTW distance matrix.

Step2. Clustering using Agglomerative Hierarchical Clustering

The second step was to cluster the DTW distance matrix into clusters by applying the Agglomerative Hierarchical Clustering (AHC) algorithm. This step was considered to understand the behaviour of the obtained clusters and reveal which student belongs to which cluster to then observe their overall performance in the targeted module. To achieve this, the DTW-based distance matrix was clustered using different linkage methods (single, complete, weighted and average) to test the efficiency of each on the produced clusters. These linkage methods determine how to cluster student performance in the assessments based on the distance value between the clusters as well as deciding the number of them being looked for. However, the best clusters were obtained via the complete linkage method, which showed an adequate correlation between each cluster and student performance profile.

4.2.3 Data Pre-processing

The improvement in the prediction of student performance was influenced by the pre-processing techniques applied to solve the misclassification and the class imbalance issues on the original dataset. In this section, the proposed pre-processing techniques utilised in this study will be explained to achieve better accuracy and prediction results for the targeted modules.

4.2.3.1 Class Attribute Transformation

Class attribute construction is a very popular data transformation technique for the pre-processing stage, which is usually applied to reduce fundamental issues arising from inadequate attributes. The main goal of attribute construction is to produce a high level attribute from the original attribute to improve the representation of the data space (Hu, 1998). Kampouridis et al. (Kampouridis and Otero, 2013) noted that the original attributes can be merged using attribute construction to provide more predictive ones.

In this study, attribute construction was applied to the class attribute to minimize the number of classes and improve the prediction result. Since the number of classes of student final grades was large with five possible values (A, B, C, D and F) affecting the performance of the predictive models, student final grades were merged. This also made more sense to identifying at-risk students. Table 4.2 shows the proposed 3 class approach, which identifies new class values of the performance of students based on their final grades. Also, some statistics on the number of students in each class, and their overall percentage was included. From this table, it is apparent that the majority class attribute was the low risk class with 70% of the entire population of students and so class balancing is explored in the next section.

Table 4.2: Class Values regarding to student final grades and the number and percentage of students were in each Class

New Class Values	Original Class Values	Number of students	Students Percentage
Low risk	A and B	233	70.22%
Medium risk	C	64	19.45%
High risk	D, E and F	32	9.72%

4.2.3.2 Class Balancing with SMOTE

The Synthetic Minority Over-sampling Technique (SMOTE) is a technique that oversamples the dataset to solve the imbalance issue of the class attribute. This is a well-known technique in data mining research due to its benefit of increasing the predictive accuracy for a specific model (Chawla et al., 2002). Since the student dataset was not large (329 records), I encountered a class imbalance issue. To solve this, SMOTE was applied to the minority classes, which were the high risk and medium risk classes (see Table 4.2) to over-sample the dataset. Therefore, the performance of the model was improved by adding synthetic instances to the minority classes. These synthetic instances were randomly inserted into the minority class depending on the K-nearest to the minority class.

Five nearest neighbours were used for each minority class (high and medium risk). Furthermore, the percentage of the oversampling size for the minority class was 300%. Oversampling

the minority classes by this percentage was determined to obtain an approximately equal balance with the low risk class (see Table 4.3). The resulting new synthetic data points were created in the following way:

- Computing the difference between the minority class instance (a student) under consideration and its nearest neighbour
- Multiplying the result by a random number between 0 and 1.
- The obtained result of the previous step is added to the minority class instance under consideration to create a new synthetic instance, as defined in the following equation:

$$x_{new} = x_i + (\hat{x}_i - x_i) \times \delta \tag{4.2}$$

Where x_{new} = the new synthetic data point, x_i = the minority class instance, $\hat{x}_i$ = the nearest neighbours to x_i and δ = random number [0,1]

Table 4.3: The number and the percentage of students in each class after SMOTE

Class	Number of Students	Percentage of Students
Low risk	233	37.76%
Medium risk	191	30.96%
High risk	192	31.19%

4.2.4 Prediction

Predictive machine learning has been used before to model student performance (El-Halees, 2009), (Romero, Lopez, Luna and Ventura, 2013), (Romero, Espejo, Zafra, Romero and Ventura, 2013). Here the C4.5 decision tree algorithm is used as the aim was to test the influence of the DTW clusters in the prediction of student performance and therefore a transparent model where the features can be examined is needed. Since the datasets were not large the best performing predictive model was identified by firstly using 10-fold cross validation on the

original datasets without any identified student cognitive clusters. Next, the DTW generated clusters (students' profiles) were added to the dataset to determine whether the identification of students' assessment profiles improved the prediction of student performance, as proposed in this investigation.

4.3 Experimental Results

The primary objective was to investigate whether the proposed approach of profiling students' online self-assessments using DTW could provide better prediction of student performance than has previously been achieved. In this section, an evaluation of the identified profiles is presented along with an explanation of student assessment profiles from clustering the online self-assessment results.

4.3.1 Profile K Estimation

Identifying the optimum number of clusters or profiles is subjective as it depends on the parameters and the measures used for partitioning the dataset (Kassambara, 2017). In this investigation, the elbow, average silhouette and gap statistic methods were all explored to determine the ideal number of clusters. The "NbClust" (Charrad et al., 2014) package in R was used for the implementation of all the statistical indices. A visualization of the results obtained from the elbow, average silhouette and gap statistic indices methods is shown in Figure 4.3.

From this figure, it is apparent that each statistical method has given a different result for the best K cluster. For example, the optimal number of clusters using the silhouette method was three clusters or profiles based on the average silhouette width between the students. Whereas, five clusters were delivered from the elbow method and two clusters for the gap statistic technique. Given this lack of agreement, I performed tests on all the 30 indices provided in NbClust package to identify the ideal number of clusters based on the "majority rule". Figure 4.4 illustrates that the optimal number of clusters is five based on the majority rules results among all the statistical indices (30 indices) that were tested on the educational

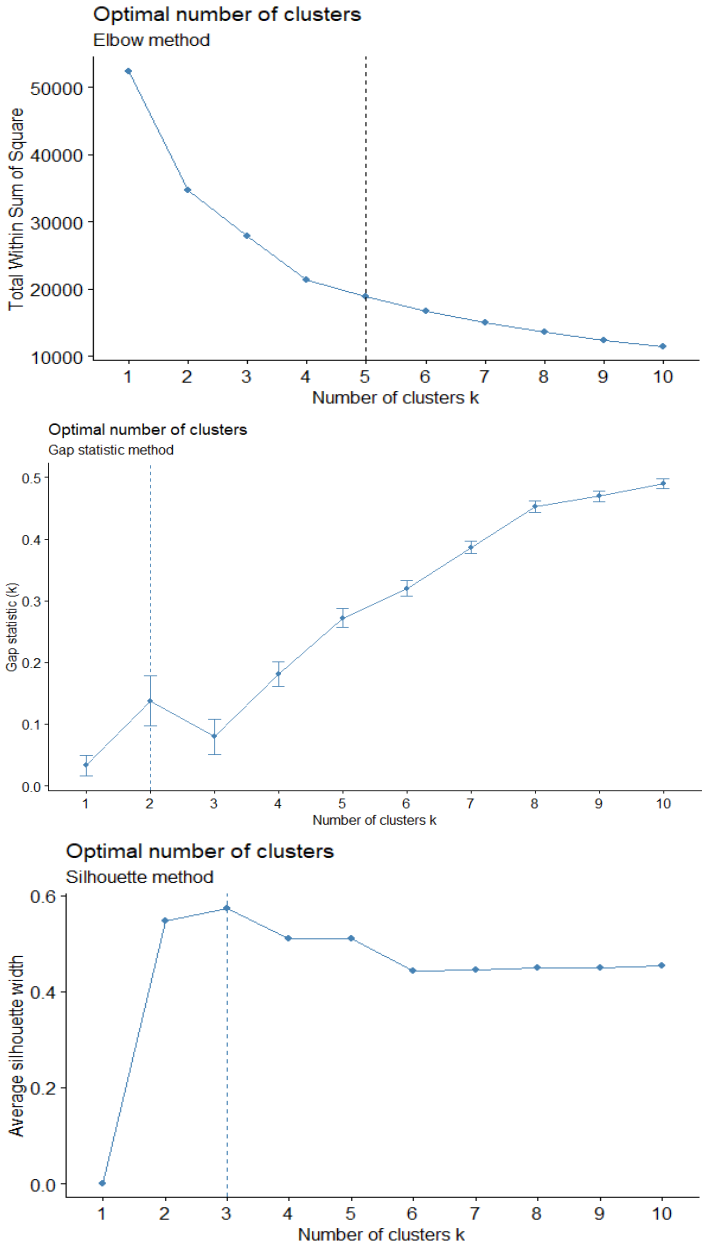


Figure 4.3: Optimal number of clusters (students' profiles) obtained using the elbow, silhouette, and gap statistic methods

datasets.

Figure 4.4: Optimal number of clusters obtained using the majority rules for all the 30 indices in the 'NbClust' package in R

4.3.2 Temporal Profiles Evaluation

Clusters were obtained by calculating the distances between the 23 self-assessment results of each student using DTW. Then, the Hierarchical Clustering algorithm was implemented on the distance matrix of DTW to generate the students' online assessment profiles. It is important to mention that, the cluster number was chosen based on the statistical methods implemented earlier in this section. Figure 4.5 shows the tree-view structure of the profiles obtained from the Hierarchical Clustering algorithm. From these clusters, we can determine which student belongs to which cluster based on the his or her unique identifier. Also, it is interesting to highlight that the clusters C3 and C4 are distinct from the other clusters, with these two containing the least number of students and having very close distance between them. These findings led to conduct further analysis to the students' grades and overall performance to investigate the underlying characteristics of the clusters.

The Hierarchical Clustering of Profiles in Figure 4.5 shows the distribution of student trajectories in the five clusters based on students' performance in 23 online self-assessments and their obtained grades. The grades in the online self-assessments were up to 40 depending on the number of questions in each assessment. By calculating the distance of each pair of students' recorded grades and then clustering these distances (distance matrix) using Hierarchical

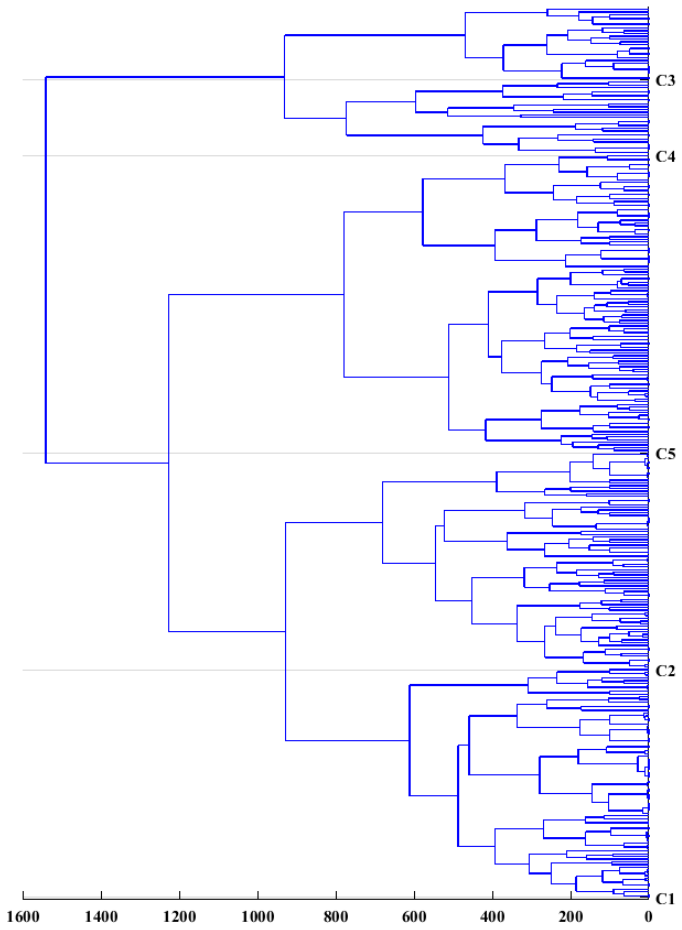


Figure 4.5: Hierarchical Agglomerative Clustering using the DTW distances Matrix for all the temporal trajectories of student's online self-assessments grades. It also shows the margin between the clusters (C1, C2, C3, C4 and C5)

clustering, the clusters (profiles) of students were obtained.

However, the difference between the resulting clusters in Figure 4.6 was not observing the performance of the students, especially between C1 and C2. Because of this, further analysis was conducted to obtain better insight regarding these time-series profiles, and optimally extract useful knowledge on students' profiles differences. I calculated and then plotted the mean values of the 23 self-assessments for all the students in each cluster, as provided in Figure 4.7.

From this plot, therefore, I compared the variation between the obtained clusters to observe the performance differences of the students in each cluster. This reveals that the student profiles

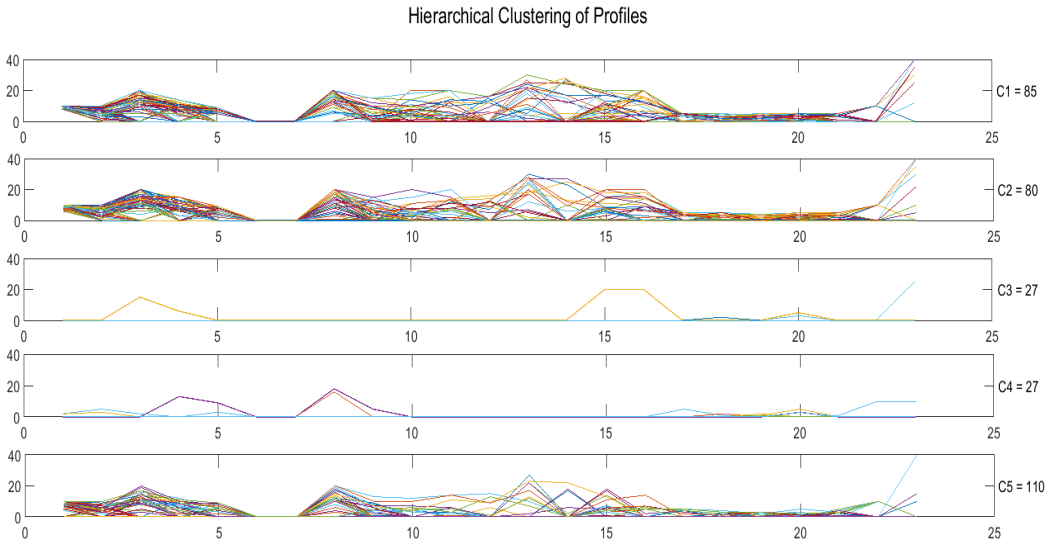


Figure 4.6: The distribution of students' self-assessment trajectories into five clusters. This indicates the hierarchical clustering of profiles and the size of each cluster. This plot was generated using MATLAB

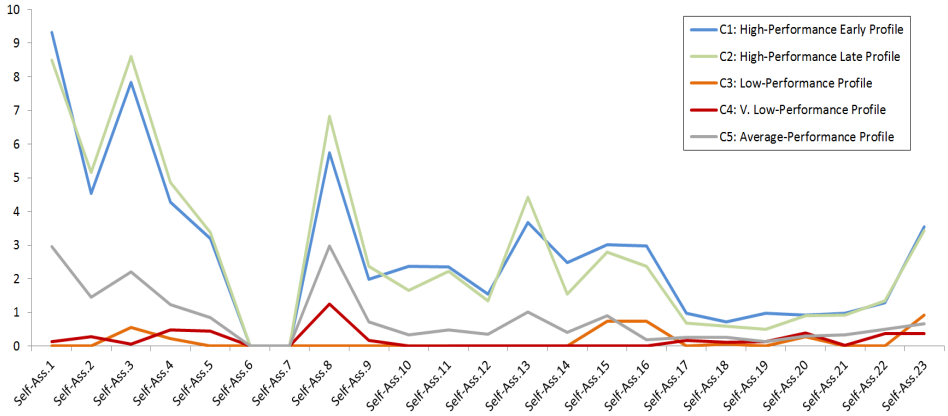


Figure 4.7: Mean of students' grades for the online self-assessments per cluster

in the online self-assessment can be categorized as follows:

- **C1: High-Performance Early Profile**

  C1 includes the first highly achieving group of students in respect of their online self-assessment grades. This profile includes students who were performing well early; obtaining the highest grades among their class in the beginning of the academic term.

- **C2: High-Performance Late Profile**

  C2 includes the second highly achieving group of students in respect of their online self-assessment grades. This profile involves students who were performing well late; obtaining the highest grades among their class from self-assessment 2 to self-assessment 9.

- **C3: Low-Performance Profile**

  C3 includes the low performance students, who did not participate in all the self-assessments and obtained low results.

- **C4: V. Low Performance Profile**

  C4 includes the very low performance students, who did not participate in all the assessments and obtained very low results.

- **C5: Medium-Performance Profile**

  C5 includes the medium performance students, who participated and obtained medium results in most of the online self-assessments.

These profiles have important implications for enhancing students' learning, because by characterizing low performance student's, early intervention can be provided by module tutors. In addition, identifying students' profiles from their online self-assessment trajectories can help instructors determine the needs of each group of students and to decide either to provide extra educational materials or assign other learning activities that may be more suitable for them.

Table 4.4 shows the distribution of students into five clusters and their overall performance in the module. Some interesting findings began to emerge as the follows. The majority of the low risk students were clustered to C1, C2 and C5, which I defined earlier as high performance nd medium performance profiles. The majority of the medium risk students (23 students) were clustered to C5, which I defined earlier as a medium performance profile. Whilst the majority of the high risk students were clustered to C4, which defined above as a very low performance profile.

Table 4.4: Distribution of students in each cluster based on their overall performance for the module

Cluster Label	Cluster Size	Low risk students (Grade A & B)	Medium risk students (Grade c)	High risk students (Grade D, E & F)
C1: High-Performance Early Profile	85 students	*61	16	8
C2: High-Performance Late Profile	80 students	*61	15	4
C3: Low-Performance Profile	27 students	21	4	2
C4: V. Low Performance Profile	27 students	11	7	*9
C5: Medium-Performance Profile	110 students	*79	*23	8

4.3.3 Predictive Model Evaluation

As shown in the proposed approach in section 2, clustering students' self-assessment trajectories was exploited to improve the prediction of their end-of course results. I performed predicting student performance in the Logic and Computation Module using the C4.5 decision tree algorithm by applying two different approaches. The first indicates the prediction of student performance using the CS1005 final grade as a dependent factor on admission attributes only. In the second proposed approach, final student grade was used as a class attribute and the cluster as an independent attribute in the prediction process. The accuracy result is the key indicator for evaluating the approaches. Table 4.5 compares the results obtained from the preliminary predictions of the two approaches and it is apparent from this table that this has been significantly improved in the second approach from 71.31% to 75.52%, when including the DTW cluster as an attribute to predict each student's class or overall performance. It appears that extracting the student performance from their self-assessment profiles in the online module improves the ability to predict how a student will perform. Also, In Table 4.5 the Kappa statistical results of the previous two prediction approaches are provided, which are statistical testing measures (Viera et al., 2005) that consider the agreement between the prediction results and the true classes of students' performance. The Kappa values were improved when students' resulting profiles were added to the dataset from 0.56 to 0.64.

However, accuracy results and weighted Kappa statistics are not the only measures for the efficiency of the predictive model. In Table 4.5, I also include the breakdown of other important measures of the sensitivity and specificity analysis including: True Positive (TP), False Positive

(FP). The TP and FP refer to the correct and incorrect detection of student risk using the C4.5 decision tree algorithm. In particular, the probability of correctly detecting low, medium and high risk students in the Logic and Computation module (CS1005) is clearly improved in terms of 0.665, 0.740 and 0.880 in the three classes, respectively compared to the identified TP Rate of the first prediction approach. Moreover, detailed information on Precision, Recall and the F-Measure are provided (see Table 4.5). Precision refers to the correct fraction of students who were classified positively out of all the students, whereas Recall pertains to the fraction that the predictive model picked up of all the positive cases of the classified students. I used the F-Measure to represent a balanced mean between Precision and Recall for comparing the two approaches. Regarding which, this measure of the high risk student class has been improved in the second approach compared to the first one from 0.83 to 0.86.

Table 4.5: Detailed accuracy of the predictive model by class

Approach	Accuracy	Kappa Value	Class	TP Rate	FP Rate	Precision	Recall	F-Measure
Student Attributes only	71.31	0.56	Low risk	0.56	0.14	0.70	0.56	0.62
			Medium risk	0.76	0.20	0.62	0.76	0.68
			High risk	0.83	0.08	0.82	0.83	*0.83
Student Attributes + Temporal Profiles	75.52	0.64	Low risk	*0.66	0.14	0.73	0.66	0.69
			Medium risk	*0.74	0.14	0.69	0.74	0.71
			High risk	*0.88	0.07	0.84	0.88	*0.86

The obtained C4.5 decision tree after deploying the DTW distance-based clusters is shown in Figure 4.8. Some interesting findings can be extracted from the resultant predictive model. Firstly, the model shows that the prediction of student performance is highly influenced by the clusters. Secondly, the clusters were related to two route codes (G400USCMPSC1 for Computer Science and G500UBUSCOMP for Business Computing) indicating differences in students' ability based on their routes. Further, the cluster is the third most influential factor after the route code and socioeconomic class. It can therefore be assumed that this result was achieved due to the association of student performance with the other student features. Thirdly, it is interesting to note that the prediction result has clearly improved when including the clusters, which means the approach provided in this chapter gives a better prediction result by including the cluster as an independent factor. These findings were very encouraging for

future investigation.

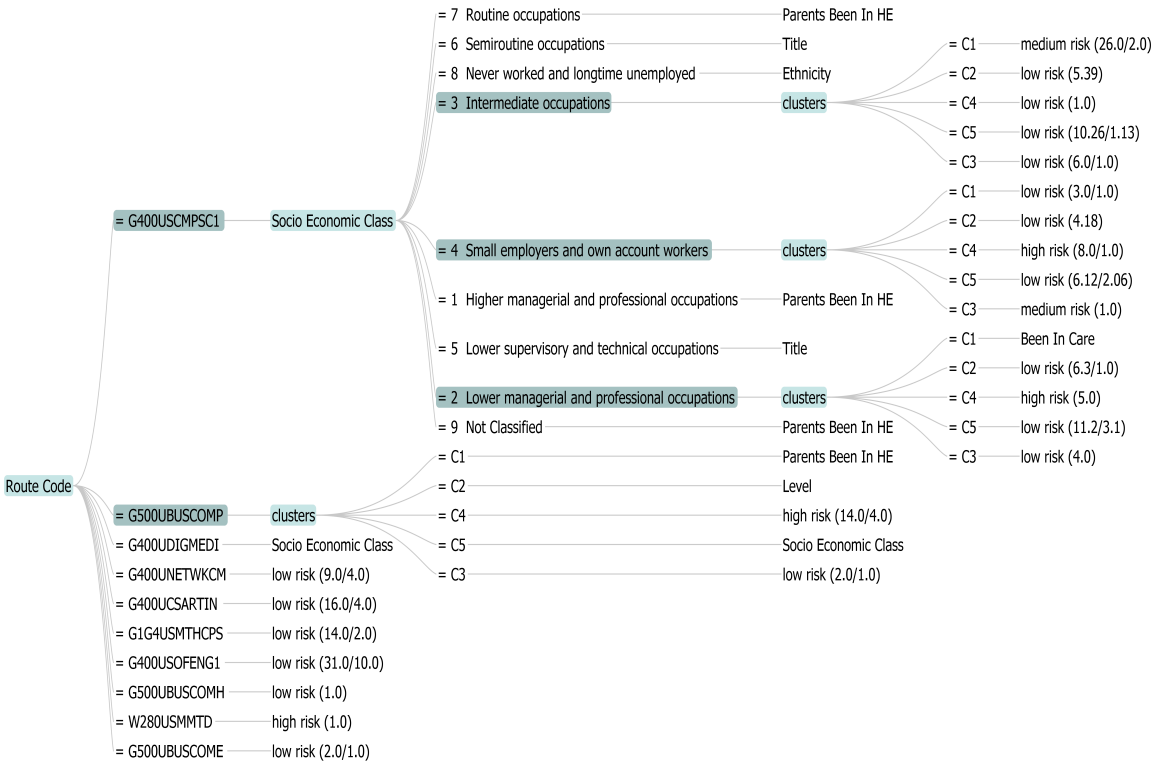


Figure 4.8: C4.5 Decision tree result for predicting student performance on the Logic and Computation Module (CS1005) based on the DTW clusters and admissions datasets. This tree was generated using the Prefuse tree Package in the WEKA Mining Tool. It indicates the most influential factors of the predictive model as being: Route code, Socioeconomic Class and DTW clusters

I then explored whether we could generalize the discovered student types to other module results. For example, if we could establish that a student is highly engaged and motivated in one module, then maybe this may help in the prediction for other module grades. To test and evaluate the proposed framework, the same approaches were applied to all Year 1 Computer Science Modules. I continued to use student online trajectories of the Logic and Computation module (CS1005) for the prediction of student performance in these modules. It is interesting that the consideration of students' temporal profiles enhances the prediction for some modules. However, these identified profiles are not relevant for enhancing the prediction

of student performance in the other modules as each one has different learning activities and objectives. As can be seen from Table 4.6, some improvement in the prediction results were found when adding the obtained DTW clusters in which the performance of students based on their online self-assessment trajectories is characterised. Whilst the prediction results of some modules were not significantly improved compared with the prediction results of CS1005, these results still report some interesting findings. It is recommended that students' online profiles should be considered as an independent factor for the measurable modules, such as CS1005, when the students are required to test their understanding and abilities of some specific elements of the module through online activities and/or assessments. This is actually a very important aspect of the learning process particularly when assessing the students for one course using an online tool. Also, each module has different objectives and learning outcomes. It is not appropriate, therefore, to apply data from one module across multiple modules that might consider complex solutions.

