Outlier Detection as a Tool for Reinforcing Data Analysis and Prediction in Education

Daria Novoseltseva

Université Fédérale
Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 3 - Paul Sabatier

Présentée et soutenue par

Daria NOVOSELTSEVA

Le 2 février 2022

La Détection d'Anomalies Comme Outil de Renforcement d'Analyse des Données et de Prédiction dans l'Éducation

---

Ecole doctorale : **EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse**

Spécialité : **Informatique et Télécommunications**

Unité de recherche :
IRIT : Institut de Recherche en Informatique de Toulouse

Thèse dirigée par
Nadine JESSEL BAPTISTE et Florence SèDES

Jury

M. Sébastien GEORGE, Rapporteur
Mme Anne BOYER, Rapporteure
M. Colin DE LA HIGUERA, Examinateur
Mme Nadine JESSEL, Directrice de thèse
Mme Florence SEDéS, Co-directrice de thèse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *02/02/2022* **par :**

Daria NOVOSELTSEVA

Outlier Detection as a Tool for Reinforcing Data Analysis and Prediction in Education

---

JURY

Anne BOYER	Université de Lorraine	Rapporteure
Sébastien GEORGE	Le Mans Université	Rapporteur
Colin de la Higuera	Université de Nantes	Examinateur
Nadine Jessel	Université Toulouse-Jean-Jaurès	Directrice de thèse
Florence Sedés	Université Paul Sabatier	Co-directrice de thèse
Agathe Merceron	Berliner Hochschule für Technik	Invitée
Petra Sauer	Berliner Hochschule für Technik	Invitée
Catherine PONS Lelardeux	INU Champollion	Invitée

École doctorale et spécialité :
MITT : Image, Information, Hypermédia
Unité de Recherche :
Institut de Recherche en Informatique de Toulouse (UMR 5505)
Directeur(s) de Thèse :
Nadine Jessel et *Florence Sedés*
Rapporteurs :
et

Abstract

Educational institutions seek to design effective mechanisms that improve academic results, enhance the learning process, and avoid dropout. The performance analysis and performance prediction of students in their studies may show drawbacks in the educational formations and detect students with learning problems. This induces the task of developing techniques and data-based models which aim to enhance teaching and learning. Classical models usually ignore the students-outliers with uncommon and inconsistent characteristics although they may show significant information to domain experts and affect the prediction models. The outliers in education are barely explored and their impact on the prediction models has not been studied yet in the literature. Thus, the thesis aims to investigate the outliers in educational data and extend the existing knowledge about them.

The thesis presents three case studies of outlier detection for different educational contexts and ways of data representation (numerical dataset for the German University, numerical dataset for the Russian University, sequential dataset for French nurse schools). For each case, the data preprocessing approach is proposed regarding the dataset peculiarities. The prepared data has been used to detect outliers in conditions of unknown ground truth. The characteristics of detected outliers have been explored and analysed, which allowed extending the comprehension of students' behaviour in a learning process.

One of the main tasks in the educational domain is to develop essential tools which will help to improve academic results and reduce attrition. Thus, plenty of studies aim to build models of performance prediction which can detect students with learning problems that need special help. The second goal of the thesis is to study the impact of outliers on prediction models. The two most common prediction tasks in the educational field have been considered: (i) dropout prediction, (ii) the final score prediction. The prediction models have been compared in terms of different prediction algorithms and the presence of outliers in the training data.

This thesis opens new avenues to investigate the students' performance in educational environments. The understanding of outliers and the reasons for their appearance can help domain experts to extract valuable information from

the data. Outlier detection might be a part of the pipeline in the early warning systems of detecting students with a high risk of dropouts. Furthermore, the behavioral tendencies of outliers can serve as a basis for providing recommendations for students in their studies or making decisions about improving the educational process.

Keywords: Outlier detection, Outlier, Anomalies, Educational data mining, Learning analytics.

Résumé

Les établissements d'enseignement cherchent à concevoir des mécanismes efficaces pour améliorer les résultats scolaires, renforcer le processus d'apprentissage et éviter l'abandon scolaire. L'analyse et la prédiction des performances des étudiants au cours de leurs études peuvent mettre en évidence certaines lacunes d'une formation et détecter les étudiants ayant des problèmes d'apprentissage. Il s'agit donc de développer des techniques et des modèles basés sur des données qui visent à améliorer l'enseignement et l'apprentissage. Les modèles classiques ignorent généralement les étudiants présentant des comportements et incohérences inhabituels, bien qu'ils puissent fournir des informations importantes aux experts du domaine et améliorer les modèles de prédiction. Les profils atypiques dans l'éducation sont à peine explorés et leur impact sur les modèles de prédiction n'a pas encore été étudié dans la littérature. Cette thèse vise donc à étudier les valeurs anormales dans les données éducatives et à étendre les connaissances existantes à leur sujet.