The results of clustering students' temporal assessment trajectories in eLearning systems using DTW therefore are significant in two respects: clustering of students' performance using the online assessments when measuring students' understanding and/or skills online so that early intervention can be provided to these cohorts; and enhancing students learning experience during the academic term to help them then to improve their academic performance, hence too, their learning outcomes.

Table 4.6: Accuracy comparison of all Year 1 predictive models.

Approach	**CS1004**	**CS1005**	**CS1803**	**CS1804**	**CS1805**	**CS1809**	**CS1810**
Student Attributes Only	65.50	71.31	84.83	80.90	71.40	70.15	61.75
Student Attributes + Temporal Profiles	66.05	75.52	85.71	81.51	71.25	74.03	61.53

4.4 Summary

Predicting student performance has become a major concern for many educational data mining researchers. Several aspects could influence the prediction process and provide optimal prediction results in early stages. Accordingly, the present chapter was designed to investigate the effect of clustering student online temporal data on improving performance prediction. I proposed a prediction approach based on temporal profiling of students' online self-assessment trajectories using DTW and Hierarchical Clustering algorithms. The most important finding was that the DTW distance based-clustering approach to these trajectories enhanced the prediction of students' performance in the targeted module. That is, the accuracy of predicting student performance in Logic and Computation Module (CS1005) increased when the DTW clusters (for the same module) was included, from 71.31% overall accuracy to 75.52%. Another important finding is that clustering student online self-assessments using DTW can be useful in identifying students' temporal profiles and for determining whether a student has a low, medium or high level of performance for the online modules. Although the increase in accuracy cannot be seen for all level one modules (e.g CS1805) but the prediction results of some modules have increased when using the online profiles of CS1005 in these modules. The reason behind these limited improvements is because of the lack of identifying online self-assessment profiles for each module.

This chapter confirms previous findings and contributes additional evidence that suggests that tracking student features of the learning management systems (eLearning platforms) for modelling performance prediction can provide useful results. Furthermore, the ML approach used for clustering students' online self-assessment trajectories on eLearning systems has significant practical applications. This method is not only useful for performance prediction, for it can also be developed as a visualisation tool on BlackBoard Learn or any other learning management system to enhance students' experience when being assessed online. The students can track their progresses in the online module throughout the academic term and identify their achievements comparatively among their class. Therefore, a number of possible future studies using the same experimental set up are apparent to generate effective models of students' per-

formance using other learning trajectories - in particular, the time series features that could be extracted from eLearning systems.

In the following chapters, the DTW clusters will be assessed in more detail, in particular, with regard to identifying the relation between the students' identified online profiles and their performance. To this end, a probabilistic graphical modelling approach, namely Dynamic Bayesian Networks (DBNs), will be used for performing predictions of student trajectories.

Chapter 5

Bootstrapping "Ordered Bayesian Networks" for Predicting Minority Class Students-At-Risk

This chapter explores a bootstrapping method to resample temporal educational datasets in order to improve learning the Bayesian Network structures for detecting the minority class of high-risk students. This chapter also investigates the incorporation of students' cognitive styles and online self-assessment profiles results from Chapters 3 and 4 into an ordered Bayesian Network so as to enhance the prediction of students' performance at different stages of their study.

5.1 Introduction

Predicting student academic performance is a major area of interest within the field of EDM in terms of ascertaining accurately what, as yet, unknown knowledge regarding this performance, such as final grades (Romero, Lopez, Luna and Ventura, 2013), will transpire to be. However, predicting students' academic performance is a very difficult task as it is influenced by social, environmental and behavioural factors (Bhardwaj and Pal, 2012); (Araque et al., 2009). Thus,

machine learning algorithms are increasingly being used to discover the relationships and the hidden patterns between these factors and the academic performance of students. There are different educational predictive models, using machine learning, for student assistance aimed at helping them to achieve an improvement in their studies. Students' academic achievement and future performance by can be modelled using Bayesian networks (BNs) to handle uncertainty in the data. For instance, (Seffrin et al., 2014) used dynamic Bayesian networks (DBNs) to interpret and analyze students' cognitive structure over time. Bayesian networks (Pearl, 2014) involve a classification approach based on probability theory (Witten and Frank, 2002) and are considered the best predictors. Such probability predicts the membership of all student-related factors and the class factor by assuming that the independency of the latter is based on the associated values with the other attributes in the prediction model (Kabakchieva, 2013). Thus, the independent effect of the attributes in the Bayesian network model plays a crucial role in classifying students. As a result, the classification model determines the accuracy of predicting classes according to the classified instances.

Previous research has recommended bootstrapping for sample-size related issues. For example, (Adèr, 2008) has recommended bootstrapping when the size of the sample is insufficient for straightforward statistical inference as the case of the number of the high-risk students in the educational datasets. It is not possible to infer an accurate probabilistic graphical model that predicts the high-risk students with very limited data trajectories compared to the medium and low risk students. Hence, a balanced sample is required for optimal statistical inference of the model. This is important for generalising the results of the sample to a larger population. However, (Athreya, 1987) states that "Naive bootstrap could be bad if the underlying population has no forth moment", such as doing some modifications on the sample size or trimming a smaller sample from the original one. This will lead to incorrect variance estimation of the underlying distribution of the sample. Because of this, a decision was made in this chapter to consider this in the resampling process and not to change the size of the sample to either a larger or smaller size to obtain efficient results.

There has not been much related work in the educational system that has exploited the bootstrap approach (Feng et al., 2009); (Beal and Cohen, 2005) for overcoming the issue of the

imbalanced datasets, where the number of instances in each class is different, with there being a much higher instance distribution in one class than the others. Regarding which, McLaren and co-authors (McLaren et al., 2004) utilised and validated their statistical results by using bootstrapping with randomisation to develop a tutoring component from students' log files and interactions in intelligent tutoring systems. The technical objective of their work was to transform the log file data into sequences of action messages. There has been no previous work in educational informatics that involved applying BNs to bootstrapped time-series datasets of students' progression information to address the issue relating to imbalanced classes. The work presented in this chapter is the first usage of them with the aim of achieving an improvement in student performance overall, especially for classifying the minority-class high-risk students. Also, this chapter contributes to the field of EDM by incorporating students' cognitive styles and online self-assessment profiles into an ordered Bayesian Network for predicting student performance at different stages of their study.

However, from a practical point of view, a common issue with classifying students from the educational datasets is that prediction is always dependent on attributes and the prediction can be easily improved or changed when we include more factors over time during the prediction process. This means time is really important for performance prediction modelling as well as including more parameters before predicting the targeted classes. Another issue with classifying students is that the educational datasets usually contain imbalanced data, especially for high risk or failed students compared to the excellent or medium performance ones. Because of this, I proposed in this chapter a Dynamic Bayesian approach for predicting academic performance using a resampling method in order to achieve more accurate prediction results even when we have imbalanced data. That is, a probabilistic graphical model that models the performance of university students is proposed here taking into account the imbalance issue in educational datasets.

In this chapter, a resampling method was exploited on students' obtained grades and other students related attributes, with bootstrapping, to compare the accuracy of the Bayesian models to ensure that more states of student overall performance are obtained than using the original time series datasets. The most salient finding is that, the accuracy of the results is

improved when determining the bias issues using the bootstrapped data from the time series educational data.

This chapter provides, first, a novel Bayesian approach for predicting university students' academic performance from time series educational data. Here, a probabilistic modelling approach was exploited to identify the relationships between student's time series attributes and their overall performance. Secondly, I explored the use of a bootstrapping method to resample the educational datasets in order to improve learning of the Bayesian structure, whilst also enhancing the detection of the students of the minority class, who are those at high risk, as early as possible.

5.2 Bayesian Networks, Dynamic Bayesian Networks and Ordered Bayesian Networks

Bayesian Networks (BNs) (Nielsen and Jensen, 2009) (Friedman et al., 2000) is a machine learning method that is implemented using a graph theory in conjunction with statistical methods to characterise the correlations and dependencies between the features or attributes. The representation of such dependencies is generated through a graph-based structure showing a graphical network based on a conditional probability distribution among the features. The graph-based structure is convenient for explaining the dependencies of the relations among the discrete or continuous features.

However, when modelling different a data type (i.e. time series data) the BN is achieved using another learning approach to represent the features at a particular time slot. This structure based learning approach is called Dynamic Bayesian Networks (DBN), which apply complex processes to capture the observed as well as the unobserved states of the features (in terms of having temporal aspects, such as time-series dimensions) (Murphy and Russell, 2002). Thus, these learning processes allow for observing and updating the network when time progresses to provide robust predictive models with consideration of the behaviour of the BN (Mihajlovic and Petkovic, 2001). This assumption will learn a dynamic model which is not determining any changes in the structure of the BN over time.

In contrast, with the discrete time BNs, three-time steps were observed for learning the Ordered Bayesian Network in this chapter, as shown in Figure 5.1. The Ordered Bayesian Network is a machine learning method that is implemented using a graph theory to learn the joint probability distribution between the features over time, with consideration of the time-dependent order between the different features. For example, the structure learning approach in Figure 5.1 was utilised to identify the joint probability distribution between students' overall performance and other related attributes over time, including module grades in Years 1 and 2, online temporal assessment profiles and students' cognitive styles, as identified from the experimental works of clustering students' temporal engagement datasets in Chapter 3. Thus, with this structure learning approach, the features will be assigned to different temporal slots (t, t+1, t+2) to indicate the time-dependant order between the features. The Ordered BN learning approach, therefore, is not exploiting a dynamic Bayesian (DBN) approach (or temporal) as the nodes at each time slot changes and are not repeated. For example, in time (t) the structure learning approach of the BN is encoded to include the background knowledge of the admissions features as an initial step, whereas in time (t+1), different features are included, such as year 1 end-of course grades. Thus, this type of the BN structure construction can be determined as ordered Bayesian Network, where each node is placed in a time slot, which indicates that no backward links (arcs) from the students' features in Year 2 to the features in the admission level or Year 1. This constrain is really important especially when considering the causality of the earlier students' performance (in Year1) to the later temporal modules in Year 2. This has the assumption that the prior probability of the BN structure is related to data given the probability of such structure (Heckerman et al., 1995).

5.3 Experimental Settings

5.3.1 Dataset pre-processing

The datasets used in this investigation were collected from Brunel University admissions and computer science databases. This consists of 14,160 records for computer science students, but I only considered 377 records as I were targeting to track the performance of the same

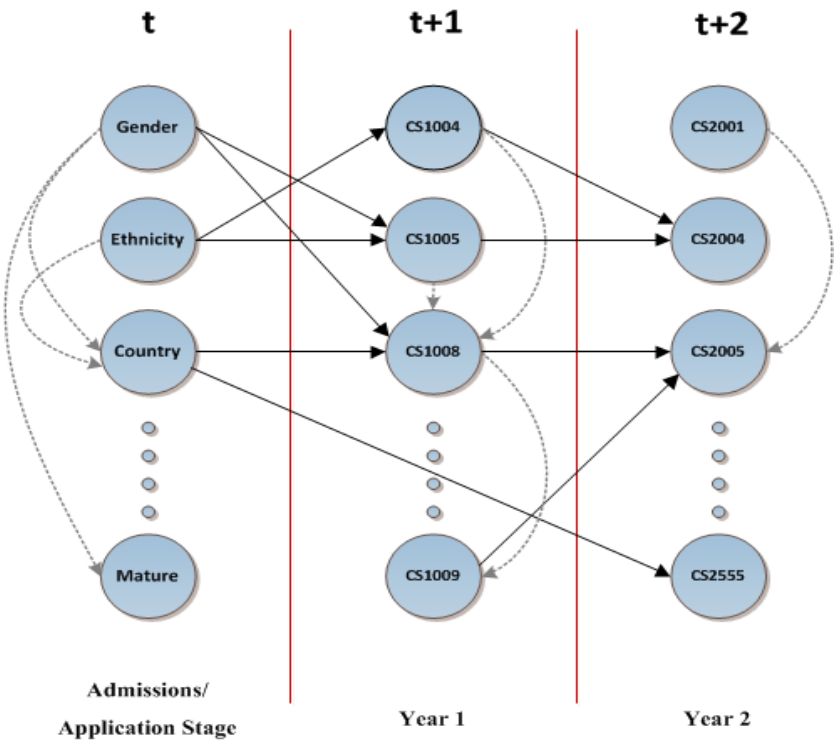


Figure 5.1: Proposed ordered Bayesian network approach for three time slots (t, t+1 and t+2). It illustrates the correlations between the admissions/applications data of students in time slot t, students' grades and other related attributes in Year 1 and Year 2 for time slots t+1 and t+2, respectively.

group of students in three different time slots: t at admission level, t+1 at Year 1 and t+2 at Year 2 (see Table 5.1), for the academic years 2014, 2015 and 2016, respectively. The datasets contained the following data categories:

- **Admissions Dataset:** includes students' application data when entering the university this was considered in this study as time (t) data, such as: nationality, ethnicity, country of birth, disability, been in care, socioeconomic class . . . etc.;
- **Progression Dataset:** this includes student coursework, online assessments and final grades. In this study I only considered student final grades for all Year 1 (t+1) and Year 2 (t+2) modules at the university for measuring students' overall academic performance;
- **Cognitive Style:** this includes students' attitude towards turning up to classes and labs in Year 1 and Year 2 at the university (obtained from Chapter 3);

- **Online Temporal Assessment Profiles:** this includes a student online self-assessment profile based on their online time-series assessment trajectories. These profiles were obtained (in Chapter 4) using Dynamic Time Warping (DTW) and Hierarchical Clustering Algorithms.

The data mentioned in Table 5.1 has been then discretized to different data categories based on the domain values of each feature. This is a fundamental pre-processing mechanism to transform categorical and nominal data to numerical values to feed the Bayesian Network and calculate the prior and posterior probabilities for all features. The Feature Engineering and the discrtization mechanism for the educational features is available in Appendix A.1 with details of the new domain values.

As with many prediction issues, educational datasets usually include imbalanced data, in this case, because not all students are performing similarly. Thus, the number of students in Low, Medium and High risk classes is not equally balanced (see Figure 5.5 (A)). Figure 5.2 indicates the methods applied for learning ordered Bayesian Network from imbalanced educational classes. Two different approaches were determined to learn such models. I focus on first predicting students overall future performance at university based on the original educational time series records. Then, the resampling approach was investigated with replacement in order to have some insights into the current problems with the imbalanced educational time series records. Hence, the issue of predicting students' performance based on imbalanced time series data can be determined using the resampling bootstrapping approach.

Whilst the main objective of this chapter is to determine the imbalance issue with time series educational data using a resampling strategy, another is to model university student academic performance using probabilistic modelling to identify the relationships between students' admission attributes, final grades for modules and their final progress at the end of Year 3. For this purpose, I designed an ordered Bayesian model on student's temporal data, taking into consideration the imbalance issue of the predictive classes. I examined the use of the bootstrapping method (see Figure 5.2) for better and early detection of students at risk by deploying temporal educational and progress data.

Table 5.1: The educational time-series datasets. It includes the admission (entry) datasets and the progression datasets for all the Year1 and Year2 modules.

Time of Data Collection	**Feature**	**Description**	**Domain Values**
Admission Level (t)	Gender	Student gender	{M, F}
	Disability	Whether the student has any disability issue	{A social/communication impairment , A specific learning difficulty, No known disability}
	Nationality	Student nationality	{British national, Belgian, Bulgarian, .., etc}
	Ethnicity	Student ethnicity	{Arab, Asian Other, White, White-British,., etc} based on Students ethnicity
	Country of Domicile	Student country of domicile	{England, Greece, Spain, . , etc} based on students country of domicile
	Country of Birth	Student country of birth	{England, Greece, Spain, . , etc} based on students country of birth
	Entry Year	The entry year of the student at the university	{2012/13, 2013/14, 2014/15, 2015/16}
	Fee Status	Tuition fee status	{Home, European, International}
	Socio Economic Class	Student socio economic class	{1 - Higher managerial and professional occupations, 2 - Lower managerial and professional occupations, - Not Classified, ... , etc}
	Previous Ed Estab LEA	The student previous education establishment	{Bexley, Hillingdon, Leicester, Harrow, City of London, Ealing, . . . , Unanswered} based on student's previous education establishment
	Been In Care	Whether the student has spent any time in care	{DATA UNAVAILABLE, No, Unanswered}
	Mature	Whether the student is fully developed physically	{Yes, No}
	Age on Entry	Student age when entry	{17, 18 . . . ..} based on students age on entry
	Parents Been In HE	Whether the students' parents have been to any higher education institution	{Yes, No, Do not Know, Prefer Not To Say}
	Route Code	The code of student chosen route at the university	{G400USCMPSC1 for Computer Science, G400USOFENG1 for Computer Science (Software Engineering), .., etc}
	Current Student?	Is the student crrently at the univerity?	{Yes, No}
	Initial cog. Style	The resulting Year 0 clusters from the cognitive style detection (Chapter 3)	{C1, C2, C3, C4, C5}
Year 1 (t+1)	CS1004	Information Systems and Organisations Module Final Grade	{A - Excellent, B - very Good, C - Good, D - Acceptable, E- PASS, F-Unacceptable}
	CS1005	Logic and Computation Module Final Grade	
	CS1803	Level 1 Group Project Reflection Module Final Grade	
	CS1805	Data and Information Assessment Module Final Grade	
	CS1809	Software Design Module Final Grade	
	CS1810	Software Implementation Event Module Final Grade	
	CS1811	Fundamental Programming Assessment Module Final Grade	
	Temporal Profiles	The resulting clusters from the DTW and hierarchical clustering process (Chapter 4)	{Very Low Performance Profile, Low Performance Profile, Average performance Profile, High Performance Late Profile, High Performance Early Profile}
	Year1 cog. Style	The resulting Year 1 clusters from the cognitive style detection (from Chapter 3)	{C1, C2, C3, C4, C5}
Year 2 (t+2)	CS2001	Year 2 Group Project Module Final Grade	{A - Excellent, B - very Good, C - Good, D - Acceptable, E- PASS, F-Unacceptable}
	CS2002	Software Development and Management Module Final Grade	
	CS2003	Usability Engineering Module Final Grade	
	CS2004	Algorithms and their Applications Module Final Grade	
	CS2005	Networks and Operating Systems Module Final Grade	
	CS2555	Work Placement Module Final Grade	
	Year2 cog. Style	The resulting Year 2 clusters from the cognitive style detection (Chapter 3)	{C1, C2, C3, C4, C5}
	CLASS	Student overall performance	{Low Risk, Medium Risk, High Risk}

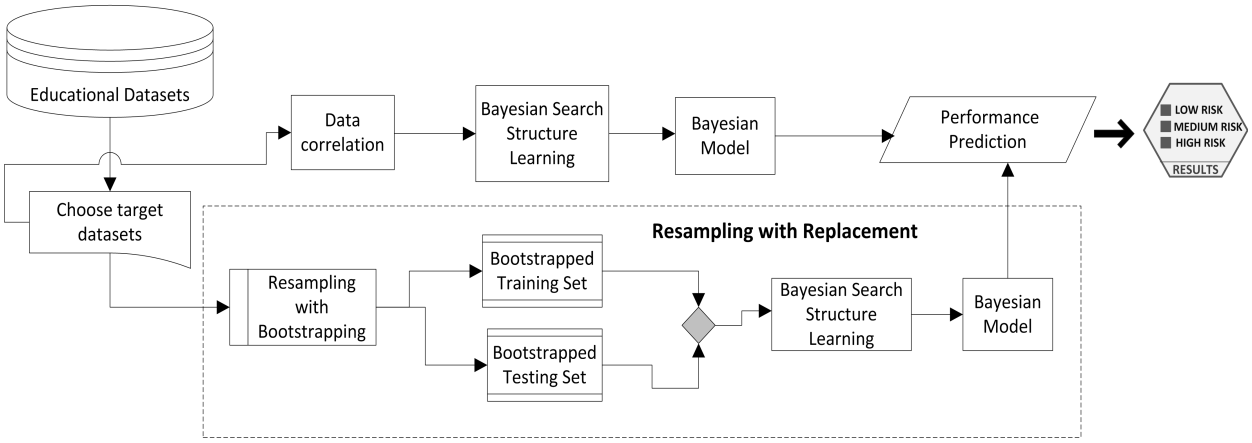


Figure 5.2: This Figure presents Bayesian structure learning and the resampling strategy used for learning the Ordered Bayesian model. Two different approaches were applied to learn the structure; use of the original and bootstrapped time series educational datasets.

5.3.2 Bayesian Structure Learning

As mentioned earlier, this chapter provides also a model for predicting students' future performance using a graphical probabilistic method with Bayesian Networks (BNs). The BN algorithm was exploited here to encode the probability distribution over the educational features as well as identifying possible correlations between the educational features and the future performance of students. To obtain that, the BN algorithm represents first a directed acyclic graph (DAG) by performing conditional independence relationships between the features (Friedman et al., 2013). The nodes in the DAG are the educational features and the links consider the conditional dependencies between the features in the graph. Thus, the links are directed from the parent nodes to the child nodes of the educational features and the lack of identifying links computes conditional independences. The conditional independence between two educational features x_1 and x_2 given x_3 represents that x_1 and x_2 are independent once x_3 is known which is encoded as:

$$p(x_1|x_2, x_3) = p(x_1|x_3) \tag{5.1}$$

Therefore, a particular feature (node) in the BN is conditionally independent of all the

other educational features, in which it gives the conditional probability distribution (CPD) for this particular feature. Computing the CPD of each feature will obtain the conditional probability table (CPT) of the BN that encodes the joint probability as:

$$p(\mathrm{x}) = \prod_{i=1}^{n} p\left(x_i|\mathrm{pa_i}\right) \tag{5.2}$$

Where x_i is a node in the BN and $\mathbf{pa_i}$ are the parents nodes of this particular. This will denote features and nodes.

The learning of the Bayesian network structure was performed using GeNIe (Druzdzel, 1999), software implemented in Java for learning and modelling Bayesian networks (BNs), dynamic Bayesian networks (DBNs), and influence diagrams (IDs). For learning the BN structure, a Bayesian search method was exploited on the background knowledge of two educational datasets: the original and bootstrapped datasets. A comparison between the trained BNs based on these two datasets was performed to obtain a very accurate and reliable predictive model. The BNs with temporal links inferred from the admissions and students historical grades, are represented in ordered BN of the featuers in time slots (t, t+1 and t+2) as shown in Figure 5.1.

To learn the ordered BN structure, a Bayesian Search (BS) structure learning algorithm was exploited using the temporal dataset to learns the highest posterior probability of the class feature. The BS algorithm essentially is a hill climbing based procedure (controlled by a scoring function) that generates the DAG to determines a maximum score/probability that give the BN structure. This imposes with an assumption that the prior probability of the BN structure is related to data given the probability of such structure (Heckerman et al., 1995).

5.3.3 Bayesian Inference and Parameter Learning

Once the DAG is generated, an important aspect to examine the BN structure is to perform the Bayesian inference for the probabilistic reasoning of the time-dependent educational features. Identifying Bayesian inference therefore enables detecting the state of the educational features as evidence based on the temporal educational data. Each feature/node in the BN is considered

by a state that is dependant the state of the other features. Inferring the BN with such approach allows for updating the evidence with new datasets to examine the posterior probabilities of the features (Koller and Friedman, 2009).

To perform the Bayesian inference, the Bayes rule (Murphy et al., 2001) was encoded to identify the posterior probability of the educational features based on the prior probability. For example, to calculate posterior probability for the class feature 'Student Performance' $P(StudentPerf|CS2005)$ given the prior probability for one of year 2 courses 'CS2005', the Bayes rule is calculated as:

$$P(StudentPerf|CS2005) = \frac{P(CS2005|StudentPerf)P(StudentPerf)}{P(CS2005)} \tag{5.3}$$

Where $P(StudentPerf)$ is the prior probability (values of student performance feature) and $P(CS2005|StudentPerf)$ is the likelyhood given the observed data for CS2005 and Student Performance.

Identifying the joint distribution of the BN based on the calculated Bayes rule among the educational features allow to infer the probability distribution of a targeted feature, given the educational data for the other features in the dataset. Exact inference identification was applied to all of the educational features to obtain the posterior probability in the BN for the class feature 'Student Performance' as well as the other educational features.

Thus, the principle goal of learning the ordered Bayesian network is to find the posterior distribution that is adapted to students' progression data in Year 1 and Year 2, which allows for identifying the states of all students' attributes as well as overall performance. The parameters of the ordered Bayesian model were learned using the expectation maximization (EM) algorithm (Moon, 1996) with the bootstrapped data. This algorithm was implemented to estimate the posterior distribution of students' attributes in time slots t, t+1 and t+2. The EM algorithm was used for learning the parameters of the BN to estimate the maximum likelihood for data (Moon, 1996), which supports learning from time series data. This iterates over a two-step approach to fully learn the BN parameters as follows:

- **Expectation (E) Step:** this step considers learning an expectation function to eval-

uate the log-likelihood for the current parameters of the resampled temporal dataset of students DS_i. Suppose that the temporal student features are $\{x_1, x_2, \ldots, x_n\}$ then the expected values of x_1, given a measurement Y_1 of the current parameter estimation is computed as:

$$x_1^{[k+1]} = E\left[x_1 | Y_1, p^{|k|}\right] \tag{5.4}$$

- **Maximization (M) Step:** this step maximizes the parameters of the log-likelihood that were expected in the previous step. In the context of student data, the expected values of $x_1^{[k+1]}$ and $x_2^{[k+1]}$ are imputed with x_3 to estimate the parameters of the log-likelihood, considering the result of log function equals to 0, this log function is provided as follows:

$$\begin{aligned} 0 &= \frac{d}{dp} \log f\left(x_1^{[k+1]}, x_2^{[k+1]}, x_3 | p\right) \\ \Rightarrow p^{[k+1]} &= \frac{2x_2^{[k+1]} - x_3}{x_2^{[k+1]} + x_3} \end{aligned} \tag{5.5}$$

The BN parameters are then learned by iterating the equations (5.4) and (5.5) until converging all the educational features in the dataset.

5.3.4 Resampling with Bootstrapping

Resampling strategies are fundamental approaches in the pre-processing phase, which are used to change the distribution of data in a dataset (Chawla et al., 2002). After the students' overall grade bands from (A, B, C , D, E and F) have been discretized to qualitative states of low, medium and high risk students, I still encountered an imbalance issue especially for the high risk students (see confusion matrix in Figure 5.5). Though, imbalance data is a very common issue in educational datasets, which affects learning the predictive models as well as making difficulties in identifying the cases of the minority classes. The minority class in this work is the high risk class, which is the class assigned for those students who obtained low grades (D, E and F) in all or most of the modules. Here a resampling approach was exploited on the datasets to obtain reliable accuracy of the prediction results using bootstrapping.

The resampling approach was exploited using bootstrap aggregation (bagging) (Moniz et al., 2016) to ensample the original students' data in Year 1 and Year 2, with replacement. This technique is useful to learn robust classifiers from the small size datasets. Therefore, it has been applied in this experiment to estimate the accuracy of the BN in predicting more student records of all classes and to avoid overfitting. To implement the bootstrap approach, the REPTree algorithm with the classification and regression tree algorithm (CART) were learned using the educational time-series data. The data was resampled using the decision tree algorithms as they are widely used for the low bias and high variance models.

Suppose we have a student dataset DS that consists of the following features $\{x_1, x_2, \ldots, x_n\}$, where each student is labeled to a particular class $C = \{C_l, C_2, \ldots, C_m\}$, in this study as $C = \{C_l : LowRisk, C_2 : MediumRisk, C_3 : HighRisk\}$. The bagging method first learns a subset DS_i from the student dataset DS with a replacement approach. Some of students trajectories in DS therefore will be repeated more than once to achieve more balanced classes. The classifier is learned then with the REPTree and the CART algorithms from the subset DS_i to initialize the weight based on the correctly classified students in the subset. The aim behind learning the classifier with the decision tree algorithms is to determine splitting students' features with the highest gain ratio, therefore the model can obtain high accuracy result on the testing data. It is important to mention that, I could have resampled any size from the educational dataset, but the same number of students' records as in the original (377 students) was decided here for better comparisons to the imbalanced data and decision making when learning the BNs. To this end, the proposed approach for resampling the student dataset is provided as follows:

The algorithm: a resampling classifier with the REPTree and the CART algorithms, as follows:

Input:

- *DS*: a dataset of student performance trajectories.
- *k=2*: the number of predictive models use for resampling the dataset.
- *Learning Methods*: REPTree and the CART algorithms

Output: a composite resampling classifier.

The resampling/bagging approach:

1. Learn a data subset DS_i from student dataset DS with a replacment strtegey.
2. **for** $i = 1$ to k **do**
3. Learn a classifier based on the student subset DS_i.
4. Predict each student class in DS_i and calculate the accuracy based on the correctly classified students in DS_i.
5. **end for**

To validate the bootstrapped data, the mean µ of the distribution to give an 95% bootstrap condence interval was then computed. The mean was $x = 0.67$ for students overall performance, which I used as an estimated value of the mean for the underlying distribution. To calculate the confidence interval I needed to measure the difference between the distribution of x around the mean μ, as follows:

$$\delta = \overline{x} - \mu \tag{5.6}$$

To get this distribution, the standard deviation was computed for the entire student records $\delta.1$ and $\delta.9$, the 0.1 and 0.9, which are critical values of δ to achieve a 95% condence interval of [$\overline{x}-\delta.1$, $\overline{x}-\delta.9$]. The StdDev for the full data was obtained from the following equation:

$$P(\delta.9 \leq \overline{x} - \mu \leq \delta.1|\mu) = 0.95 \quad \Longleftrightarrow \quad P(\overline{x} - \delta.9 \geq \mu \geq \overline{x} - \delta.1|\mu) = 0.95 \tag{5.7}$$

However, the bootstrap offers a direct approach to obtain the distribution of δ, which can be measured by the distribution of:

$$\delta^* = \overline{x}^* - \overline{x} \tag{5.8}$$

Where, $\overline{x}^*$ indicates the mean of the bootstrap data. One bootstrapped sample was generated of size of 377, which was the size of the original data.

5.4 Experimental Results

The results presented in this section provide an evaluation of the proposed Bayesian approach in predicting third year university students' academic performance at the admission stage, followed by the Year 1 and Year 2 stages. Two key experiments were undertaken in this investigation: learning from the original data and learning from the bootstrapped data. I set up these two experiments to show improvement in predicting the academic performance of a student, especially for high risk students, who belonged to the minority class in these experiments.

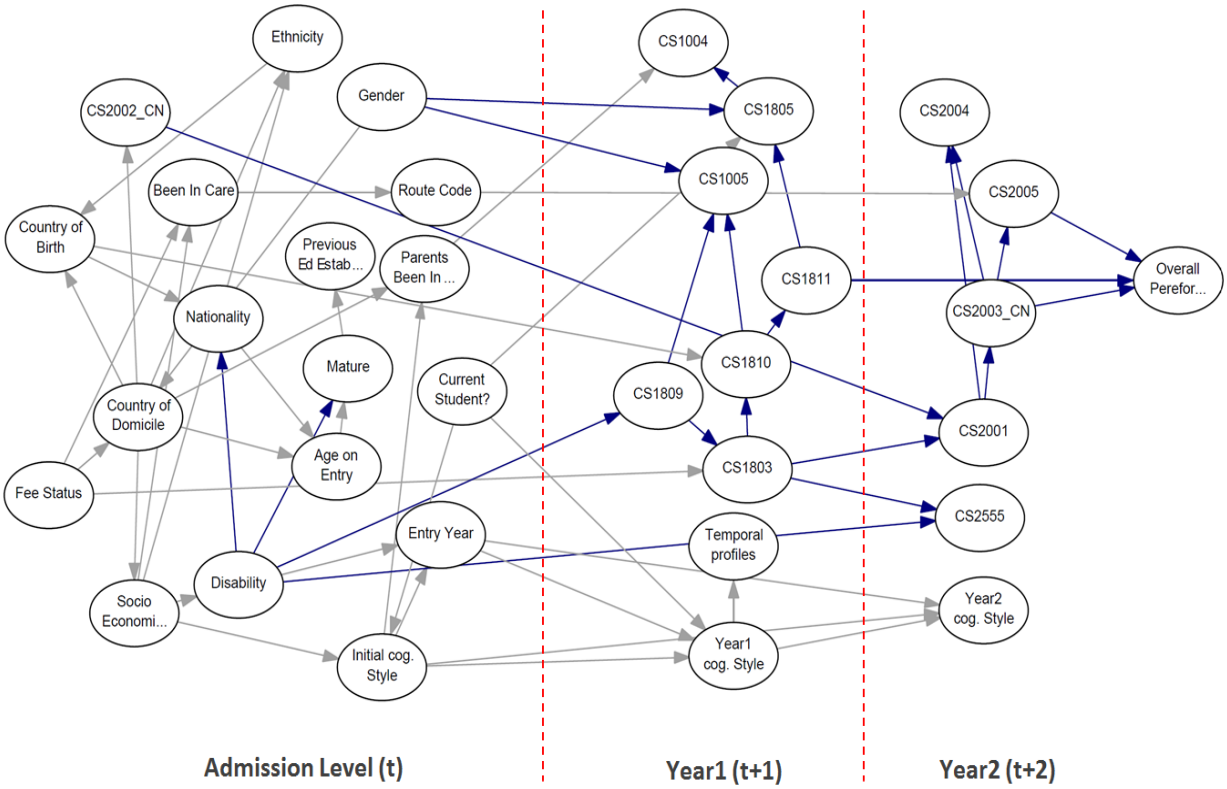


Figure 5.3: Ordered Bayesian structure learned from the bootstrapped temporal educational data. This represents the correlations between students' admissions attributes, Year 1 and Year 2 grades, other attributes and overall performance. The strong relationships between students' attributes and overall academic performance were coloured in blue.

In the learning process, the ordered Bayesian structure was learned from students' data

using the attributes from three different time slots. These were students' admissions attributes (time slot t) when they entered the university, their obtained grades at Year 1 (time slot t+1) and their achieved grades for Year 2 (time slot t+2). In Figure 5.3, I present the discovered correlations between students' admission data and progression data (grades) for the three steps of the ordered Bayesian approach. It is interesting to note that some of the admission nodes in time slot t influence students' achievement in some of the Year 1 and Year 2 modules. In addition, students' overall performance at the end of Year 2 was mainly influenced by their grades in CS1811 (Fundamental Programming Assessment), CS2003 (Usability Engineering) and CS2005 (Networks and Operating Systems), which are compulsory modules for computer science students.

5.4.1 Bootstrapping Validation Results

To examine the effectiveness of bootstrapping on improving prediction of the high risk students and the other classes, a comparison between the two approaches was provided in Figure 5.4. The accuracy results were obtained using the 10 –fold cross validation for predicting the academic performance in time slot t+2, based on all student data, as mentioned earlier. Figure.5.4 shows a significant improvement in identifying the low, medium and high risk students for the bootstrapped data. In other words, the most accurate result for predicting academic performance was obtained using these data. For example, the accuracy obtained for the high risk class using the bootstrapped data was 0.94, whilst when using the original data it was only 0.63.

The confusion matrices in Figure 5.5 indicate the number of predicted low, medium and high risk students for the class attribute 'overall performance' using the original and the proposed bootstrapped data approach. It also reveals the percentage of classification accuracy for each predicted class using the original dataset (A) and bootstrapped dataset (B).

For evaluating the performance of the temporal predictive model, I performed sensitivity and specificity analysis on the cohort of students who were predicted to be at low risk, medium risk and high risk for the best performed Bayesian model, which was when applied using the bootstrapped data. To this end, the Receiver Operating Characteristics curve (ROC) and the

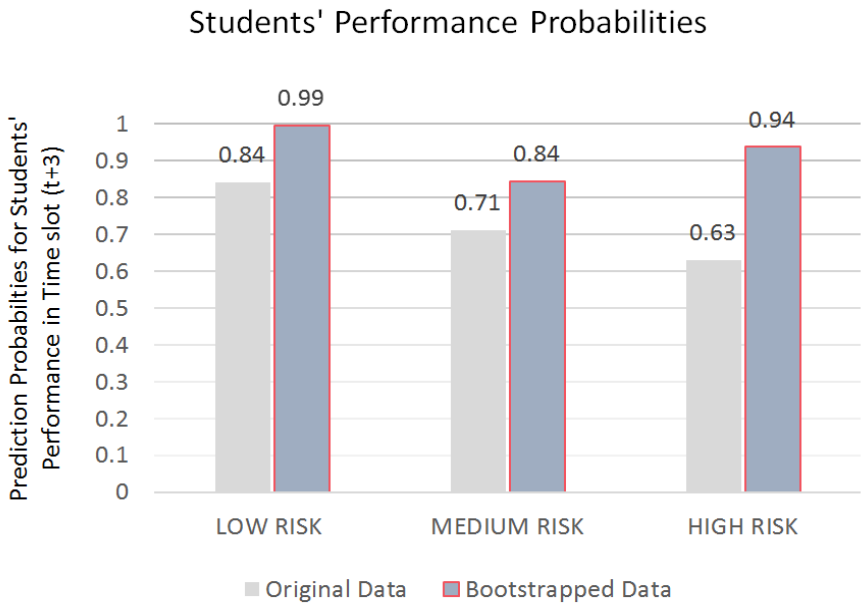


Figure 5.4: Prediction probabilities for students' overall performance using two approaches (original and bootstrapped data). It represents the accuracy results for the three classes.

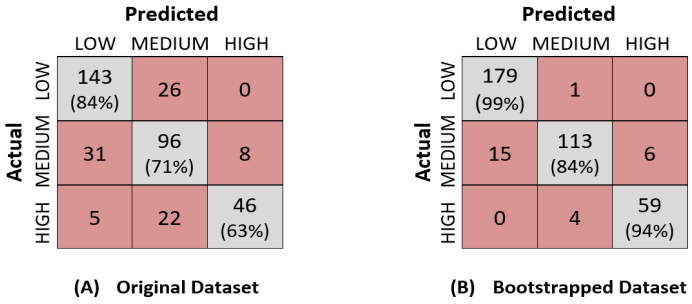
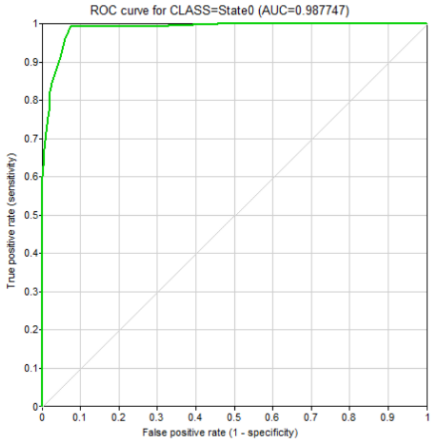


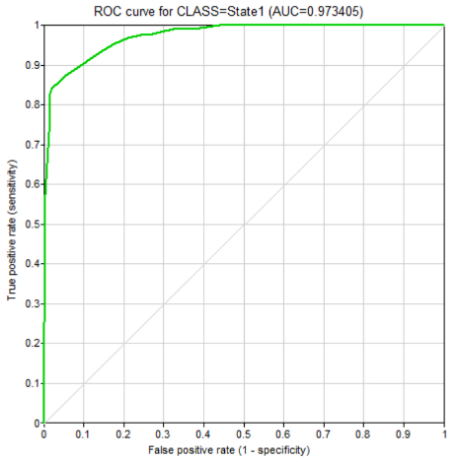
Figure 5.5: Academic performance confusion matrices comparing prediction results for the class attribute (student academic performance) using the original dataset (A) and bootstrapped dataset (B).

Area under the Curve (AUS) were visualized, as shown in Figure 5.6. These two performance measurements were used as I had a multi class predictive model. It can be seen from the ROC curves in Figure 5.6 that for the low risk prediction (a) and high risk prediction (c) are very close to 100% sensitivity and 100% specificity, which means a perfect discrimination of the overall prediction accuracy based on the bootstrapped educational data.

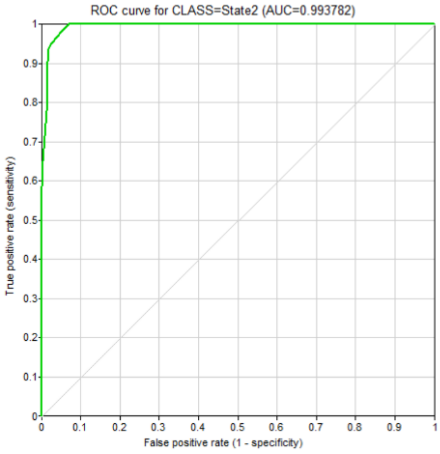
The validation of the ordered Bayesian approach in predicting the academic performance in Year 3 was then examined using supplied test sets, as shown in Figure 5.7. This is a key result



(a) Low Risk ROC curve



(b) Medium Risk ROC curve



(c) High Risk ROC curve

Figure 5.6: ROC curves of students' overall performance using the bootstrapped data. It represents the ROC curves for the three states, these being: (a) state 0 for the low risk, (b) state 1 for the medium risk and (c) state 2 for the high risk students.

that showing how the "Ordered BNs" allow to make predictions that are increasingly accurate as more evidence is made available when bootstrapping is used. Firstly, students' performance was predicted based on admissions data only at time slot t for the two datasets explained in the methodology section, which were the original data and the bootstrapped data. Secondly, I added more data, which were students' progressions and final grades at Year 1, to observe how better I can predict using the temporal approach. After that, the performance was predicated using all students' attributes. It is apparent from Figure 5.7 that, the prediction was improved in time slot t+1 when Year 1 grades were added to the admissions data. This improvement was due to the direct relation between students' achieved grades in Year 1 with their overall performance.

The most important limitation lies in the reasonableness of the model assumptions of drawing inferences when using the bootstrapped educational sample. This is because with the Bayesian bootstrapping the predictive model/classifier is inferred based on the posterior distribution of the parameters, rather than the sampling distributions, as in inferred in the normal Bayesian network, which will affect the resulting distributions as well as the interpretation of the graphical model. However, the resulting inference of the Bayesian bootstrap can be considered as an advantage, because it produces probability statements about parameters rather than frequency statements about statistics. Moreover, the generalisability of the prediction results using the bootstrapped educational sample is subject to certain limitations. The educational data includes biased features, such as students' gender, nationality, ethnicity and so on, which exist in the datasets due to the data collection or sampling approaches. Thus, the Bayesian bootstrapping will also carry bias into its implementation process as the Bayesian model is inferred using biased features, which should be not determined in the classification process. This issue is addressed in the next chapter to perform transparent classifiers for ethical decision making with predicting student performance using the advances of the BNs and Deep learning methods. However, if the biased features are explored and handled the generalisation of the learning approaches will be satisfactory for many practical applications.

It is possible, therefore, that the Bayesian bootstrapping of the ordered Bayesian Networks could be applied to many other domains, when the analysts aim to demonstrate how good

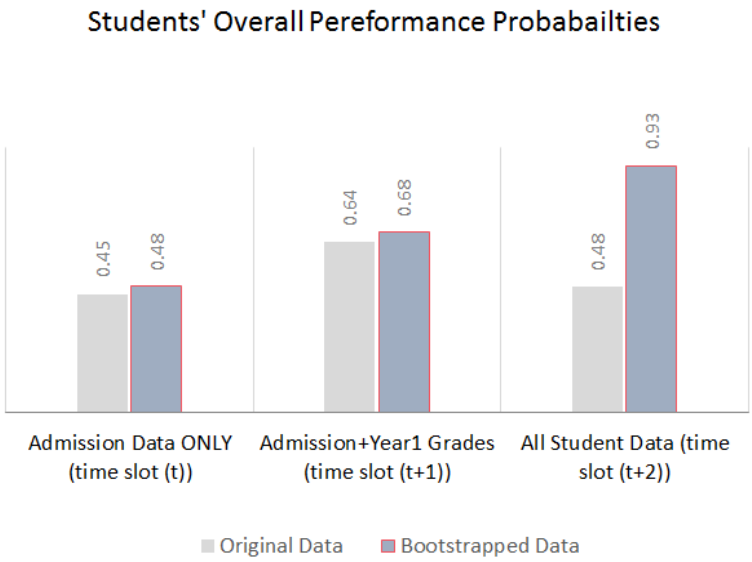


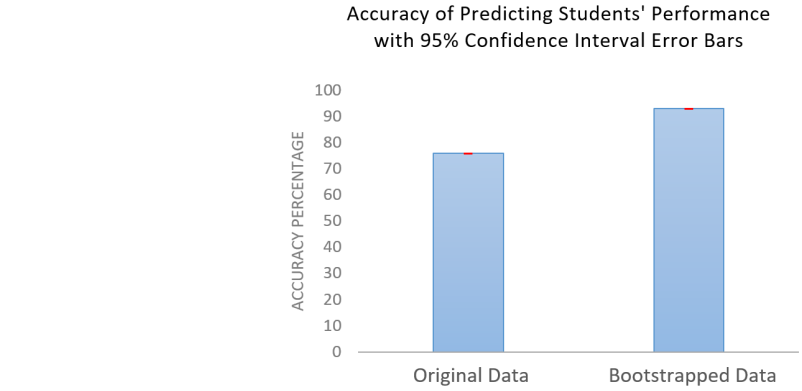
Figure 5.7: Validation probabilities for students' overall performance based on the original and bootstrapped data. It represents the accuracy results for the two different attempts. These are when student admissions data were used in time slot (t), then admissions data plus Year 1 student grades in time slot (t+1) and finally, all students' data, these being admissions data,Year 1 and Year 2 students grades in time slot (t+2).

the time-series data trajectories are in supporting the findings of the model-based analysis. Furthermore, Bayesian bootstrapping is another alternative approach to the non-parametric Bayesian models, which is less computationally demanding. This method could see greater use as computational environments change.

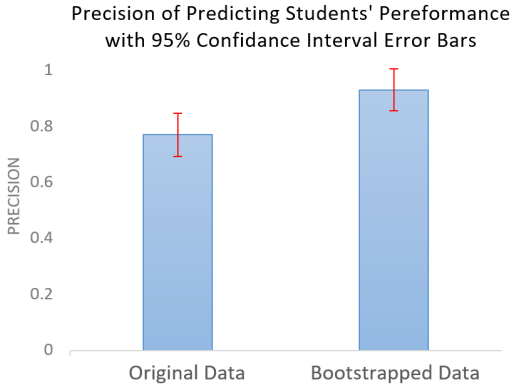
5.4.2 Confidence Interval Results

This section presents the influence of using bootstrapping to improve learning a Bayesian network model for the purpose of predicting the academic performance of students. The plotted chart in Figure 5.8 (A, B and C) compares the accuracy, the precision and the sensitivity results for predicting the academic performance of students, among 377 students' time series records for two different datasets (original data, bootstrapped data). I also show the error bars with a 95% confidence interval, which helps in observing the difference between error bars where they overlap or not.