La thèse présente trois études de cas de détection de données anormales pour différents contextes éducatifs et modes de représentation des données (jeu de données numériques pour une université allemande, jeu de données numériques pour une université russe, jeu de données séquentiel pour les écoles d'infirmières françaises). Pour chaque cas, l'approche de prétraitement des données est proposée en tenant compte des particularités du jeu de données. Les données préparées ont été utilisées pour détecter les valeurs anormales dans des conditions de vérité terrain inconnue. Les caractéristiques des valeurs anormales détectées ont été explorées et analysées, ce qui a permis d'étendre les connaissances sur le comportement des étudiants dans un processus d'apprentissage.

L'une des principales tâches dans le domaine de l'éducation est de développer des mécanismes essentiels qui permettront d'améliorer les résultats scolaires et de réduire l'abandon scolaire. Ainsi, il est nécessaire de construire des modèles de prédiction de performance qui sont capables de détecter les étudiants ayant des problèmes d'apprentissage, qui ont besoin d'une aide spéciale. Le deuxième objectif de la thèse est d'étudier l'impact des valeurs anormales sur les modèles de prédiction. Nous avons considéré deux des tâches de prédiction les plus courantes dans le domaine de l'éducation: (i) la prédiction de l'abandon sco-

laire, (ii) la prédiction du score final. Les modèles de prédiction ont été comparés en fonction de différents algorithmes de prédiction et de la présence de valeurs anormales dans les données d'entraînement.

Cette thèse ouvre de nouvelles voies pour étudier les performances des élèves dans les environnements éducatifs. La compréhension des valeurs anormales et des raisons de leur apparition peut aider les experts du domaine à extraire des informations précieuses des données. La détection des valeurs aberrantes pourrait faire partie du pipeline des systèmes d'alerte précoce pour détecter les élèves à haut risque d'abandon. De plus, les tendances comportementales des valeurs aberrantes peuvent servir de base pour fournir des recommandations aux étudiants dans leurs études ou prendre des décisions concernant l'amélioration du processus éducatif.

Mots-clés: Détection des valeurs anormales, valeurs anormales, anomalies, exploration de données éducatives, analyse de l'apprentissage.

Contents

Chapter 1

Introduction

The scariest moment is always just before you start.
—Stephen King

In the 1980s the world faced the digital revolution: computers and programs became actively pervade to all fields of our life. They completely changed the ways of storage and implementation of data. The evolution of programs simplified a lot of concerns, e.g. they can make complex calculations on our laptops or can accumulate the level of our physical activity in our smartphones. Later, humanity confronted the following issues: firstly, the gathered data became too big to analyse by simple methods; secondly, new tasks related to specific topics arose. As a result, in the 2020s, interdisciplinary fields related to working with data in various domains are demanded, and education is not an exception. The arising of massive online courses, using the internet for education, the designing new educational environments, all this induced the accumulation of educational data. The analysis of this data might point out drawbacks in educational systems. This generated the tasks of extracting knowledge and insights from structured and unstructured data and developing various methods, processes, algorithms in education. There are two main communities, that aim to work out these tasks Educational Data Mining (EDM) [1] and Learning Analytics (LA) [2]. Despite the fundamental difference, both EDM and LA share the same interests –to improve teaching and learning.

Daily, data collected from the real world is becoming larger in both size and dimensions. The increasing complexity of data induces the nascence of new patterns which do not conform to expected behaviour or deviate from the majority of data objects in terms of some features. These objects are called outliers and the process of finding them in the data is called outlier detection. Definition of outliers usually depends on the context of the data and its domain. For example, black swan events in finance and history, or spam messages in mailbox.

In educational data, outliers can be considered in different possible dimensions: as students, teachers, courses, or learning strategies (Fig. 1.1). However, the existing works devoted to outlier detection in the educational domain concern identifying students with uncommon and rare characteristics [3, 4, 5, 6]. This might be explained by several reasons. First, in the educational environments students can be considered as consumers, especially, in the case of fee-paying education. Consumers should be always satisfied by provided services (i.e. teaching and learning). Therefore, their studying experience has to be analysed and improved. Secondly, measuring students' performance, attitude, or engagement is the best and easiest way to evaluate the educational system. The data about students is simply accumulated, it can be represented in more or less union way, and it usually contains more information. For example, in the task of investigating one course with one teacher and one group of students we will be able to accumulate one row of course's features, one row of teacher's features, but N (where N is the number of students in the group) rows of features for students. Nevertheless, outlier detection in education remains a hardly investigated subject, which requires more examples and studies. In this thesis, outlier detection is proposed as a tool for reinforcing the analysis and prediction in the educational domain. In the following sections, we discuss how outliers can improve the understanding of analysed data and enhance existing prediction practices.