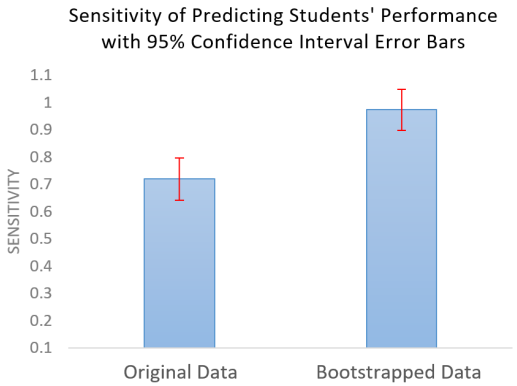
It is apparent from Figure 5.8 (a) the error bars are quite small due to the corresponding



(a)



(b)



(c)

Figure 5.8: Confidence interval (CI) error bar charts for the accuracy (a), the precision (b) and the sensitivity (c) for predicting the academic performance of the students based on the original and the bootstrapped data approaches.

confidence interval results. Whenever the confidence interval error bars do not overlap, as clearly illustrated in Figure 5.8 (b and c), for the precision and the sensitivity analysis, then, this means that the two datasets are statistically significant.

5.5 Summary

The prediction of the academic performance of students has been increasingly emerging in the educational field as it is now possible to transform huge amounts data into useful knowledge that can be utilized to enhance education by making appropriate interventions at an early stage.

However, this is a very difficult task in the educational data mining field because of the issues associated with data. Usually, educational datasets have missing, inaccurate, imbalanced data, which are also very common issues in all the other research fields. Learning from imbalanced data requires approaches and techniques to transform such data into useful knowledge (He and Garcia, 2009). To this end, a resampling approach was provided to learn "Ordered Bayesian Networks" with bootstrap aggregating (bagging). This approach was adopted to tackle the imbalance issue with educational datasets, thereby providing a more accurate educational predictive model which can be updated with new evidence as the student progresses through each year of study.

The objective of this chapter was to model an "Ordered Bayesian Network" for predicting the performance of students and the early detection of students at risk of failing or dropping a module based on progression data. This model can be updated with new evidence as the student progresses through each year of study. For this purpose, students' admission, Year 1 and Year 2 data were used incrementally in conjunction with other related attributes to predict the academic performance at Year 3, taking into consideration the imbalanced issue of the educational data. A set of two Bayesian models were learned from the educational time series data. The first was learned from the original data, whereas the second model was learned from the resampled data (via bootstrapping). The obtained BN models were evaluated in terms of predicting more states of students' overall performance from temporal educational data using the two different approaches.

In the current study, important analytically relevant findings were found when comparing the two different data approaches used for learning the DBNs. The results show that more states of student's overall performance were achieved when learning from the bootstrapped data, especially for the minority class which was for detecting the high-risk students. Also, I have demonstrated how the bootstrapped resampling approach enhances overall prediction of student academic performance using time series educational data in DBN. These findings have significant implications for developing education and enhancing students' learning using artificial intelligence.

These findings are intended to be used to differentiate between the different cohorts of

students who perform with similar dynamics and therefore, simplify them to obtain better understanding on students' performance. Further experimental works are needed to explore the extension of these Bayesian models with the investigation of latent attributes, with the aim of capture hidden factors that may influence the dynamics of students' academic performance.

In the following chapter, the proposed methodology, especially the ordered BNs, is compared with other classification approaches, including deep learning to explore the explicit modelling of bias in educational classifiers. Also, the identified BNs will be examined further to obtain the influence strength between the features in order to Identify the most important ones for the reasoning of the class. This is important to compare other methods with the bootstrapping approach and being more precise in bootstrapping time series data.

Chapter 6

Ethical Decision Making in Machine Learning Classifiers with Probabilistic modeling and Deep Learning methods

In this chapter, the impact of biased features in educational datasets is explored to generate transparent ML models for unbiased classification. This includes the experimental setup for learning the predictive classifiers with Bayesian Networks (BNs) so that the impact of any biased features can be assessed and the use of Multi-label Deep Convolution Neural Networks (CNNs) that are designed to remove any biases.

6.1 Introduction

Explainable AI (XAI) is behind the development of many real-world successful applications. It concerns 'understanding' the functionality of a particular classifier to seek better knowledge on the behaviour of a black box model (Holzinger et al., 2017). It also involves producing new explainable AI systems, while enabling users not only to understand, but also, to trust these systems (Gunning, 2017). Indeed, several black-box algorithms have been implemented for categorising the correlation between the input data and the output (Jia et al., 2018). However,

classifying data based on these algorithms without an explanation raises numerous concerns, one of which is the bias of decisions made by classifiers. This can be because of biases that were existing in the datasets due to the sampling approach or due to the way decisions were made historically within the data resulting in biased labels (Agurto et al., 2010).

Hence, classification accuracy and explainability must be balanced. Therefore, both greatly influence the decisions made when implementing AI models. A common solution to identifying which features are influencing a decision is to detect conditional independencies by using Probabilistic Graphical Models (Pearl, 2011), e.g. implementing Bayesian Networks (BNs). This provides a transparent explanation as to how a classification has been made, and therefore highlights the use of unwanted dependencies (e.g. involving sensitive variables such as gender or ethnicity). However, removing these relationships/features may not remove "proxies" for sensitive variables (for example a postcode may be a proxy for household income) and so we need to ensure that all influences concerning sensitive features are handled.

In this chapter, a number of approaches to classification using different features selections were explored for making 'performance prediction'. As stated earlier the datasets may include proxies which might implicitly reflected sensitive features and here I show how the deep Convolutional Neural Network (CNN) implemented in the context of Human Activity Recognition in (Jason, 2018) can be adjusted to learn a representation that does not incorporate these biased features and proxies. The learned CNNs can therefore remove any proxies and also improve prediction.

In detail, three experiments have been compared to investigate the modeling of bias during classification. The first experiment was mainly determined to predict the performance of the students using the original feature space with a standard deep multi-label CNN. Whereas in the second experiment, the prediction has been made after removing the sensitive features from the feature space to capture the influence of those features on the classifier performance. The impact of the sensitive features were explored explicitly by using a BN to make a transparent classification decision – by exploring the discovered BN structure to identify the Markov Blanket which demonstrates which features are used for classification. Finally, in the third experiment, a deep Multi-label ConvNet was used where the sensitive features/variables were

considered as outputs to make them independently correlated to the class feature and remove the effect of proxies in other features. These experiments have been explored therefore to show:

- The features that are independent of the class so we ensure that the fundamental dependencies on the posterior probability of the class are not removed.
- How the deep Multi-label CNN improves the accuracy of the BNs.
- How the improved CNN can remove any proxies (and also improve prediction).

Exploring these constraints enable the deep learning classifiers to perform transparent models for ethical decision making while predicting student performance.

In this connection, the first section provides the BN classification methodology used for the identification of the class-dependent features having identified the sensitive features, as well as the proposed approach for explicitly representing the redundant dependencies with a Deep Multi-label ConvNet. The remaining sections of the chapter proceed as follows: the experimental results for feature importance identified by the BNs, followed by provision of the classification results obtained using the BN and deep multi-label ConvNets and finally, in the conclusion there is a summary of the findings for modeling the biased features for maximizing classification.

6.2 Methods

The technical objective of this chapter is to explicitly represent the unwanted correlations between the educational features for unbiased prediction of student academic performance. To evaluate the proposed method, 1D multi-label ConvNets were implemented with the Keras (Gulli and Pal, 2017) and TensorFlow Python libraries using Google Colaboratory (Google CoLab). CoLab is a project launched recently (in 2018) by Google to support deep learning research on cloud. It is based on the Jupyter notebook, which runs Python 3, with a pre-configuration of deep learning libraries, such as Keras, Matplotlib, and TensorFlow.

6.2.1 Dataset Description and Correlation Identification

A real-world educational time series dataset was analysed for evaluating the proposed approach in this chapter. This dataset was also used in Chapter 5 to learn an Ordered Bayesian Network via a bootstrapping approach to predict student performance. However, more students' records (instances) were included for the experimental works of this chapter to avoid over-fitting the deep learning classifiers. This means that the model will literally only work well for the set we trained, making the learning of the layers pointless. Also, deep learning needs a lot of training data owing to the large number of parameters needing to be tuned by a learning algorithm. Therefore, the sample used in this chapter included 491 student entry and performance records for the academic years 2012-2016, respectively.

Each progression trajectory determined one student's achieved grades in Year 1, Year 2, and Year 3 at the university, as well as his/her application (admission) data such as: student demographics, previous educational institution and parent education level, which was considered as temporal data (see Table 6.1). The total number of features in the feature space was 34. Here, a three-class problem was created to predict the overall performance of a student (Low Risk: 0, Medium Risk: 1, or High Risk: 2). These classes were set explicitly to detect the 'High Risk' students based on their overall grades and other factors, so that early interventions could be provided.

Table 6.1 presents basic statistics of the educational time-series features used for this experiment, including the mean (μ), variance (σ^2) and StdDev (σ). These statistical measurements provide an insight on the quality and variability of the features used, especially that the sample size in this experiment is quite limited with 491 student records. Considering these statistics give also insights into the initial properties of each feature. As we can observe from this table, the standard deviation values show how consistent is the feature from the mean. Most of the educational features have good spread around the mean.

A step was exploited further to identify the correlation matrix of the educational features based on the multi-variate Gaussian distribution. This correlation matrix provides any possible correlations of each feature that clustered around the mean. This step was mainly determined to obtain an initial insight on the correlations between the features and the class 'Student

Table 6.1: Basic statistics of the educational time-series features used in this experiment.

Feature	**Mean**(μ)	**Variance**(σ^2)	**StdDev**(σ)	**Min**	**Max**	**Count**
Gender	0.170	0.141	±0.376	0	1	491
Age_on_Entery	0.215	0.206	±0.454	0	2	491
Ethnicity	1.679	0.926	±0.962	0	4	491
Disability	8.088	6.059	±2.461	0	9	491
Nationality	0.782	2.655	±1.629	0	6	491
Country of Domicile	0.093	0.281	±0.530	0	6	491
Country of Birth	1.286	4.189	±2.047	0	6	491
Entry Year	1.947	0.838	±0.915	0	3	491
Fee Status	0.066	0.062	±0.249	0	1	491
Previous Ed Estab LEA	3.501	6.947	±2.636	0	8	491
Socio Economic Class	2.714	5.896	±2.428	0	10	491
Been In Care	1.963	0.052	±0.228	0	2	491
Mature	0.894	0.095	±0.308	0	1	491
Parents Been In HE	1.194	1.454	±1.206	0	3	491
Route Code	3.228	2.278	±1.509	0	7	491
Current Student?	0.371	0.234	±0.484	0	1	491
CS1004	1.308	1.187	±1.089	0	6	491
CS1005	0.934	1.121	±1.059	0	6	491
CS1803	0.355	0.554	±0.744	0	6	491
CS1805	1.090	1.428	±1.195	0	6	491
CS1809	0.899	1.054	±1.026	0	6	491
CS1810	0.844	0.989	±0.994	0	6	491
CS1811	1.263	1.529	±1.237	0	6	491
CS2001	1.244	2.685	±1.639	0	7	491
CS2002	1.607	1.830	±1.353	0	7	491
CS2004	1.806	2.465	±1.570	0	7	491
CS2003	1.568	2.954	±1.719	0	7	491
CS2005	1.881	2.345	±1.531	0	7	491
CS2555	0.889	0.519	±0.721	0	6	491
Temporal profiles	2.973	2.228	±1.493	0	4	491
Initial cog. Style	2.095	1.523	±1.234	0	4	491
Year1 cog. Style	2.180	1.318	±1.148	0	4	491
Year2 cog. Style	2.210	1.352	±1.163	0	4	491
CLASS/Student Performance	0.745	0.579	±0.761	0	2	491

Performance'. Identifying these correlations will indicate the importance of the relationship between features, therefore it can be used as a basic quantity for modeling the unbiased classifiers. Furthermore, this process shows the presence of any relationships between the features and the class, so to ensure that we are not removing any of the fundamental features. However, due to the large feature-space of the educational data, Table 6.2 presented only the correlations of the features on the class 'Student Performance'. The table presents the correlation values identified from the correlation matrix of the educational features between the class feature 'Student Performance' and all the other features of the feature-space.

Table 6.2: Feature correlation values of the class and the other features, obtained from the correlation matrix of the multi-variate Gaussian distribution.

	Student Performance
Gender	-0.081
Country of Birth	-0.057
Mature	-0.047
Been In Care	-0.039
Current Student?	-0.017
Parents Been In HE	-0.007
Socio Economic Class	-0.005
Disability	0.003
Fee Status	0.005
Country of Domicile	0.006
Route Code	0.011
Previous Ed Estab LEA	0.021
Nationality	0.024
Age_on_Entery	0.036
Temporal profiles	0.039
Intial cog. Style	0.071
Ethnicity	0.077
Entry Year	0.080
Year1 cog. Style	0.129
Year2 cog. Style	0.148
CS1803	0.259
CS2555	0.268
CS1810	0.348
CS1809	0.386
CS1005	0.494
CS1811	0.498
CS1004	0.567
CS1805	0.575
CS2004	0.616
CS2002	0.616
CS2001	0.630
CS2005	0.663
CS2003	0.721

From this table, it is apparent that the features were sorted from the less correlated to the most correlated. There is some evidence that the admission (application) features may have a low impact on predicting the class. Some of these features have negative correlations on student performance but these are all very small. These were Gender, Country of Birth, Mature, Been In Care, Current Student, Parents Been In HE, Socio Economic Class. Having those negative correlations in the dataset determine that the features do impact the class but in a negative way in which they may represent causality for predicting the class, but no existing

patterns can be indicated. Therefore, the negative correlations can be eliminated or projected as outputs without decreasing the performance of the prediction results.

6.2.2 Identifying Feature Dependency and Importance with BNs

A constraint-based feature selection algorithm was implemented for this work to identify feature importance from the graphical model of BN. For implementing this, first, the Parents and Children (PC) algorithm was exploited (Spirtes et al., 2000) to learn the BN structure of the feature space via discovering the probabilities of the parents and children's nodes for all features. This process imposes with an assumption that the prior probability of the BN structure is related to the features given the probability of such structure. This implementation was performed through the MXM Package in R, as it includes functions for learning BNs, performing feature selection and cross validation (Lagani et al., 2016). The MXM was also used as it characterises the target and predictor features by performing several conditional independence tests with the MMPC (Max-Min Parents and Children) algorithm (Brown et al., 2004) in order to identify the significant features on predicting the class. Basically, the MMPC Algorithm encodes several conditional independence tests to identify the irrelevant features. The final features, in our case the signature or the significant features for predicting the class/student Performance will be identified after all those elimination processes as the MMPC outputs them as signatures.

Essentially, the MMPC() inputs the objects of the target feature 'class' and educational dataset. Assigning the target feature to the MMPC() will fit the target and ensure that the signature feature does not contain the "class" feature. After encoding this algorithm, the MMPC() has identified the signature on predicting the class as it can be seen below (the output results), which highlights the "CS2003", "CS1805", "CS1811", "CS2004", "CS1004", "CS2005", "CS1809", "CS2555" and "CS2001" contents as features of high importance in relation to the selected target feature "class" content.

```
> summary(mmpcobject_learning.test_CLASS)
    Length      Class       Mode
         1 MMPCoutput         S4
> mmpcobject_learning.test_CLASS@selectedVarsOrder
[1] 27 20 23 26 17 28 21 29 24
> colnames(learning.test_dataset)[27]
[1] "CS2003"
> colnames(learning.test_dataset)[20]
[1] "CS1805"
> colnames(learning.test_dataset)[23]
[1] "CS1811"
> colnames(learning.test_dataset)[26]
[1] "CS2004"
> colnames(learning.test_dataset)[17]
[1] "CS1004"
> colnames(learning.test_dataset)[28]
[1] "CS2005"
> colnames(learning.test_dataset)[21]
[1] "CS1809"
> colnames(learning.test_dataset)[29]
[1] "CS2555"
> colnames(learning.test_dataset)[24]
[1] "CS2001"
```

This process can therefore help to observe the Markov Blankets (MB) of the sensitive features to represent the optimal set of features for predicting a given target feature (Koller and Sahami, 1996). In detail, the Markov Blanket of a feature shows the sets of parents, children and the parents of the children nodes as constructed by the graphical structure of the BN (Yaramakala and Margaritis, 2005) to shield the target feature from the other features in the network. For instance, the Bayesian Network in Figure 6.1 highlights the Markov Blanket of a student performance feature. The MB of this feature (red node) therefore consists of seven highlighted features as CS1805, CS1811, CS1004, CS2003, CS2002, CS2001, CS2005 in which they protect student performance node from the effect of the other features outside this MB. The identification of these features is important to ensure that we are not handling any important characteristics of the class feature when applying the deep learning approach. These findings of the MB confirm the other results of the signature features learned by the MMPC algorithm which are indicating the features of high importance in relation to "Student Performance" content.

Since the Markov Blankets of the features can be extracted from the structure of the learned BN, a step was performed further to identify the sensitive features in the BN. A validation technique is used to explore the probability distributions of the parameters in the BN (Castillo et al., 1997). This step has been performed to investigate the effect of the probabilities of the

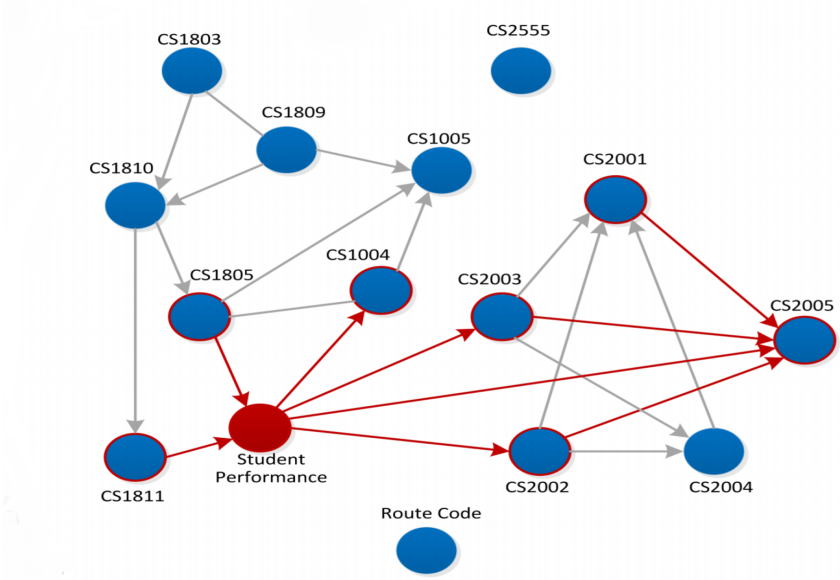


Figure 6.1: Markov Blanket idntification for the feature of student performance (red node) using the BN network structure learned with the PC algorithm. The network represents sets of parents, children and the parents of the children nodes for predicting the target feature.

features on the posterior probabilities of the output feature 'Student Performance'. Identifying the most important features that influence the reasoning of the class will ensure that we are not removing any of the significant dependencies of the BN when implementing the multi-label convolutional network. Also, this step was investigated to explore the importance of the features that could have resulted in classifier bias which plays a crucial role for the knowledge acquisition as well as the validation of the classifiers. To perform this, an algorithm proposed by Kjærulff (Kjærulff and van der Gaag, 2013) was exploited to explore this.

Given a target node 'Student Performance' in our case, the algorithm estimates a set of derivatives of the posterior probability of Student Performance's node over each feature in the BN. These derivatives determine the importance of the features for calculating the posterior probabilities of the output 'Student Performance'.

6.2.3 Learning Deep Multi-label ConvNet Classifiers

Data pre-processing steps were performed on the datasets to be fitted and evaluated by the CNN. Firstly, the one-hot-encoding scheme was applied to the labels/the target features in or-

der to encode them as they were categorical. These target features were Class, Gender, Age on Entry and Ethnicity, with each having two or more categories. With this approach the features have been transformed from categorical to binary. For example, the Ethnicity feature has five different categories, these being Arab, Asian, White, Black and Other. When converting this feature to binary, each category is considered as a feature. Hence, five new features have been added to the feature space to replace the original categorical feature 'Ethnicity', with each having a 0 or 1 value. These new features are Ethnicity _Arab, Ethnicity _Asian, Ethnicity _White, Ethnicity _Black and Ethnicity _other. If the Ethnicity of the student, for instance, is Asian, the value of this feature is considered as 1, whereas all the other Ethnicity features are considered as being 0. This encoding scheme is a very effective approach as it does not involve changing the semantics of the original educational data, especially that the educational data includes very sensitive features. Also, this encoding approach will boost the model since it keeps all the natural aspects for the different features. Therefore, the number of the target features is increased to 13 instead of 3, which are Low risk, medium risk, high risk, Male, Female, Age 17-19, Age 20-24, Age 25+, Ethnicity_Arab, Ethnicity_Asian, Ethnicity_White, Ethnicity_Black, Ethnicity_other.

Subsequently, the training and testing datasets were reshaped into arrays of one dimension, as [len(train/test), nb_features, 1] based on the length and number of features in the datasets, with the new array for training data being [377,43,1] and for testing [114,43,1].

The structure used for performing the deep multi-label CNNs is presented in Figure 6.2, which was based on a simplified model of the 1D CNN for Human Activity Recognition in (Jason, 2018) that predict human movements from sequences of accelerometer data (time-series data). Student data was determined in these experiments as time-series trajectories (see the input layer in Figure 6.2). Each trajectory includes one student's admission data: student demographics, previous educational institution and parent education level as well as his/her grades achieved in Year 1, Year 2, and Year 3, respectively. All were considered as one time-series records based on the time for recording that feature. The main reason therefore for using the Deep 1D multi-label CNN as the educational trajectories contain time-series features. Preparing the data trajectories to capture the time dependent aspects was very

important to achieve accurate and reliable predictions. The feature extraction layers used in these experiments include two 1D convolutional layers, namely Conv1d 1 and Conv1d 2, with 1D 'MaxPooling', 'Dropout' of 0.4 and 'ReLU' activation, with all being inserted after the second convolutional layer. The MaxPooling layer was determined to reduce the dimension of the data, whereas the dropout layer was to avoid overfitting. After this, the output layer was composed of one fully connected layer, with the 'ReLU' activation function and one last dense layer for projecting the labels. This fully connected layer was used to represent the vector of the features of the input. The last layer for the multi-labels CNN was computed with the 'Sigmoid' activation function to determine the probability of the 13 classifiers/labels, it has been set to this function as I have a multi-label classification problem.

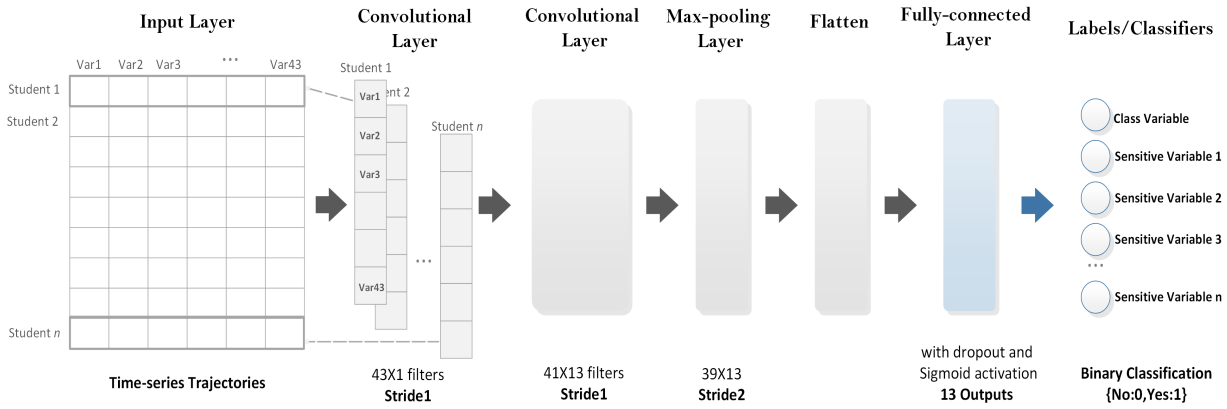


Figure 6.2: Deep 1D multi-label CNN Structure Learning Approach.

A 10-fold cross-validation approach was used in the three experiments due to the limited number of records in the training and testing sets, in terms of 491 time-series educational overall records, 70% (377 records) for training and 30% (114 records) for testing. The training data set was split into 10 folds, with the folds 1-9 being turned into the training data and fold 10 a set for validation (see Figure 6.3) and then each fold is used as a test set iteratively. This approach was performed to tune the CNN hyperparameters and to optimise the classifier. Consequently, the CNN model was trained, while considering all the hyperparameters learned during the 10-fold cross-validation. An evaluation of the CNN model was undertaken next, with

testing data of 114 time-series records (unseen data) being utilised to evaluate its performance. This process was repeated 10 times, with the same test set in each fold. The mean accuracy and standard deviation were reported then to assess the performance of the CNN models.

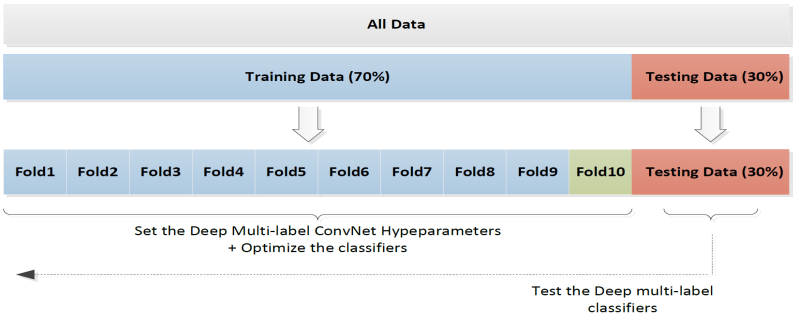


Figure 6.3: 10-fold cross-validation approach for optimizing and testing the CNNs.

Given a set of data $S = \{x^{(i)}, y^{(i)}\}$ containing m time-series patches, where $x^{(i)}$is the training patch and $y^{(i)}$ is the corresponding label for the patch. $y^{(i)} \in \{0, 1\}$, if the $y^{(i)} = 1$ then the $x^{(i)}$ is positive for this particular label or classifier, whereas the opposite for determining the negative label $y^{(i)} = 0$.

Hence, for computing the probability for the labels of the CNN model Z_j was projected. Let ${z_j}^{(i)}$ be the last layer (fully connected) of unit j for the training patches $x^{(i)}$and then, the probability of the label $y^{(i)}$ is calculated by:

$$p\left(y^{(i)} = j|z_j^{(i)}\right) = \frac{e^{z_j^{(i)}}}{\sum_{l=1}^{k} e^{z_l^{(i)}}} \tag{6.1}$$

This was computed for all the labels $y^{(i)}$ of the output ${z_j}^{(i)}$.

The CNNs were trained for 30 epochs, with a batch size of 50 and different values of epoch were tested on the training samples. When optimising the CNNs to 30 epochs, the validation loss was reduced, which indicates that the model's coverage is good when using this value. The CNNs were compiled using the 'binary-crossentropy' loss function and the (Kingma and Ba, 2014) optimiser. The loss function was set to binary-crossentropy as the labels of the CNNs involved binary decisions of 0 and 1. Performing this loss function in the multi-label classification was aimed at deciding whether the case belonged to that label or not.

6.2.4 Evaluation Matrices for Multi-Label Classification

The evaluation of multi-label classification is a very challenging task and the criteria for evaluating this type of classification approach is entirely different than in the traditional classification approach 'the single-label problem'. For example, the most popular evaluation matrices in the single-label problems are Accuracy, Precision, Recall and Roc area specified for each class (Fawcett, 2006). However, the original notion of these measurements cannot capture more than one correct label as the case of the multi-label classification problem, because each instance in the dataset has been connected to one label either corrects or not.

Initially, the evaluation matrices in Multi-Label classification can be categorized into three different approaches depending on the classification problem. These are partitions evaluation, ranking evaluation, and label hierarchy evaluation (Tsoumakas et al., 2009). In this work, the multi-label ConvNet was evaluated with the partitions evaluation approach as the main target was to capture the quality of classifying the labels. Another reason is that with this evaluation approach we can determine how far is the trained network in predicting the labels comparing to the actual ones. With this strategy, the notion partially correct will captured to evaluate the average variation between the predicated and actual labels over all the instances in the test set. Consequently, Godbole et. (Godbole and Sarawagi, 2004) suggested a set of definitions for measuring accuracy, precision, recall, and F1 measure in consideration with the notion of being partially correct.

Let T be a dataset containing n multi-label instances (X_i, Y_i),

$1 \leq i \leq n, (X_i \in X, Y_i \in Y = \{0,1\}\, K)$, with a label set $L, |L| = K$. Let h be a classifier for a multi-label classification model and $Z_i = h_{(xi)} = \{0,1\}$, k be the set of labels that are predicted by the classifier h for the x_i.

Accuracy (ACC): Accuracy is specified as the fraction of the predicted labels to the total labels (the predicted and the actual) for each instance. Overall accuracy is calculated then by the average among all instances.

$$ACCURACY, ACC = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \tag{6.2}$$

Precision (PREC): Precision is the fraction of predicted labels to the total labels of actual ones, averaged among (over) all instances.

$$PRECISION, PREC = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Z_i|} \tag{6.3}$$

Recall (REC)/Sensitivity (SN): Recall is the correctly predicted labels to the total number of predicted labels, averaged among all instances.

$$RECALL, REC = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i|} \tag{6.4}$$

F1-Measure (F1): is the mean of Recall (REC) and Precision (PREC) over all predicted labels.

$$F1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2\,|Y_i \cap Z_i|}{|Y_i| + |Z_i|} \tag{6.5}$$

Therefore, when calculating all the above-mentioned matrices for the Multi-label classification model, the overall performance can be simply observed as in the single-label classification case. The higher average/mean results of ACC, REC, SP, PREC, and F1, the better of overall performance accuracy for the learning classification algorithm.

6.3 Experimental Results and Discussion

The experimental work consists of the influence strength and sensitivity analysis of the BN results followed by the deep Multi-label ConvNet for representing the class as well as the biased features. Thus, the results obtained from the BN algorithm was evaluated first for finding feature correlations and importance from the MB of the learned BN. Subsequently, in this section a detailed evaluation of the Multi-Label CNNs is presented to capture the significance of the proposed approach in this work.

6.3.1 Influence Strength and Sensitivity Analysis of BN

The learned BN from the temporal educational data is shown in Figure 6.4. This network detects the correlations between the features to predict the class 'Student Performance' as well as the influence strength identified between the features. The influence strength is calculated from a conditional probability tables (CPT) of the child nodes. Essentially, this can be achieved via performing some sort of distances between the probability distributions of the child nodes and the parent nodes. Table 6.3 provides the influence strength values that been identified between the parents and child nodes based on three different approaches of measuring the distances. These were Average, Maximum and Weighted. As we can noticed from this table the results of the three different measurements are quite similar. As a result, a representation of the weighted approach was selected to show a graphical influence strength on the network between the parent nodes and the child nodes based on the marginal probability distribution (more information about the exploited BNs with the other influence strength methods is shown in Appendix B.2).

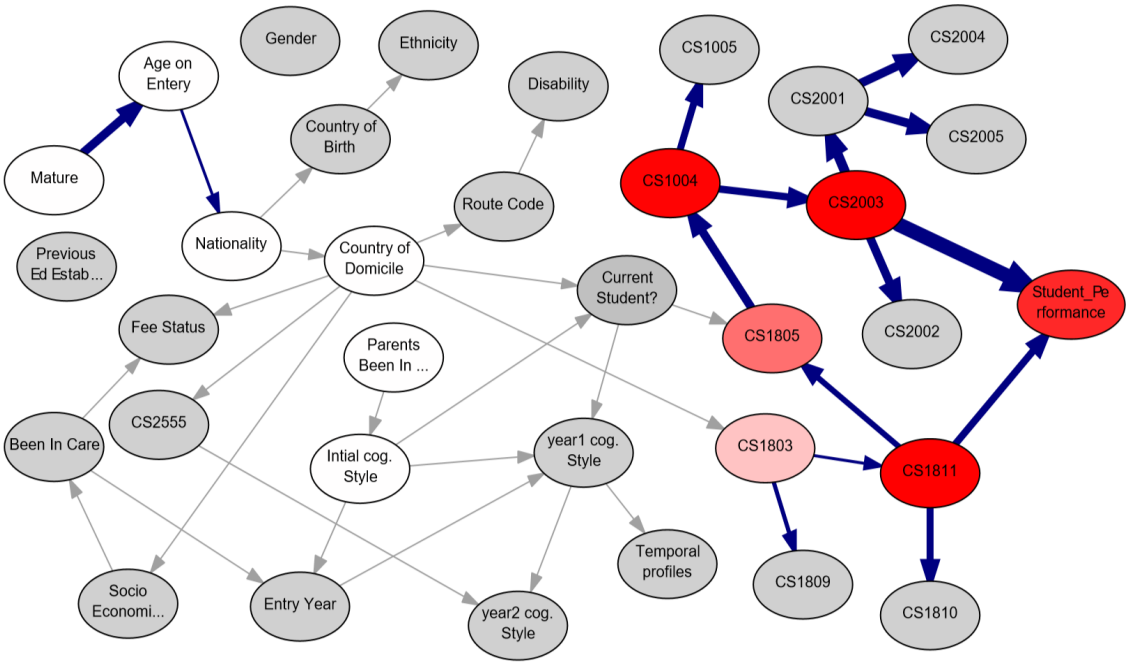


Figure 6.4: Influence strength and sensitivity analysis of the BN based on the marginal probability distribution of the parent and child nodes.

Table 6.3: Influence strength values of the correlations between the parents and child nodes.

Parent	**Child**	**Average**	**Maximum**	**Weighted**
year1 cog. Style	Temporal profiles	4.39E-17	4.39E-17	4.39E-17
year1 cog. Style	year2 cog. Style	4.39E-17	4.39E-17	4.39E-17
Socio Economic Class	Been in Care	4.79E-17	6.80E-17	4.79E-17
Route Code	Disability	9.81E-18	1.96E-17	9.81E-18
Parents Been In HE	Initial cog. Style	0	0	0
Nationality	Country of Domicile	5.03E-17	8.09E-17	5.03E-17
Nationality	Country of Birth	2.90E-17	4.81E-17	2.90E-17
Mature	Age on Entry	0.426724	0.848928	0.163045
Intial cog. Style	Entry Year	7.85E-17	1.57E-16	7.85E-17
Intial cog. Style	Current Student?	2.78E-17	5.55E-17	2.78E-17
Intial cog. Style	year1 cog. Style	0	0	0
Entry Year	year1 cog. Style	1.96E-17	3.93E-17	1.96E-17
Current Student?	CS1805	5.43E-17	9.93E-17	5.11E-17
Current Student?	year1 cog. Style	0	0	0
Country of Domicile	Route Code	0	0	0
Country of Domicile	Socio Economic Class	4.24E-17	6.51E-17	4.24E-17
Country of Domicile	Fee Status	5.55E-17	1.11E-16	5.55E-17
Country of Domicile	Current Student?	0	0	0
Country of Domicile	CS1803	9.87E-17	1.59E-16	9.94E-17
Country of Domicile	CS2555	1.02E-16	1.68E-16	1.05E-16
Country of Birth	Ethnicity	6.09E-17	7.93E-17	6.69E-17
CS2555	year2 cog. Style	7.40E-17	1.42E-16	8.73E-17
CS2003	CS2001	0.313901	0.656048	0.196694
CS2003	CS2002	0.291389	0.688189	0.16124
CS2003	Student Performance	0.311993	0.593084	0.257858
CS2001	CS2004	0.322836	0.706863	0.163135
CS2001	CS2005	0.277733	0.73375	0.175944
CS1811	CS1805	0.1783	0.398358	0.128716
CS1811	CS1810	0.227302	0.516141	0.138164
CS1811	Student Performance	0.150816	0.217358	0.140775
CS1805	CS1004	0.270798	0.60108	0.164474
CS1803	CS1811	0.200368	0.628067	0.065903
CS1803	CS1809	0.224276	0.628067	0.097312
CS1004	CS1005	0.268009	0.499605	0.154704
CS1004	CS2003	0.232062	0.539674	0.149187
Been In Care	Entry Year	0	0	0
Been In Care	Fee Status	0	0	0
Age on Entry	Nationality	0.204849	0.596554	0.06575

The thickness and the colour of the arcs in Figure 6.4 indicates the influence strength between the parent nodes and the child nodes. Therefore, the highlighted 'blue' arcs have

the strongest influence in the network. Features that were assigned to be "sensitive" were 'Gender', 'Ethnicity' and 'Age on Entry'. Notice that based on this structure it appears that these variables have no direct influence on the class 'Student Performance'.

The BN in Figure 6.4 is also indicating the link strength analysis of the significant parameters on the class node. For example, the red nodes are the significant parameters for calculating the posterior probability distribution of the class 'Student Performance'. Hence, we can clearly distinguish the important dependencies and their correlations with the class node. It is clear that the posterior probability of the class 'Student Performance' was learned from some of Year 1 and Year 2 modules. This result was achieved due to the nature of the features in Year 1 and 2 and their relation to student overall performance, which was directly impact the result. However, the grey and transparent nodes did not include parameters for calculating the posterior probability of the target class. From this BN, it can be observed that most of the grey-coloured nodes contained admission features that did not directly impact on the overall performance of the students. Whilst the BN included categorical features (the admission features), these did not influence the results since they were encoded before the BN learning process. This finding provides evidence supporting the results obtained earlier from the correlation matrix that the features of the Year 1 and 2 modules are mainly correlated to the class. Although the results are applied in context of Brunel's data, they show great influence of students' achievement in Year 1 and Year 2 to their overall performance in Year 3.

These results provide additional evidence with respect to modelling time-series independent features to the prediction of future performance. It also contributes to the EDM by offering year by year profiling of students according to their learning developments or characteristics. This can be very useful therefore to provide personalised education especially for the low performing students. These results can be widened to exploit cognitive models that reflect students' skills and the obtained knowledge.

To measure the performance of the BN model, however, the accuracy, sensitivity, precision and F1- Score were evaluated for predicting the class variable 'Student Performance'. The model achieved 0.76 overall accuracy results for predicting the three classes: Low Risk, Medium Risk, and High Risk (see Table 6.6). Therefore, these measurements wew identified so a

fair comparison can be obtained to investigate the importance of the proposed model when implementing the deep multi-label CNNs. However, the accuracy for the BN as more evidence was added has been investigated further: firstly, for predicting the class entering just the application evidence, then the Year 1 results, then Year 2. This process was determined to investigate that how influential the sensitive variables are on the prediction results (see Figure 6.5). In other words, applying these evidences constrains will score the mutual information between each variables and performance given this network.

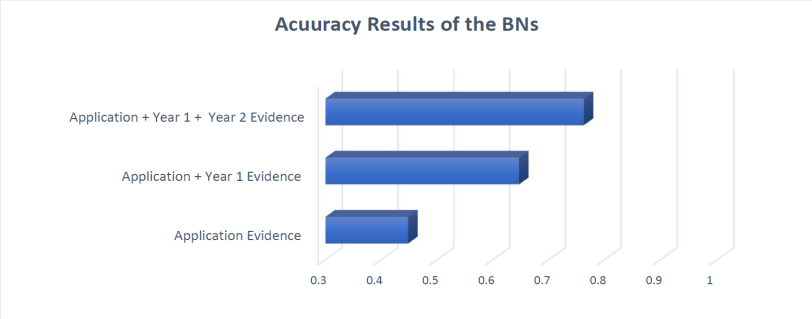


Figure 6.5: Bayesian networks accuracy comparison for predicting the class variable 'Student Performance'.

6.3.2 Deep Multi-label ConvNet Models Evaluation

In this section, I present evidence that with the use of the proposed method of the deep multi-label CNNs the performance of classifiers can be improved, particularly with the constraint that they should not be influenced by the biased or the redundant features. With this constraint, the classifiers will be learned from the strong dependencies between the features in the feature space. I investigate further the redundant features (i.e. those that have the lowest influence on the class) used for classification in experiment 3.

Keeping all the parameters of ConvNet, apart from the final dense layer the 'output layer' in experiment 3 to project 13 classifiers instated of 3, a Multi-label ConvNet was trained using the 10-fold cross-validation for the training set, with the output layer predicting the redundant features as well as the original class features. By doing so, these constraints forced the CNN to learn only the strong feature dependencies of the feature space, which maximised the performance of the predictive classifiers.

In Figure 6.6, the 10-fold cross-validation accuracy results using the three approaches are provided. Firstly, to classify the class variables only using the original feature-space. Secondly, to classify the class with removing the sensitive variables from the dataset. Finally, to classify all the redundant 'sensitive' variables as well as the class variables. The results reveal that when projecting the sensitive variables (in experiment 3: the Multi-label ConvNet for classifying all the sensitive variables (13 Labels)) the accuracy has improved dramatically. Therefore, the results confirm that the transferred unwanted features to the output layer are useful for enhancing the predictive model.

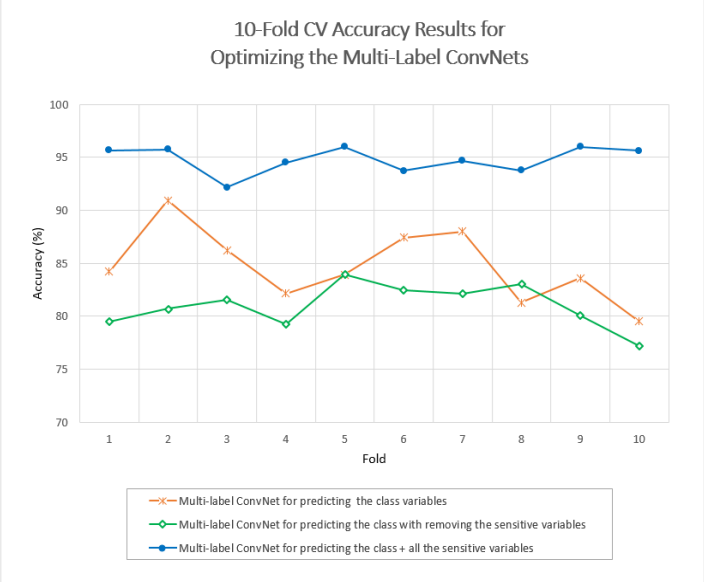


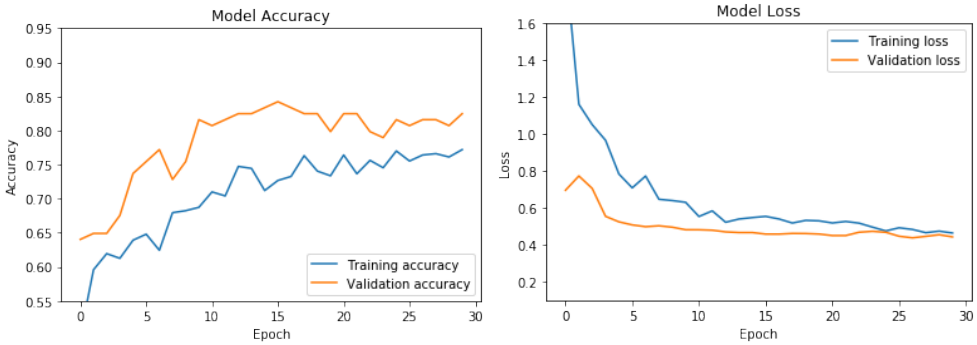
Figure 6.6: 10-fold cross validation accuracy results. It shows the accuracy per fold for validating the multi-label CNN models using the three experiments mentioned in the methodology section.

The CNN average accuracy and loss results were then plotted in Figure 6.7 for all experiments to explore the validation of the method used when applying some constraints (in experiment 3) to represent the sensitive variables in the data set. From these figures, it can be observed that the validation average accuracy has improved dramatically from 84% in Figure 6.7 (a) to around 95% in Figure 6.7 (c) when projecting the class variables as well as all the redundant or sensitive variables. In addition, it is clear that the interpretation of the loss for optimizing the three models have decreased as the models are improved, in which it indicate the behaviour of the models on the validation sets. For instance, the model loss of experiment

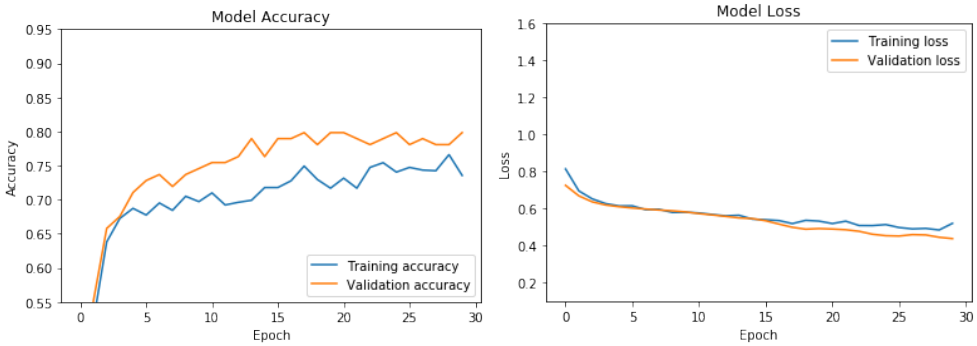
3 in Figure 6.7 (c) show a sharp drop in the loss value compared to experiment 1 and 2 as a result of the improvement of the classifier performance. Despite the data set having very few cases to optimize the classifiers, the model shows an impressive prediction result when using the proposed approach introduced in this chapter.

The confusion matrices testing results were further explored for predicting the unseen cases using the three different classification approaches (see Table 6.4). The ConvNet algorithm returns the probabilities for the predicted cases in each classifier. To determine the classes, a classification threshold was set to 0.5, whereby those cases above the threshold were considered 1 and 0 for the results below 0.5. From Table 6.4, it can be observed that the true positive (TP) results of the 'Low Risk', 'Medium Risk', and 'High Risk' classifiers have been improved in the third approach (experiment 3) when projecting the classes alongside the sensitive variables. For example, the correctly classified high risk students in experiment 1 was 220 students, but when constricting the classifiers of experiment 3 to detect the sensitive features (biased ones) alongside the class features, the predicted positive cases was improved to 376 students. However, misclassification for some of the sensitive labels can be noticed in Table 6.4, these being: Age 25+, Ethnicity Arab and Ethnicity-other due to the limited number of positive cases for these, but this does not affect the overall classification model accuracy as the technical target was to perform unbiased classification for the class labels.

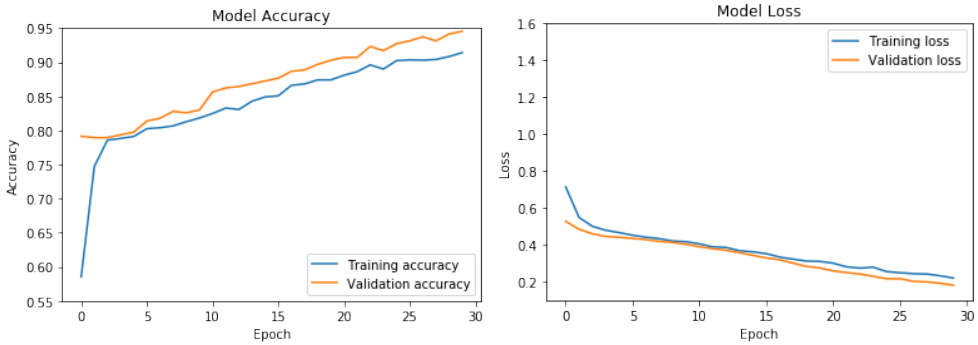
Besides observing good classification results in the confusion matrices, further evaluation analysis was performed to obtain classifiers accuracy (ACC), sensitivity (SN), specificity (SP), precision (PREC) and F1-score for the three approaches (see Table 6.5). With these detailed evaluation matrices, the usefulness of projecting the unwanted dependencies with the class variables for performing unbiased classification can be observed. The models overall testing accuracy are indicated therefore in Table 6.6 for predicting the labels of the class variable (Low, Medium and High Risk) with the BN, the original Multi-label ConvNets and the proposed approach of Multi-label ConvNet for predicting all the sensitive variables. As a result of removing the sensitive features from the feature-space in experiment 2, the model accuracy was decreased from 84.74% to 80.99%. This reduction mainly resulted owing to the removal of some of the dependent parameters and proxies that were associated with the sensitive features. In



(a) Experiment 1: CNN trained and validated with the original feature space to predict the class variables.



(b) Experiment 2: CNN trained and validated with removing the sensitive variables from the feature-space to predict the class.



(c) Experiment 3: CNN trained and validated with some constrains to predict the class variables as well as the sensitive variables.

Figure 6.7: Accuracy and loss results from experiment 1,2 and 3 for the average performance predictive models showing the CNN models for predicting: (a) the class variables only, (b) the class virables with removing the sensitive variables from the feature space and (c) the class variables as well as all the sensitive variables.

Table 6.4: Confusion matrices testing results for the multi-label ConvNets of predicting the labels of the class variable (Low, Medium and High Risk) vs the class + the sensitive variables (with threshold=0.5).

	Experiment 1: CNN for predicting the labels of the class variables				**Experiment 2: CNN for predicting the class variables with removing the sensitive variables**				**Experiment 3: CNN for predicting the class variables and the sensitive variables (proposed approach)**			
Predicted Labels	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**	**TN**
Low Risk	360	30	80	670	354	47	119	620	372	18	7	743
Medium Risk	250	120	130	640	217	153	138	632	345	15	24	752
High Risk	220	170	0	750	337	53	159	591	376	14	0	750
Male	-	-	-	-	-	-	-	-	950	0	82	108
Female	-	-	-	-	-	-	-	-	116	74	0	950
Age 17-19	-	-	-	-	-	-	-	-	870	0	23	247
Age 20-24	-	-	-	-	-	-	-	-	212	28	26	874
Age 25+	-	-	-	-	-	-	-	-	0	30	0	1110
Ethnicity-Arab	-	-	-	-	-	-	-	-	0	80	0	1060
Ethnicity-Asian	-	-	-	-	-	-	-	-	536	4	43	557
Ethnicity-White	-	-	-	-	-	-	-	-	244	66	60	770
Ethnicity-Black	-	-	-	-	-	-	-	-	20	130	3	987
Ethnicity-other-Mixed	-	-	-	-	-	-	-	-	0	60	0	1080

contrast, the model accuracy results in experiment 3 reveal that when projecting the unwanted dependencies of the biased features to the output layer to classify all the sensitive variables (13 Labels) the model accuracy increased to 96.67%. I further captured the skill of the predictive models with the standard deviation of the mean accuracy. The standard deviation in Table 6.6 has been decreased when using the proposed approach of this paper from +/-3.263 to +/-0.622. This measurement shows the importance of plotting the sensitive features as the standard deviation has been reduced, which proof that the mean is not far from the predictions. Hence, the results obtained in Tables 6.4, 6.5 and 6.6 reveal a significant finding that confirms that the transferred sensitive features to the output layer are useful for enhancing the temporal predictive model. The CNN in experiment 3 removed the proxies of the sensitive features, which in turn, improved the performance prediction for the target classes 'Low Risk', 'Medium Risk', and 'High Risk'.

Figure 6.8 provides a chart showing the improvement in the model accuracy, sensitivity and F1-Score when predicting the biased features/sensitive variables using the multi-label

Table 6.5: Evaluation matrices of Multi-label ConvNets for predicting the labels of the classifiers with the three experimental approaches using the test data with 'threshold= 0.5'.

	Experiment 1: Multi-label ConvNet for predicting the labels of the class variables					**Experiment 2: Multi-label ConvNet for predicting the class variables with removing the sensitive variables**					**Experiment 3: Multi-label ConvNet for predicting all the sensitive variables**				
Predicted Labels	**ACC**	**SN/REC**	**SP**	**PREC**	**F1- Score**	**ACC**	**SN/REC**	**SP**	**PREC**	**F1- Score**	**ACC**	**SN/REC**	**SP**	**PREC**	**F1-Score**
Low Risk	0.90	0.82	0.96	0.92	0.86	0.85	0.75	0.93	0.88	0.81	0.98	0.98	0.98	0.95	0.94
Medium Risk	0.78	0.66	0.84	0.67	0.67	0.74	0.61	0.80	0.59	0.60	0.96	0.93	0.98	0.96	0.94
High Risk	0.86	1.00	0.81	0.56	0.36	0.81	0.68	0.92	0.86	0.76	0.96	1.00	0.98	0.96	0.98
Male	-	-	-	-	-	-	-	-	-	-	0.93	0.92	1.00	1.00	0.96
Female	-	-	-	-	-	-	-	-	-	-	0.94	1.00	0.93	0.60	0.75
Age 17-19	-	-	-	-	-	-	-	-	-	-	0.98	0.97	1.00	1.00	0.98
Age 20-24	-	-	-	-	-	-	-	-	-	-	0.95	0.89	0.97	0.88	0.88
Age 25+	-	-	-	-	-	-	-	-	-	-	0.97	N/A	0.97	0.00	N/A
Ethnicity-Arab	-	-	-	-	-	-	-	-	-	-	0.93	N/A	0.93	0.00	N/A
Ethnicity-Asian	-	-	-	-	-	-	-	-	-	-	0.96	0.93	0.99	0.99	0.96
Ethnicity-White	-	-	-	-	-	-	-	-	-	-	0.89	0.80	0.93	0.79	0.79
Ethnicity-Black	-	-	-	-	-	-	-	-	-	-	0.88	0.87	0.88	0.13	0.23
Ethnicity-other-Mixed	-	-	-	-	-	-	-	-	-	-	0.95	N/A	0.95	0.00	N/A

Table 6.6: Evaluation matrices for predicting the labels of the class variable (Low, Medium and High Risk) with the BN, the original Multi-label ConvNets and the proposed approach of Multi-label ConvNet for predicting all the sensitive variables.

	BN for predicting the class variables					**Experiment 1: Multi-label ConvNet for predicting the class variables**					**Experiment 2: Multi-label ConvNet for predicting the class variables with removing the sensitive variables**					**Experiment 3: Multi-label ConvNet for predicting all the sensitive variables**				
Labels	**ACC**	**SN**	**SP**	**PREC**	**F1- Score**	**ACC**	**SN**	**SP**	**PREC**	**F1-Score**	**ACC**	**SN**	**SP**	**PREC**	**F1-Score**	**ACC**	**SN**	**SP**	**PREC**	**F1-Score**
Low Risk	0.82	0.79	0.46	0.79	0.79	0.90	0.82	0.96	0.92	0.86	0.85	0.75	0.93	0.88	0.81	0.98	0.98	0.98	0.95	0.94
Medium Risk	0.67	0.67	0.44	0.60	0.63	0.78	0.66	0.84	0.67	0.67	0.74	0.61	0.80	0.59	0.60	0.96	0.93	0.98	0.96	0.94
High Risk	0.79	0.79	0.67	0.89	0.84	0.86	1.00	0.81	0.56	0.36	0.81	0.68	0.92	0.86	0.76	0.96	1.00	0.98	0.96	0.98
Average	**0.76**	**0.75**	**0.52**	**0.76**	**0.76**	**0.84**	**0.83**	**0.87**	**0.72**	**0.63**	**0.80**	**0.68**	**0.88**	**0.78**	**0.72**	**0.97**	**0.97**	**0.98**	**0.96**	**0.95**
Model Accuracy (std.)	**76.32% (+/-0.805)**					**84.74% (+/-3.263)**					**80.994% (+/-1.926)**					**96.67% (+/-0.622)**				

ConvNets compared to the BN for predicting the classes only. In this figure, the results for the prediction of the class variables (Low Risk, Medium Risk, and High Risk) using the proposed approach of experiment 3 are provided (the highlighted bars). The model achieves an average accuracy result of 0.97 using this approach. Furthermore, the model reports an average of 0.97 for the sensitivity result, an average of 0.98 for the precision result and an average of 0.95 for the F1-Score result in relation to predicting the class variables. In addition, the SN, SP and PREC charts illustrate the improvements that were achieved as more constrains be added to exploit the classifiers. This can be clarified by how the classifiers were trained from a representation that has not been influenced by the biased features. For example, in the

BN the students were always categorised as 'Low Risk' students when setting the evidence of the gender as 'Female' and Ethnicity as 'White'. In other words, preventing these redundant feature dependencies, such as 'Female' and 'Ethnicity', the classifiers will be guided only to the significant dependencies.

It would be appropriate, therefore, to undertake profiling, such that the predictive models target the academic performance of different cohorts of students. This can help in assessing the students and thus, identifying those who may need more support. However, profiling students using demographical data must be implemented with some constraints to ensure achieving unbiased classifiers. These constraints include observing the influence strength of the demographical parameters/features on students' overall performance based on the marginal probability distribution of the Bayesian Network. Exploring this constraint will enable the users to identify the biased demographic features. This identification, however, will not remove the proxies that are associated with the biased features. Another important constraint has to be determined to handle the less dependent demographic features on students' achievements to ensure removing any associated proxies with these biased features. By taking into account these constraints, the model will be guided to the unbiased features and hence, the performance prediction will be improved.

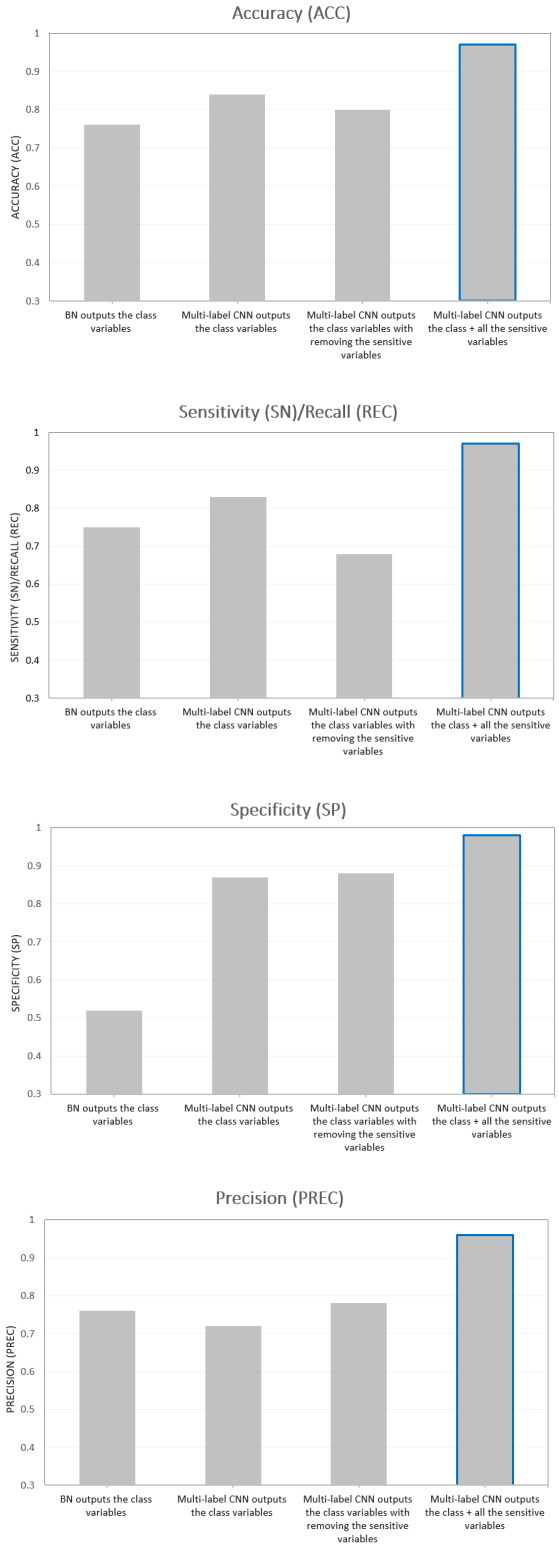


Figure 6.8: Evaluation matrices ACC, SN, SP and PREC for predicting the labels of the class variables (Low, Medium and High Risk) with the BN and multi-label ConvNets approaches

6.4 Summary

The identification of new methods for gaining trust in AI decisions is a primary task for many researchers. Given AI is encountering many application challenges related to the understanding of the technical setup and the explicability of AI systems, in this work, a new dimension to the implementation of transparent classification systems has been explored. A well-known problem in educational AI was addressed in this chapter, where the performance of any classifier is dependent on the quality of the educational features. In fact, Convolutional Neural Network cannot do a miracle when the first learning layers are not appropriate designed to extract good features. I have proposed an approach aimed at increasing trust in AI not just by removing the unwanted features from the feature space, but also, by performing unbiased classification. A classification model consisting of several constraints was identified to make intelligent decisions without being biased by other concepts. These constraints are much stronger than the ensembling bootstrapping approach used in my previous work (Al-Luhaybi et al., 2019) for performing unbiased classification. I have focused on the identification of the biased features, which should be not determined in the classification process in order to make ethical decisions. This exploration is important as it shows that some machine learning techniques can be used in order to force CNN to learn strong feature dependencies in the feature space.

The experimental results have revealed that the deep 1D Multi-label CNN successfully eliminated the biased features dependencies, and result in unbiased classifiers. However, understanding the decisions of AI models is fundamental for the adaptation stage of many real-world AI systems. Further works are required to project intelligent models that are human-interpretable including approaches such as 'Google Explainable AI' tools which explore the weightings within different hidden layers of the network models.

Chapter 7

Conclusion

This thesis involved examining whether non-linear implementations, such as sequential clustering, structure modelling and deep learning, can efficiently exploit explainable and transparent models for ethical prediction of academic performance. The principal aim was to achieve transparent models, while also obtaining strongly reliable accuracy results. This chapter provides the conclusions of the research conducted in this thesis. Firstly, the contributions in the context of the extant literature are considered to show how the findings have added to the state-of-the-art of educational informatics. Then, the limitations of the research are discussed. Finally, future directions for the development of reliable and ethical machine learning classifiers for performance prediction are put forward.

7.1 Thesis Contributions

The thesis presents the implementation, and evaluation of machine learning methods using students time series educational trajectories. It has identified explainable classifiers by exploiting structured learning methods, using ordered Bayesian Networks, with consideration of the time dependent order of the educational features. The methods also target analysing temporal data to identify new temporal features of students' progression, such as cognitive styles and online assessment profiles, while enhancing overall prediction of academic performance for characterising high-risk students. Significant analytical findings emerged when implementing

the predictive models using the temporal trajectories. This is unlike other prediction models, where the models were implemented using some selected educational features. The thesis makes the following contributions.

7.1.1 Temporal Profiling of Time-Series Educational Trajectories

The literature review confirmed that most of the available ML models have been focused on behavioural and engagement aspects of students. There have been very few works conducted on students' assessments and participation based on their online temporal (time-series) progression trajectories. Most of the previous research in EDM consider clustering students used some educational features not the full time-series performance trajectories. This was mainly developed for the purpose to extract students' patterns from the activity counts of the completed assessments to detect students learning behaviours. However, we investigate in this thesis a novel approach to profile the full time-series progression trajectories in order identify new features to enhance predicting the academic performance of students. In this thesis, a distance-based similarity clustering approach using the Dynamic Time Warping (DTW) method was exploited to identify students' temporal cognitive styles. The method is a temporal similarity based clustering method, which involves clustering students' time-series engagement trajectories based on their attendance at lectures and online tests as well as their progression results for Year 1 and Year 2 courses. The main aim of using this method is to capture students' cognitive patterns that could improve the prediction process of student academic performance. The cognitive styles identified managed to explain the performance contrast of the students between the clusters, such that those that included students with low performance (high risk students) are characterised.

Moreover, most of the previous research on education that has considered clustering students has used a selection of extracted features of activity data and not the entire sequential performance trajectory. A predictive model that assesses the performance of university students was identified in this thesis based on profiling students' time-series trajectories of online self-assessments. The principle idea behind the approach is temporal clustering of students' time series data by applying Dynamic Time Warping (DTW) to measure the distance between

their self-assessment trajectories and using this information in conjunction with student admission data to predict overall performance. The predictive model is capable of identifying different profiles of students' online self-assessment, these being classified as: very low, low, medium, high and very high performance profiles. Also, significant improvement to the prediction results was found when adding the generated the students' profiles as an independent attribute in the prediction process. This proved that using the entire assessment trajectories are more insightful than predicting performance with some selected attributes. These findings make major contributions to research on educational informatics and demonstrate the effectiveness of DTW distance-based clustering in profiling students for improved performance prediction.

7.1.2 Ordered Bayesian Network Modelling

A new concept of a structured learning approach was introduced in this thesis, that is, an Ordered Bayesian Network. It is a machine learning method that is implemented using graph theory to learn the joint probability distribution between the features over time, with consideration of the time-dependent order between the different features. It is introduced in this thesis for the case of modelling students' data collected at different times: at admission, Year 1, and Year 2. It is a very important method particularly when considering the causality of the earlier attributes, such as students' performance in Year 1 to the later attributes/modules in Year 2.

The concept of the ordered BN was extended to explore the incorporation of students' cognitive styles and online self-assessment profiles for predicting performance at different stages for the early detection of those at risk of failing or dropping out of the course. Important analytically relevant findings were achieved when learning the structure using the Ordered Bayesian Networks. The findings indicate that more states of student performance were achieved when learning using time dependent features in the ordered network, especially for enhancing the predictions of the minority class which was for detecting at risk students. These results are intended to be utilised to differentiate between the various cohorts of students who perform with similar performance and thus, obtain improved models.

7.1.3 Bootstrapping Temporal Bayesian Models

The research conducted far on educational data has targeted extracting useful patterns for characterizing the learning styles of students or modelling student's performance. Limited experimental works have been conducted to handle the issues that were associated with the quality of the educational datasets. These issues are including imbalanced classes, which is also a very common machine learning problem in all the other research fields. Learning from imbalanced data requires special techniques to convert it into valuable knowledge. A resampling approach was conducted in this thesis to learn "Ordered Bayesian Networks" with bootstrap aggregating (bagging). This approach was implemented to tackle the imbalance issue with the educational datasets, thereby providing a more accurate educational predictive model, which can be updated with new evidence as the student progresses through each year of study. The predictive model shows how the bootstrapped resampling approach enhances the overall prediction of student academic performance using time series educational data. Moreover, the bootstrapped temporal model managed to identify more cases of the minority class of high-risk students. These findings have important implications for enhancing the learning process of educational predictive models using the advances of resampling techniques, such as bootstrapping.

7.1.4 Explainable Machine Learning Models

The majority of the available predictive models for student performance and outcomes are unexplainable, having been developed using black box algorithms that are not transparent in how the predictive model or classifier reached its decisions, e.g. Random Forest, or Neural Network. Hence, the target when using these algorithms by many researchers is to obtain high accuracy results as an important evaluation measurement for algorithm performance, without any consideration as to how reliable are these models for real-world inference. Classifying students using these predictive models, however, raises concerns during the practical and the application process. In particular, the educational data can contain biased features that exist in the datasets, such as students' ethnicity, nationality, gender and so on, which could result biased decisions by the classifier. The transparency of the classification processes and the

decisions made by the predictive classifiers are also important evaluation measurements for gaining trust in classifier decisions.

This thesis has involved exploring linear and non-linear PCA, CNNs and BNs to exploit transparent models for predicting student performance. The main issue addressed is biased classification by black box methods. In particular, there has been exploration of ML methods to perform transparent models using the advances of the BNs and Deep Learning methods. Investigating these different methods enables the users to engage in transparent models for ethical decision making, while improving student performance prediction. These methods involve modelling the biased features during classification using deep 1D Multi-label CNN to ensure that classification decisions are not affected by any of the biased features, such as ethnicity or gender. The findings have revealed that this CNN successfully excluded the biased features dependencies. In addition, the results show a great improvement in the model accuracy when handling the biased features/sensitive variables using the multi-label ConvNets compared to BN for predicting academic performance. These results allow assessment of the predictive models to explicitly representing the unwanted correlations between the educational features for unbiased prediction of student academic performance. Understanding the decisions of AI models is crucial for the adaptation stage of many real-world AI systems.

7.2 Limitations

The results of the experimental works conducted in this thesis were affected by the quality and the volume of the educational data. However, learning robust and accurate predictive models required numerous pre-processing techniques to prepare the educational data trajectories before learning the classifiers. Whilst accurate predictive classifiers were obtained, I will explain in this section the limitations of the research, especially with regards to the data related issues that impacted on the predictive and explanatory power of the predictive models. These shortcomings were as follows:

- **Limited Data Trajectories**

The small sample size of the educational data was resulted due to the time series consideration in preparing the datasets for the experimental works of Chapter 5, in particular, capturing the dynamics of the same group of students in different time slots with their application 'admission' data and Year1 and Year2 performance data. Also, some of the students dropped out of the academic programs they were enrolled on or changed their topic of study, which caused missing performance data for Year 1 or Year 2 CS modules. It was decided to consider only the full time-series data trajectories and ignore the missing ones so as to obtain accurate and robust classifiers.

- **Imbalanced Classes**

The imbalanced classes was another issue that affected learning of the predictive models as well as in identifying the cases of the minority classes. This was caused due to the limited number of high-risk students compared to the low and medium risk ones. The minority class in this thesis was the high-risk class, with those students who obtained low grades or failed (e.g. D, E, and F) in all or most of the modules. Learning from imbalanced classes occasionally results in misclassification and it required several approaches and techniques to transform such data into useful forms. For example, in chapter 4, the issue of the imbalanced classes was addressed with an oversampling technique, using SMOTE to oversample the dataset to solve the imbalance issue of the minority class attribute. Implementing such technique on the educational data generates synthetic instances (e.g. High-risk Students) without considering that these synthetic instances could be neighboring cases to the other classes. This can cause an overlapping issue with the other classes and can produce further noise. Furthermore, in Chapter 5, a resampling approach was applied with bootstrap aggregation to learn temporal Bayesian Networks from imbalanced performance classes. Although the classifier performance of predicting the academic performance has been improved with bootstrapping, the prediction result might be affected by the representation and the skewed sampling of the original

educational dataset.

- **Heterogeneous Data**

Educational data can be extracted from different sources (e.g. progression, admission, departments' databases, online repositories) which could bring an additional challenge to machine learning researchers. For example, the modeling approaches implemented in this thesis were adapted to analyse educational temporal data categories that were collected from Brunel University's admissions, Computer Science department databases, and the Blackboard Learn. However, the categories of this type of data (multi-source) were genuinely complex due to the variation of the different datasets included to obtain such data. Also, the domain values of the educational features (especially the admission features) were complex as the majority included categorical values, which were complicated to discretise and integrate into one temporal and consistent dataset.

- **Running Time**

Whilst the majority of the temporal models were learned and validated in a reasonable time and the step of modelling the biased features using the deep Multi-label Convolutional Networks (with Keras and Tensorflow on Google's Colaboratory) was completed successfully, there was a long running time. In particular, this was the case when learning the parameters of the Multi-label CNNs. This was expected as a shared Virtual Machine (VM) resource was being utilised. However, this was overcome for the later experiments using GPU acceleration to run the Python notebooks on Google Colab.

7.3 Future Work

7.3.1 Explainable AI

The development of the approach presented in this thesis with graphical models and deep learning contributes to the debate on ethical decision making with machine learning. With

this approach, unbiased and accurate classifiers, which consequentially improve performance prediction, were achieved. However, it is important to ensure that the data-driven models are applied responsibly, such that not just the learning approach is explainable, but also, the outputs are trustworthy. A new research direction would be useful to explore the explainable AI frameworks offered by Google to provide transparent temporal machine learning classifiers to users. Moreover, mimic learning could be constructed to learn explainable classifiers for transparent models. This technique uses deep learning approaches that mimic other models (e.g. BNs) without accessing the original data. Thus, the new capability will intend to provide user interpretable classifiers that are transparent in how and why the decision is made therefore the users can trust these decisions.

7.3.2 Temporal Profiles Assessment

It would be useful to assess the temporal clusters in more detail, in particular, with regards to identifying the correlations between the identified cognitive styles of students and their future work performance. To this end, graphical modelling approaches (e.g. Hidden Markov Models) and deep learning (e.g. Long Short Term Memory Models) are recommended for assessing the clusters as well as increasing the prediction of students' academic performance.

7.3.3 Synthetic Data Generation for Time-series Educational Data

As mentioned earlier, the temporal data extracted from the educational datasets were very limited due to the consideration of the time-series aspects in generating such data. It would be interesting to investigate synthetic data generation approaches with time-series data, especially where the educational datasets include confidential information, such as students' demographics and performance grades. This data could be generated with generative machine learning methods. For example, it would be interesting to investigate the Ordered Bayesian Networks and Deep Learning that exploited the robust classifiers in Chapter 5 and Chapter 6 for generating such data.

Appendix A

A.1 Feature Engineering of Educational Data

Feature engineering is a fundamental pre-processing mechanism for preparing the datasets before learning the conditional probabilities that project Bayesian and deep learner structures. As can be observed from Table 1, the original educational features include discrete as well as continuous domain values, which cannot be handled and inferred for the probabilistic reasoning of the networks. A discretisation technique was thus applied to the continuous features in the educational datasets. This was performed with an unsupervised learning method to exploit the hierarchical clustering to each continuous feature. When implementing the clustering method, the discretisation intervals were automated based on the bin count for each continuous feature. The clustering method used for discretising the data was exploited as follows:

Input: the number of student trajectories in the dataset (S) and the number of the desired bins (DB) for the educational feature

1. Let DB indicates the desired number of bins, then set (DB) to DB=S (where each student trajectory/record initialized by its cluster).
2. If DB=DB quit else DB=DB-1 by merging the two bins (those of the smallest separation among the mean values).
3. Repeat step 2 until coverage.

Table A.1 .1: The discrtization values used for pre-processing the educational datasets.

Feature	Domain Value	Equivalent Value
Gender	M	0
	F	1
Disability	A mental health condition, such as depression, schizophrenia or anxiety disorder	0
	A specific learning difficulty such as dyslexia, dyspraxia or AD(H)D	1
	A social/communication impairment such as Aspergers syndrome/other autistic spectrum disorder	2
	You have two or more impairments and/or disabling medical conditions	3
	You have a disability, impairment or medical condition not listed above	4
	Blind or a serious visual impairment uncorrected by glasses	5
	Deaf or a serious hearing impairment	6
	Wheelchair user/mobility difficulties	7
	A long standing illness or health condition such as cancer, HIV, diabetes, chronic heart disease, or epilepsy	8
	No known disability	9
Nationality	British	0
	European	1
	American	2
	Australian	3
	African	4
	Asian	5
	Middle Eastern	6
Ethnicity	Arab	0
	Asian	1
	White	2
	Black	3
	Other	4
Countery of Domicil	UK	0
	Eurpoe	1
	America	2
	Austraia	3
	Africa	4
	Asia	5
	Middle East	6
Countery of Birth	UK	0
	Eurpoe	1
	America	2
	Austraia	3
	Africa	4
	Asia	5
	Middle East	6
Fee Status	Home	0
	European	1
	Overseas	2
Socio Economic Class	Higher managerial and professional occupations	0
	Lower managerial and professional occupations	1
	Intermediate occupations	2
	Small employers and own account workers	3
	Lower supervisory and technical occupations	4
	Semi-routine occupations	5
	Routine occupations	6
	Never worked and long-time unemployed	7

	Not Classified	8
Previous Ed Estab LEA	South west england	0
	South east england	1
	Greater London	2
	Eastern england	3
	East midlands	4
	West midlands	5
	Welsh LEA	6
	North west england	7
	Yourkshir and Humerside	8
	North east england	9
	NA	10
Been In Care	Yes	0
	No	1
	NA	2
Mature	Yes	0
	No	1
Age on Entry	17-19	0
	20-24	1
	25 and over	2
Parents Been In HE	Yes	0
	No	1
	Prefer not to say	2
	NA	3
Route Code	Computer Science (Artificial Intelligence)	0
	Computer Science (Digital Media and Games)	1
	Computer Science (Network Computing)	2
	Computer Science	3
	Computer Science (Software Engineering)	4
	Business Computing (eBusiness)	5
	Business Computing	6
	Business Computing (Social Media)	7
Current Student?	Yes	0
	No	1
Module Grade	MA (Mitigating Circumstances Accepted)	0
	FT (Failed Terminated)	1
	F	2
	E	3
	D	4
	C	5
	B	6
	A	7
Temporal Profiles	Very Low Performance Profile	0
	Low Performance Profile	1
	Average performance Profile	2
	High Performance Late Profile	3
	High Performance Early Profile	4
Initial Cog. Style, Year 1 Cog. Style and Year 2 Cog, Style	C1	0
	C2	1
	C3	2
	C4	3
	C5	4

Appendix B

B.1 Feature Subspace Learning Additional Results

In the linear PCA, the most corresponding features with the PCs can be elicited by observing the ranked eigenvectors or loadings. Thus, the features with low values can be excluded due to the low dependencies between their loadings and PCs. Table 2 illustrates the loadings of the first 19 PCs (93% Variance) obtained by the varimax rotation. Varimax rotation kaiser1959computer is adapted here to maximise the variances via squaring the loadings between the features and the PCs abdi2003factor. After rotating the features with Varimax, the interpretation of the loadings will be simplified as each original feature will be associated with a small number of PCs, or in other words, each PC characterises a small number of features. Therefore, conducting varimax rotation makes the learned sub-space invariant as the PCs turn out to be uncorrelated with each other. The significance between the features and the PCs can be determined by observing the loading values of each feature. For example, if any given feature has a high loading value on a single PC and very low loading values (near-zero) on the other PCs, then the feature is highly correlated to this particular PC. Thus, identifying the unwanted dependencies between the features and the PCs can be determined by identifying the features with very low loadings in all the PCs. Then, the feature is statically not significant due to the low dependency between this particular feature and the PCs. In Table 2, it can be observed that the 'Gender', 'Country of Domicile', 'Fee Status', 'Mature', 'Age on Entry', 'Current Student?' and 'CS2555' features obtained the low corresponding loading values among the PCs.

Table B.1.1: Loadings of the first 19 PCs (93% Variance) obtained by the varimax rotation (Empty cells have zero loadings).

Feature	PC1	PC2	PC3	PC4	PC 5	PC6	PC7	PC 8	PC9	PC10	PC11	PC12	PC13	PC14	PC15	PC16	PC17	PC 18	PC19
Gender	-0.01	0.01		0.01	0.02	0.01	0.04	0.02	0.01		0.03	-0.03	0.03	-0.05	-0.04	0.04	-0.01	-0.02	
Disability	-0.01		0.01	1.00						-0.01			0.01	-0.01		0.01	-0.01	-0.01	-0.01
Nationality	-0.01	-0.01		0.01		-0.01			-0.01	0.99	-0.01	-0.01	-0.01	-0.02		0.02	-0.01	0.01	0.03
Ethnicity	-0.07		0.03	-0.01	0.02		0.03	0.01	-0.01		-0.04	-0.01	0.06	0.92	-0.06	0.06	-0.09		0.03
Country of Domicile	0.06	-0.01	0.03	-0.01	0.01	-0.02	-0.02	-0.03	-0.06	0.06	0.01		-0.08	0.01	0.01	0.02	-0.03	0.11	-0.04
Country of Birth					1.00	-0.01	-0.01				0.01		-0.01				0.02	0.02	0.03
Entry Year	-0.03				-0.01		0.06	0.03	0.02		-0.02	-0.01	0.09	-0.05	0.09	-0.03	-0.06	0.75	0.11
Fee Status			0.05	0.01			0.01	-0.01	-0.01	0.01				0.02		-0.01	-0.01	0.05	0.01
Socio Economic Class		1.00								0.01				-0.01			-0.01		0.04
Previous Ed Estab	0.01		0.99	-0.01		0.01	-0.01	0.01	0.01		-0.01		-0.01	-0.03		0.01	0.01		-0.01
Been In Care		-0.01			-0.01		-0.01	-0.01		-0.01			-0.01	-0.01		0.01	0.01		-0.03
Mature	-0.01	-0.03	-0.03	0.01	-0.01	-0.02		0.01	-0.02	-0.03			-0.02	-0.03	0.03	0.03	-0.02		-0.02
Age on Entry	0.02	0.02	0.05	-0.01		0.03			0.02	0.07	0.01	0.01	0.03	0.03	-0.06	-0.03	0.06	0.01	0.01
Parents Been In HE	0.01		0.01				0.01	0.01				0.98	0.02	0.01	0.01			-0.04	0.03
Route Code						-0.01	-0.01	-0.01	0.99	0.01					-0.03			0.02	-0.02
Current Student?	0.01			0.01	-0.01		-0.01	-0.03	-0.02	0.02	0.01	0.01	-0.02		-0.01	-0.02		-0.33	
CS1004	0.51			-0.01	0.02	-0.02	0.12	0.09	-0.01	0.01	-0.09	0.02	0.09	-0.13	0.05	0.04	-0.12	-0.16	-0.01
CS1005	0.18	-0.02			0.02	0.09	0.19	-0.01	0.04	-0.03	-0.03	-0.10	0.23		0.13	0.08	-0.47	-0.27	0.39
CS1803	0.04	0.01	0.02	0.01		0.01	-0.07	-0.06	-0.06	-0.05	-0.02	-0.07	-0.06	0.04	0.38		0.06	0.23	-0.04
CS1805	0.79	0.01			-0.02	-0.02	-0.11	-0.08		-0.01	0.05	-0.01	-0.09	0.07	-0.09	-0.01	0.13	0.16	
CS1809	-0.07		-0.01	-0.02	0.01		0.02	0.01	0.03		-0.02	0.07	0.06	-0.08	0.79	0.05	-0.07	-0.01	0.03
CS1810	0.16			0.04	-0.02			0.05	0.05	0.06	0.21	0.01	-0.15	0.33	0.38	-0.21	0.22	-0.09	0.02
CS1811	-0.04	-0.01			0.01		0.03	-0.01	-0.01	-0.01	0.96		0.02	-0.06	-0.04	0.04	-0.04		0.02
CS2001			-0.02		0.01	0.02	-0.95	0.02	0.01		-0.01	-0.01	0.06	-0.02	0.05	0.02	-0.07	-0.09	
CS2002			0.02	0.01	-0.02	-0.04	-0.11	-0.03	-0.02	0.02	0.04	0.06	-0.11	0.04	-0.06	-0.07	-0.78	0.17	-0.05
CS2004	-0.02		0.01		0.01		0.01				-0.02	-0.01	0.04	-0.03	-0.04	-0.96	-0.04	-0.03	0.05
CS2003	0.02	0.01	0.01		-0.01	-0.02	-0.05	-0.02	-0.01	0.02	0.05	0.01	0.90	0.04	-0.05	-0.02	0.11	0.11	-0.01
CS2005	0.05	0.03	-0.01	-0.01	0.04	0.04	0.10	0.03		0.03	-0.02	-0.01	0.14	0.02	0.09	-0.02	-0.14	-0.06	-0.89
CS2555	-0.01	0.04	0.01	-0.02	-0.01	-0.01		0.05	0.03	-0.02	0.09	-0.05	-0.13	0.05	0.01	-0.01	-0.16		-0.14
Temporal profiles	0.02		-0.02			0.01	-0.01	0.98		-0.01	0.01		-0.01		-0.04	0.01	0.01	0.05	0.03
Initial cog. Style	0.10	-0.02	-0.06	0.02	0.01	0.52	0.01	-0.08		-0.01		0.10	-0.09	0.03	-0.12	0.01	-0.02	0.24	-0.05
Year 1 cog. Style	-0.02		0.02			0.59	-0.02	0.05				-0.02	0.02	-0.01	0.03		0.04	-0.05	0.03
Year 2 cog. Style	-0.06	0.01	0.02	-0.01	-0.01	0.60				0.02	0.01	-0.05	0.03	-0.01	0.04	-0.01		-0.09	
No. of NON-Zero Loadings	26	16	20	19	22	21	24	26	21	23	24	22	30	28	26	27	29	26	28
Variance 2 (%)	**2.57**	**6.95**	**5.99**	**6.08**	**4.18**	**3.57**	**2.48**	**2.23**	**2.32**	**2.71**	**1.79**	**1.50**	**3.54**	**0.99**	**1.45**	**2.64**	**2.32**	**1.18**	**1.80**

B.2 Chapter 6 Additional Results

- **Feature Importance with BNs**

The influence strength identified between the features and the class node with the average and maximum influence strength distance measures is presented in Figures 1 and 2. These representations were learned from the conditional probability tables (CPT) of the child nodes in the BNs through performing some sort of distances between the probability distributions of the child nodes and the parent nodes. The influence strength in those figures is presented by the thickness and the colour of the arcs. The strong influences in the networks have been highlighted in 'blue'. The BNs are also representing the sensitivity analysis of the significant parameters on the class node. For instance, the red nodes include significant parameters for calculating the posterior probability distribution of the class 'Student Performance', whereas the grey ones do not contain any of the parameters used for learning the class. Hence, the important dependencies and their correlations with the class node can clearly be distinguished. Moreover, the results show that the average and maximum influence strength approaches have obtained quite similar representations in terms of characterising the important features on learning the posterior probability of the class.

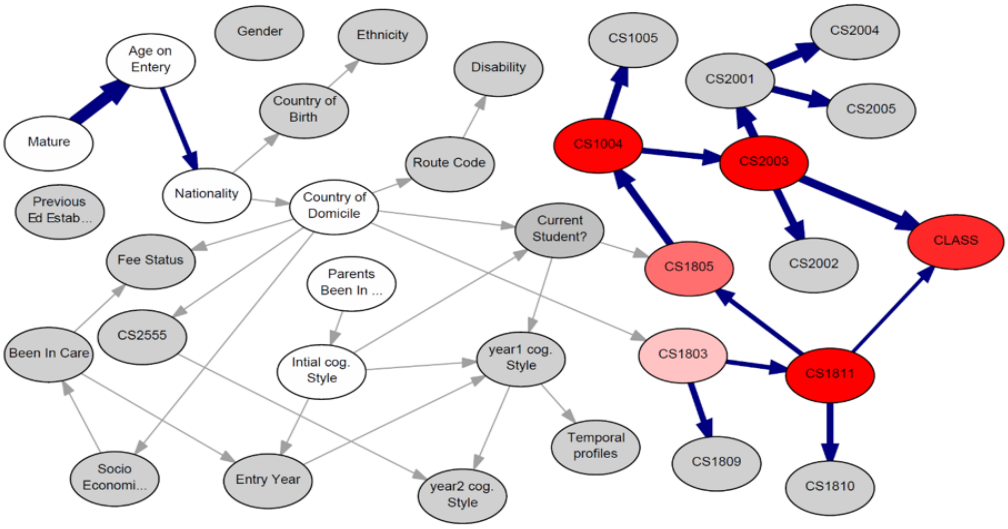


Figure B.2.1: Average influence strength which takes the average over distances between the parent and child nodes.

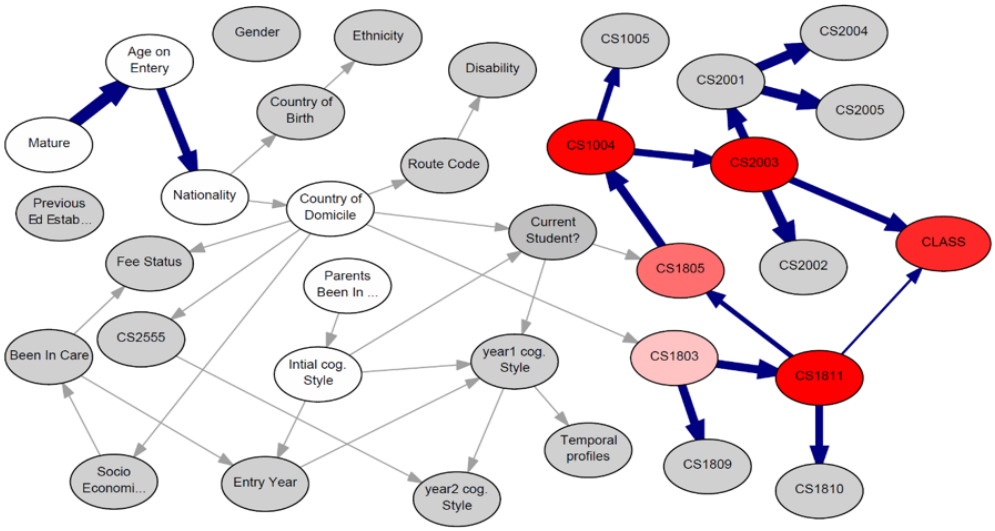


Figure B.2.2: Maximum influence strength which calculated based on the largest distance between distributions.

- **Deep Multi-Label ConvNets Results**

The results obtained with the CNNs for ethical decision making in machine learning classifiers are presented next. This includes the CNN network structure and the 10-fold cross validation accuracy results obtained for the three different experiments presented in chapter 6. These results were obtained with the Keras and TensorFlow Python libraries on Google Colab. Figure 3 provides the output results of experiment 1. In this experiment, the labels of the class variable were predicted, of which there were only three (Low Risk, Medium Risk, and High Risk), as can be seen with the structure of the network, keeping all the feature dependencies in the input layer. Whereas in experiment 2, the prediction was performed after removing the biased features from the original feature-space, as shown in Figure 4, to investigate whether removing the sensitive features (biased ones) from the feature-space will enhance the performance of the classifiers. Whilst the issue of using all feature dependencies (even the biased ones) was demonstrated in experiment 3, to investigate the possibility of performing unbiased and transparent classification (see Figure 5). In this experiment, the biased features identified from the Markov Blanket (MB) of the BN were assigned as labels/classifiers in the output layer alongside the class variables to ensure that they were conditionally independent of the class.

From these figures, it is clear that the structure used for performing the deep multi-label CNNs included the same feature extraction layers for better comparisons between the three approaches. The feature extraction layers included two 1D convolutional layers, namely Conv1d and Conv1d, with 1D 'MaxPooling', 'Dropout' of 0.4 and 'ReLU' activation, all being inserted after the second convolutional layer. The MaxPooling layer was determined to reduce the dimensions of the data, whereas the dropout layer was set to avoid overfitting. After this, the output layer was composed of one fully connected layer, with the 'ReLU' activation function and one last dense layer for projecting the labels. This fully connected layer was used to represent the vector of the features of the input. The last layer for the multi-labels CNN was computed with the 'Sigmoid' activation function to determine the probability of the three classifiers for experiments 1 and 2 as well as three labels for experiment 3. The last layer was set to the 'Sigmoid' function, as there was a multi-label classification problem.

Learning CNN for projecting the class labels (3 labels)

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d_355 (Conv1D)	(None, 41, 3)	12
conv1d_356 (Conv1D)	(None, 39, 3)	30
max_pooling1d_178 (MaxPoolin	(None, 19, 3)	0
dropout_355 (Dropout)	(None, 19, 3)	0
activation_178 (Activation)	(None, 19, 3)	0
flatten_178 (Flatten)	(None, 57)	0
dropout_356 (Dropout)	(None, 57)	0
dense_355 (Dense)	(None, 100)	5800
dense_356 (Dense)	(None, 3)	303

Total params: 6,145
Trainable params: 6,145
Non-trainable params: 0

Accuracy per fold for validating the CNN with 10-fold cross validation

[84.21052945287603, 90.93567457115441, 86.25731729624565, 82.16374516487122, 83.91812935210112, 87.4268981448391, 88.01169792811075, 81.28655040473268, 83.62572663708737, 79.53216077988607]

Figure B.2.3: Deep 1D multi-label CNN network structure and 10-fold CV accuracy results of experiment 1 exploited by the Keras and TensorFlow Python libraries.

Learning CNN for projecting the class labels when removing the sensitive features (3 labels)

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d_23 (Conv1D)	(None, 31, 3)	12
conv1d_24 (Conv1D)	(None, 29, 3)	30
max_pooling1d_12 (MaxPooling	(None, 14, 3)	0
dropout_23 (Dropout)	(None, 14, 3)	0
activation_12 (Activation)	(None, 14, 3)	0
flatten_12 (Flatten)	(None, 42)	0
dropout_24 (Dropout)	(None, 42)	0
dense_23 (Dense)	(None, 100)	4300
dense_24 (Dense)	(None, 3)	303

Total params: 4,645
Trainable params: 4,645
Non-trainable params: 0

Accuracy per fold for validating the CNN with 10-fold cross validation

[79.53216496266818, 80.70175564079955, 81.57894757756016, 79.2397675807016, 83.91813102521395, 82.4561455793548, 82.16374296891061, 83.04093825189692, 80.11695920375355, 77.19298318812722]

Figure B.2.4: Deep 1D multi-label CNN network structure and 10-fold CV accuracy results of experiment 2 exploited by the Keras and TensorFlow Python libraries.

Learning CNN for projecting the class labels as well as the sensitive features (13 labels)

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d_13 (Conv1D)	(None, 41, 13)	52
conv1d_14 (Conv1D)	(None, 39, 13)	520
max_pooling1d_7 (MaxPooling1	(None, 19, 13)	0
dropout_13 (Dropout)	(None, 19, 13)	0
activation_7 (Activation)	(None, 19, 13)	0
flatten_7 (Flatten)	(None, 247)	0
dropout_14 (Dropout)	(None, 247)	0
dense_13 (Dense)	(None, 100)	24800
dense_14 (Dense)	(None, 13)	1313

Total params: 26,685
Trainable params: 26,685
Non-trainable params: 0

Accuracy per fold for validating the CNN with 10-fold cross validation

[95.6815100552743, 95.74898951932003, 92.17274042597988, 94.53440841875577, 96.01889670940868, 93.72469500491493, 94.66936818340368, 93.7921678810789, 96.01889315404391, 95.61403383288467]

Figure B.2.5: Deep 1D multi-label CNN network structure and 10-fold CV accuracy results of experiment 3 exploited by the Keras and TensorFlow Python libraries.

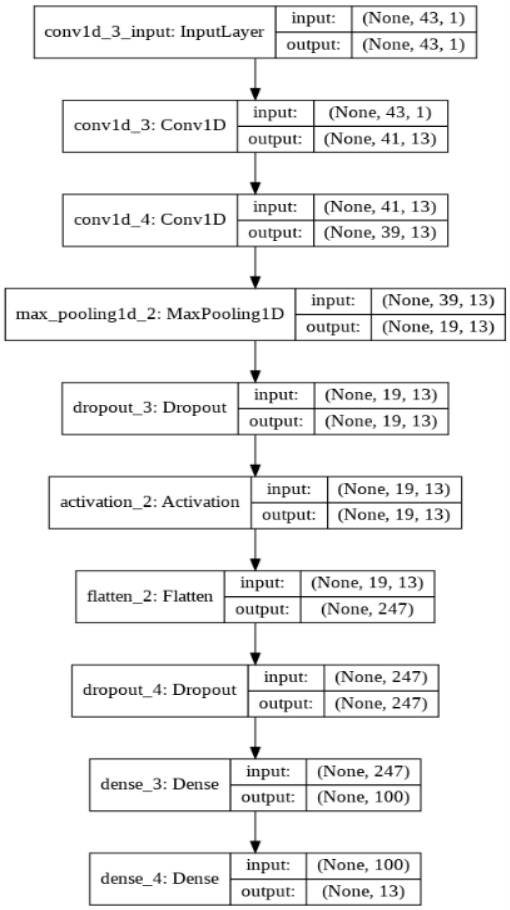


Figure B.2.6: Deep 1D multi-label CNN network structure of experiment 3 exploited by the Keras and TensorFlow Python libraries on Google Colaboratory (Google CoLab).

Bibliography

Abu Tair, M. M. and El-Halees, A. M. (2012), 'Mining educational data to improve students' performance: a case study', *Mining educational data to improve students' performance: a case study* **2**(2).

Adèr, H. J. (2008), *Advising on research methods: A consultant's companion*, Johannes van Kessel Publishing.

Agurto, C., Murray, V., Barriga, E., Murillo, S., Pattichis, M., Davis, H., Russell, S., Abràmoff, M. and Soliz, P. (2010), 'Multiscale am-fm methods for diabetic retinopathy lesion detection', *IEEE transactions on medical imaging* **29**(2), 502–512.

Aher, S. B. and Lobo, L. (2011), Data mining in educational system using weka, *in* 'International Conference on Emerging Technology Trends (ICETT)', Vol. 3, pp. 20–25.

Al-Luhaybi, M., Yousefi, L., Swift, S., Counsell, S. and Tucker, A. (2019), Predicting academic performance: A bootstrapping approach for learning dynamic bayesian networks, *in* 'International Conference on Artificial Intelligence in Education', Springer, pp. 26–36.

Al-Radaideh, Q. A., Al-Shawakfa, E. M. and Al-Najjar, M. I. (2006), Mining student data using decision trees, *in* 'International Arab Conference on Information Technology (ACIT'2006), Yarmouk University, Jordan'.

Alfiani, A. P., Wulandari, F. A. et al. (2015), 'Mapping student's performance based on data mining approach (a case study)', *Agriculture and Agricultural Science Procedia* **3**, 173–177.

Araque, F., Roldán, C. and Salguero, A. (2009), 'Factors influencing university drop out rates', *Computers & Education* **53**(3), 563–574.

Arsad, P. M., Buniyamin, N. et al. (2013), A neural network students' performance prediction model (nnsppm), *in* '2013 IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)', IEEE, pp. 1–5.

Athreya, K. (1987), 'Bootstrap of the mean in the infinite variance case', *The Annals of Statistics* pp. 724–731.

Baker, R. et al. (2010), 'Data mining for education', *International encyclopedia of education* **7**(3), 112–118.

Beal, C. and Cohen, P. (2005), Comparing apples and oranges: Computational methods for evaluating student and group learning histories in intelligent tutoring systems, *in* 'Proceedings of the 12th International Conference on Artificial Intelligence in Education', Citeseer, pp. 555–562.

Bekele, R. and Menzel, W. (2005), 'A bayesian approach to predict performance of a student (bapps): A case with ethiopian students', *algorithms* **22**(23), 24.

Bhardwaj, B. K. and Pal, S. (2012), 'Data mining: A prediction for performance improvement using classification', *arXiv preprint arXiv:1201.3418* .

bin Mat, U., Buniyamin, N., Arsad, P. M. and Kassim, R. (2013), An overview of using academic analytics to predict and improve students' achievement: A proposed proactive intelligent intervention, *in* 'Engineering Education (ICEED), 2013 IEEE 5th Conference on', IEEE, pp. 126–130.

Bishop, C. M. et al. (1995), *Neural networks for pattern recognition*, Oxford university press.

Boud, D. and Brew, A. (1995), 'Developing a typology for learner self assessment practices', *Research and development in Higher Education* **18**(1), 130–135.

Brown, L. E., Tsamardinos, I. and Aliferis, C. F. (2004), A novel algorithm for scalable and accurate bayesian network learning., *in* 'Medinfo', pp. 711–715.

Carmona, C., Castillo, G. and Millán, E. (2008), Designing a dynamic bayesian network for modeling students' learning styles, *in* '2008 eighth IEEE international conference on advanced learning technologies', IEEE, pp. 346–350.

Castillo, E., Gutiérrez, J. M. and Hadi, A. S. (1997), 'Sensitivity analysis in discrete bayesian networks', *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* **27**(4), 412–423.

Castro, F., Vellido, A., Nebot, A. and Mugica, F. (2007), Applying data mining techniques to e-learning problems, *in* 'Evolution of teaching and learning paradigms in intelligent environment', Springer, pp. 183–221.

Challis, D. (2005), 'Committing to quality learning through adaptive online assessment', *Assessment & Evaluation in Higher Education* **30**(5), 519–527.

Chan, A. B. and Vasconcelos, N. (2007), Classifying video with kernel dynamic textures, *in* '2007 IEEE Conference on Computer Vision and Pattern Recognition', IEEE, pp. 1–6.

Charrad, M., Ghazzali, N., Boiteau, V., Niknafs, A. and Charrad, M. M. (2014), 'Package 'nbclust'', *Journal of Statistical Software* **61**, 1–36.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002), 'Smote: synthetic minority over-sampling technique', *Journal of artificial intelligence research* **16**, 321–357.

Chen, C.-M., Chen, M.-C. and Li, Y.-L. (2007), Mining key formative assessment rules based on learner profiles for web-based learning systems, *in* 'null', IEEE, pp. 584–588.

Cobo, G., García-Solórzano, D., Santamaria, E., Morán, J. A., Melenchón, J. and Monzo, C. (2011), Modeling students' activity in online discussion forums: A strategy based on time series and agglomerative hierarchical clustering., *in* 'EDM', pp. 253–258.

Conati, C., Gertner, A. and Vanlehn, K. (2002), 'Using bayesian networks to manage uncertainty in student modeling', *User modeling and user-adapted interaction* **12**(4), 371–417.

Cortez, P. and Silva, A. M. G. (2008), 'Using data mining to predict secondary school student performance'.

Druzdzel, M. J. (1999), Genie: A development environment for graphical decision-analytic models, *in* 'Proceedings of the AMIA Symposium', American Medical Informatics Association, p. 1206.

El-Halees, A. M. (2009), 'Mining students data to analyze e-learning behavior: A case study', *Mining students data to analyze e-Learning behavior: A Case Study* **29**.

Fawcett, T. (2006), 'An introduction to roc analysis', *Pattern recognition letters* **27**(8), 861–874.

Feng, M., Beck, J. E. and Heffernan, N. T. (2009), 'Using learning decomposition and bootstrapping with randomization to compare the impact of different educational interventions on learning.', *International Working Group on Educational Data Mining* .

Friedman, N., Goldszmidt, M. and Wyner, A. (2013), 'Data analysis with bayesian networks: A bootstrap approach', *arXiv preprint arXiv:1301.6695* .

Friedman, N., Linial, M., Nachman, I. and Pe'er, D. (2000), 'Using bayesian networks to analyze expression data', *Journal of computational biology* **7**(3-4), 601–620.

Fukunaga, K. (2013), *Introduction to statistical pattern recognition*, Elsevier.

García, P., Amandi, A., Schiaffino, S. and Campo, M. (2007), 'Evaluating bayesian networks' precision for detecting students' learning styles', *Computers & Education* **49**(3), 794–808.

Geng, Z. and Zhu, Q. (2005), 'Multiscale nonlinear principal component analysis (nlpca) and its application for chemical process monitoring', *Industrial & engineering chemistry research* **44**(10), 3585–3593.

Godbole, S. and Sarawagi, S. (2004), Discriminative methods for multi-labeled classification, *in* 'Pacific-Asia conference on knowledge discovery and data mining', Springer, pp. 22–30.

Gray, G., McGuinness, C. and Owende, P. (2014), An application of classification models to predict learner progression in tertiary education, *in* '2014 IEEE International Advance Computing Conference (IACC)', IEEE, pp. 549–554.

Gulli, A. and Pal, S. (2017), *Deep Learning with Keras*, Packt Publishing Ltd.

Gunning, D. (2017), 'Explainable artificial intelligence (xai)', *Defense Advanced Research Projects Agency (DARPA), nd Web* **2**.

Hämäläinen, W. and Vinni, M. (2006), Comparison of machine learning methods for intelligent tutoring systems, *in* 'International Conference on Intelligent Tutoring Systems', Springer, pp. 525–534.

Hämäläinen, W. and Vinni, M. (2011), 'Classifiers for educational data mining', *Handbook of Educational Data Mining, Chapman & Hall/CRC Data Mining and Knowledge Discovery Series* pp. 57–71.

Hand, D. J. (2007), 'Principles of data mining', *Drug safety* **30**(7), 621–622.

He, H. and Garcia, E. A. (2009), 'Learning from imbalanced data', *IEEE Transactions on knowledge and data engineering* **21**(9), 1263–1284.

Heckerman, D., Geiger, D. and Chickering, D. M. (1995), 'Learning bayesian networks: The combination of knowledge and statistical data', *Machine learning* **20**(3), 197–243.

Herman, A. (2007), 'Nonlinear principal component analysis of the tidal dynamics in a shallow sea', *Geophysical Research Letters* **34**(2).

HESA (2019), 'The higher education statistics agency (hesa) website, student enrolments by level of study'.
URL: *https://www.hesa.ac.uk/data-and-analysis/sfr247/figure-3*

Hoffmann, H. (2007), 'Kernel pca for novelty detection', *Pattern recognition* **40**(3), 863–874.

Hogo, M. A. (2010), 'Evaluation of e-learning systems based on fuzzy clustering models and statistical tools', *Expert systems with applications* **37**(10), 6891–6903.

Holzinger, A., Biemann, C., Pattichis, C. S. and Kell, D. B. (2017), 'What do we need to build explainable ai systems for the medical domain?', *arXiv preprint arXiv:1712.09923* .

Hssina, B., Merbouha, A., Ezzikouri, H. and Erritali, M. (2014), 'A comparative study of decision tree id3 and c4. 5', *International Journal of Advanced Computer Science and Applications* **4**(2), 13–19.

Hu, Y.-J. (1998), Constructive induction: covering attribute spectrum, *in* 'Feature Extraction, Construction and Selection', Springer, pp. 257–272.

Ibabe, I. and Jauregizar, J. (2010), 'Online self-assessment with feedback and metacognitive knowledge', *Higher Education* **59**(2), 243–258.

Ibrahim, Z. and Rusli, D. (2007), Predicting students' academic performance: comparing artificial neural network, decision tree and linear regression, *in* '21st Annual SAS Malaysia Forum, 5th September'.

Is C5.0 Better Than C4.5 (n.d.), `https://rulequest.com/see5-comparison.html`. Accessed: 2020-02-21.

Jain, A. K. (2010), 'Data clustering: 50 years beyond k-means', *Pattern recognition letters* **31**(8), 651–666.

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013), *An introduction to statistical learning*, Vol. 112, Springer.

Jason, B. (2018), *Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python.*, Machine Learning Mastery.

Jia, S., Lansdall-Welfare, T. and Cristianini, N. (2018), Right for the right reason: Training agnostic networks, *in* 'International Symposium on Intelligent Data Analysis', Springer, pp. 164–174.

Kabakchieva, D. (2013), 'Predicting student performance by using data mining methods for classification', *Cybernetics and information technologies* **13**(1), 61–72.

Kampouridis, M. and Otero, F. E. (2013), Using attribute construction to improve the predictability of a gp financial forecasting algorithm, *in* 'Technologies and Applications of Artificial Intelligence (TAAI), 2013 Conference on', IEEE, pp. 55–60.

Kassambara, A. (2017), *Practical guide to cluster analysis in R: Unsupervised machine learning*, Vol. 1, STHDA.

Kaur, G. and Singh, W. (2016), 'Prediction of student performance using weka tool', *An International Journal of Engineering Sciences* **17**, 8–16.

Kaur, P., Singh, M. and Josan, G. S. (2015), 'Classification and prediction based data mining algorithms to predict slow learners in education sector', *Procedia Computer Science* **57**, 500–508.

Keogh, E. J. and Pazzani, M. J. (2000), A simple dimensionality reduction technique for fast similarity search in large time series databases, *in* 'Pacific-Asia conference on knowledge discovery and data mining', Springer, pp. 122–133.

Keogh, E. J. and Pazzani, M. J. (2001), Derivative dynamic time warping, *in* 'Proceedings of the 2001 SIAM international conference on data mining', SIAM, pp. 1–11.

Keogh, E. and Ratanamahatana, C. A. (2005), 'Exact indexing of dynamic time warping', *Knowledge and information systems* **7**(3), 358–386.

Kim, K. I., Jung, K. and Kim, H. J. (2002), 'Face recognition using kernel principal component analysis', *IEEE signal processing letters* **9**(2), 40–42.

Kim, K., Park, S. and Kim, H. (2001), 'Kernel principal component analysis for texture classification', *IEEE Signal Processing Letters* **8**(2), 39–41.

Kingma, D. P. and Ba, J. (2014), 'Adam: A method for stochastic optimization', *arXiv preprint arXiv:1412.6980* .

Kjærulff, U. and van der Gaag, L. C. (2013), 'Making sensitivity analysis computationally efficient', *arXiv preprint arXiv:1301.3868* .

Koller, D. and Friedman, N. (2009), *Probabilistic graphical models: principles and techniques*, MIT press.

Koller, D. and Sahami, M. (1996), Toward optimal feature selection, Technical report, Stanford InfoLab.

Kramer, M. A. (1991), 'Nonlinear principal component analysis using autoassociative neural networks', *AIChE journal* **37**(2), 233–243.

Labatut, V. and Cherifi, H. (2012), 'Accuracy measures for the comparison of classifiers', *arXiv preprint arXiv:1207.3790* .

Lagani, V., Athineou, G., Farcomeni, A., Tsagris, M. and Tsamardinos, I. (2016), 'Feature selection with the r package mxm: Discovering statistically-equivalent feature subsets', *arXiv preprint arXiv:1611.03227* .

Lopez, M. I., Luna, J. M., Romero, C. and Ventura, S. (2012), 'Classification via clustering for predicting final marks based on student participation in forums.', *International Educational Data Mining Society* .

Lykourentzou, I., Giannoukos, I., Mpardis, G., Nikolopoulos, V. and Loumos, V. (2009), 'Early and dynamic student achievement prediction in e-learning courses using neural networks', *Journal of the American Society for Information Science and Technology* **60**(2), 372–380.

MacQueen, J. et al. (1967), Some methods for classification and analysis of multivariate observations, *in* 'Proceedings of the fifth Berkeley symposium on mathematical statistics and probability', Vol. 1, Oakland, CA, USA, pp. 281–297.

Mansur, M., Sap, M. and Noor, M. (2005), Outlier detection technique in data mining: a research perspective, *in* 'Postgraduate Annual Research Seminar'.

Martis, R. J., Acharya, U. R. and Min, L. C. (2013), 'Ecg beat classification using pca, lda, ica and discrete wavelet transform', *Biomedical Signal Processing and Control* **8**(5), 437–448.

Mayilvaganan, M. and Kalpanadevi, D. (2014), Comparison of classification techniques for predicting the performance of students academic environment, *in* '2014 International Conference on Communication and Network Technologies', IEEE, pp. 113–118.

McLaren, B. M., Koedinger, K. R., Schneider, M., Harrer, A. and Bollen, L. (2004), 'Bootstrapping novice data: Semi-automated tutor authoring using student log files'.

Megalooikonomou, V., Li, G. and Wang, Q. (2004), A dimensionality reduction technique for efficient similarity analysis of time series databases, *in* 'Proceedings of the thirteenth ACM international conference on Information and knowledge management', ACM, pp. 160–161.

Merceron, A. and Yacef, K. (2005), Educational data mining: a case study., *in* 'AIED', pp. 467–474.

Michalski, K. and Michalski, R. (2004), Educational data mining and reporting: Analyzing student data in order to improve educational processes, *in* 'EdMedia: World Conference on Educational Media and Technology', Association for the Advancement of Computing in Education (AACE), pp. 1088–1094.

Mihajlovic, V. and Petkovic, M. (2001), 'Dynamic bayesian networks: A state of the art', *University of Twente Document Repository* .

Minaei-Bidgoli, B., Kashy, D. A., Kortemeyer, G. and Punch, W. F. (2003), Predicting student performance: an application of data mining methods with an educational web-based system, *in* '33rd Annual Frontiers in Education, 2003. FIE 2003.', Vol. 1, IEEE, pp. T2A–13.

Młynarska, E., Greene, D. and Cunningham, P. (2016), Time series clustering of moodle activity data, *in* '24th Irish Conference on Artificial Intelligence and Cognitive Science (AICS'16), University College Dublin, Dublin, Ireland, 20-21 September 2016'.

Moniz, N., Branco, P. and Torgo, L. (2016), Resampling strategies for imbalanced time series, *in* '2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)', IEEE, pp. 282–291.

Moon, T. K. (1996), 'The expectation-maximization algorithm', *IEEE Signal processing magazine* **13**(6), 47–60.

Murphy, K. P. and Russell, S. (2002), 'Dynamic bayesian networks: representation, inference and learning'.

Murphy, K. et al. (2001), 'The bayes net toolbox for matlab', *Computing science and statistics* **33**(2), 1024–1034.

Nielsen, T. D. and Jensen, F. V. (2009), *Bayesian networks and decision graphs*, Springer Science & Business Media.

Nonlinear PCA by Matthias Scholz (n.d.), `http://www.nlpca.org/`. Accessed: 2020-02-20.

Oladokun, V., Adebanjo, A. and Charles-Owaba, O. (2008), 'Predicting students academic performance using artificial neural network: A case study of an engineering course'.

Osmanbegovic, E. and Suljic, M. (2012), 'Data mining approach for predicting student performance', *Economic Review: Journal of Economics and Business* **10**(1), 3–12.

Pearl, J. (2011), 'Bayesian networks'.

Pearl, J. (2014), *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Elsevier.

Pearson, K. (1901), 'Liii. on lines and planes of closest fit to systems of points in space', *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **2**(11), 559–572.

Perera, D., Kay, J., Koprinska, I., Yacef, K. and Zaïane, O. R. (2008), 'Clustering and sequential pattern mining of online collaborative learning data', *IEEE Transactions on Knowledge and Data Engineering* **21**(6), 759–772.

Perera, D., Kay, J., Koprinska, I., Yacef, K. and Zaiane, O. R. (2009), 'Clustering and sequential pattern mining of online collaborative learning data', *IEEE Transactions on Knowledge and Data Engineering* **21**(6), 759–772.

Peterson, P. L., Baker, E. and McGaw, B. (2010), *International encyclopedia of education*, Elsevier Ltd.

Pfannkuch, M., Forbes, S., Harraway, J., Budgett, S. and Wild, C. (2013), Bootstrapping students' understanding of statistical inference, Technical report, Summary research report for the Teaching and Learning Research Initiative ....

Poh, N. and Smythe, I. (2014), To what extend can we predict students' performance? a case study in colleges in south africa, *in* '2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)', IEEE, pp. 416–421.

Quadri, M. M. and Kalyankar, N. (2010), 'Drop out feature of student data for academic performance using decision tree techniques', *Global Journal of Computer Science and Technology* .

Riehmann, P., Hanfler, M. and Froehlich, B. (2005), Interactive sankey diagrams, *in* 'IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.', IEEE, pp. 233–240.

Romero, C., Espejo, P. G., Zafra, A., Romero, J. R. and Ventura, S. (2013), 'Web usage mining for predicting final marks of students that use moodle courses', *Computer Applications in Engineering Education* **21**(1), 135–146.

Romero, C., Lopez, M.-I., Luna, J.-M. and Ventura, S. (2013), 'Predicting students' final performance from participation in on-line discussion forums', *Computers & Education* **68**, 458–472.

Romero, C. and Ventura, S. (2007), 'Educational data mining: A survey from 1995 to 2005', *Expert systems with applications* **33**(1), 135–146.

Romero, C. and Ventura, S. (2010), 'Educational data mining: a review of the state of the art', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **40**(6), 601–618.

Rosipal, R., Girolami, M., Trejo, L. J. and Cichocki, A. (2001), 'Kernel pca for feature extraction and de-noising in nonlinear regression', *Neural Computing & Applications* **10**(3), 231–243.

Scholz, M. (2012), 'Validation of nonlinear pca', *Neural processing letters* **36**(1), 21–30.

Scholz, M. and Fraunholz, M. J. (2008), 'A computational model of gene expression reveals early transcriptional events at the subtelomeric regions of the malaria parasite, plasmodium falciparum', *Genome biology* **9**(5), R88.

Scholz, M., Fraunholz, M. and Selbig, J. (2008), Nonlinear principal component analysis: neural network models and applications, *in* 'Principal manifolds for data visualization and dimension reduction', Springer, pp. 44–67.

Seffrin, H. M., Rubi, G. L. and Jaques, P. A. (2014), A dynamic bayesian network for inference of learners' algebraic knowledge, *in* 'Proceedings of the 29th Annual ACM Symposium on Applied Computing', pp. 235–240.

Sembiring, S., Zarlis, M., Hartama, D., Ramliana, S. and Wani, E. (2011), Prediction of student academic performance by an application of data mining techniques, *in* 'International Conference on Management and Artificial Intelligence IPEDR', Vol. 6, pp. 110–114.

Shen, S. and Chi, M. (2017), Clustering student sequential trajectories using dynamic time warping, *in* 'Proceedings of the 10th International Conference on Educational Data Mining', pp. 266–271.

Spearman, C. (1904), '" general intelligence," objectively determined and measured', *The American Journal of Psychology* **15**(2), 201–292.

Spirtes, P., Glymour, C. N., Scheines, R. and Heckerman, D. (2000), *Causation, prediction, and search*, MIT press.

Street, W. N. and Kim, Y. (2001), A streaming ensemble algorithm (sea) for large-scale classification, *in* 'Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 377–382.

Subasi, A. and Gursoy, M. I. (2010), 'Eeg signal classification using pca, ica, lda and support vector machines', *Expert systems with applications* **37**(12), 8659–8666.

Talavera, L. and Gaudioso, E. (2004), Mining student data to characterize similar behavior groups in unstructured collaboration spaces, *in* 'Workshop on artificial intelligence in CSCL. 16th European conference on artificial intelligence', pp. 17–23.

Tan, A. C. and Gilbert, D. (2003), 'Ensemble machine learning on gene expression data for cancer classification'.

Tsoumakas, G., Katakis, I. and Vlahavas, I. (2009), Mining multi-label data, *in* 'Data mining and knowledge discovery handbook', Springer, pp. 667–685.

Tsoumakas, G. and Vlahavas, I. (2007), Random k-labelsets: An ensemble method for multilabel classification, *in* 'European conference on machine learning', Springer, pp. 406–417.

Van Der Maaten, L., Postma, E. and Van den Herik, J. (2009), 'Dimensionality reduction: a comparative', *J Mach Learn Res* **10**(66-71), 13.

Verleysen, M. and François, D. (2005), The curse of dimensionality in data mining and time series prediction, *in* 'International Work-Conference on Artificial Neural Networks', Springer, pp. 758–770.

Viera, A. J., Garrett, J. M. et al. (2005), 'Understanding interobserver agreement: the kappa statistic', *Fam med* **37**(5), 360–363.

Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S. et al. (2001), Constrained k-means clustering with background knowledge, *in* 'Icml', Vol. 1, pp. 577–584.

Wang, Q. and Megalooikonomou, V. (2008), 'A dimensionality reduction technique for efficient time series similarity analysis', *Information systems* **33**(1), 115–132.

Witten, I. H. and Frank, E. (2002), 'Data mining: practical machine learning tools and techniques with java implementations', *Acm Sigmod Record* **31**(1), 76–77.

Xia, R., Zong, C. and Li, S. (2011), 'Ensemble of feature sets and classification algorithms for sentiment classification', *Information Sciences* **181**(6), 1138–1152.

Yadav, S. K. and Pal, S. (2012), 'Data mining: A prediction for performance improvement of engineering students using classification', *arXiv preprint arXiv:1203.3832* .

Yang, K. and Shahabi, C. (2004), A pca-based similarity measure for multivariate time series, *in* 'Proceedings of the 2nd ACM international workshop on Multimedia databases', ACM, pp. 65–74.

Yang, K. and Shahabi, C. (2005), A pca-based kernel for kernel pca on multivariate time series, *in* 'Proceedings of ICDM 2005 workshop on temporal data mining: algorithms, theory and applications held in conjunction with the fifth IEEE international conference on data mining (ICDM'05)', pp. 149–156.

Yaramakala, S. and Margaritis, D. (2005), Speculative markov blanket discovery for optimal feature selection, *in* 'Fifth IEEE International Conference on Data Mining (ICDM'05)', IEEE, pp. 4–pp.

Zakrzewska, D. (2008), Cluster analysis for users' modeling in intelligent e-learning systems, *in* 'International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems', Springer, pp. 209–214.

Zhang, H. (2004), 'The optimality of naive bayes', *AA* **1**(2), 3.