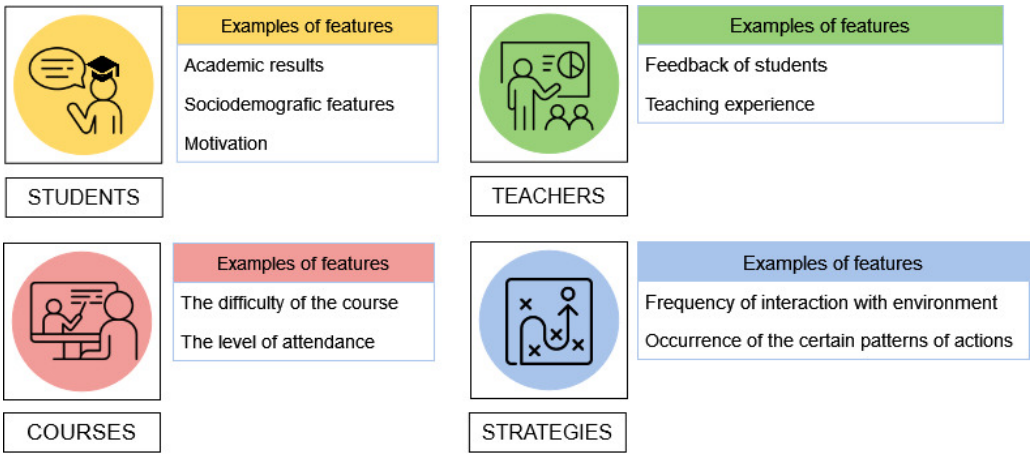


Figure 1.1: Possible dimensions of outliers in educational domain.

1.1 Outlier detection as a tool for data understanding

Modern data-based approaches allow to gain required information, find patterns and relations in data. The identifying and analysis of interesting or unexpected patterns are very important in many domains since they might show the significant and relevant insights.

In the beginning, the main goal of outlier detection was deleting them from the analysed dataset in order to avoid incorrect results. Sometimes built models do not reflect the true real world. This usually happens when data contains anomalies, which leads to model misspecification and biased parameter estimation. Therefore, detection and removing such kinds of outlying observations was a widely used step towards extracting a coherent analysis. Later, outlier detection acquired new forms and contexts: anomalies, discordant observations, exceptions, faults, defects, damage, surprise, novelty, peculiarities, or contaminants in various application domains. Thus, the researchers started to emphasize the importance of understanding outliers and the reasons for their arising. As an illustration, a vast number of studies seek to investigate the factors which cause the appearance of hospital high length of stay outliers [7, 8]. The length of stay in hospital is directly linked with hospital costs. Therefore, identifying the length of stay outliers can contribute to better knowledge of hospital costs and help the management of these institutions control those costs. Another example is related to the food industry. In [9] authors investigate the safety attitude behaviour of Khebab vendors. The findings of the study show that outliers have the following tendencies of attitude behaviour: home-based food safety socialization or customer dissatisfaction. Authors assert that understanding the basis for behavioural outliers in food safety practices can be vital for persuading and transforming future unfavourable food safety behaviour.

Outliers might be aberrations caused by errors in measurement and recording. Likewise, they may contain valuable information. In any case, it is important to question and analyse outliers. The following questions should be asked: Why/Where/When outliers are occurring? What they might mean? The answer could differ from domain to domain, but it is important to have the responses rather than ignore the data, regardless of the significance.

1.2 The challenges of analysis and prediction of students' performance

Currently, many educational institutions seek to improve the educational systems through developing the analytical tools that enable to analyse students' performance and extract important knowledge from data. Such tools can bring benefits for all stakeholders of the educational process: administrators, teachers, and students. For this reason, educational data mining and learning analytics have developed in the past decades [10].

One of the main efforts in the educational field is to build models able to precisely predict students' performance. Student performance includes such tasks as dropout prediction, final score prediction, and students' classification. Despite all these tasks focused on different research objects, they have the same global goal: to detect drawbacks in learning system and detect students with learning problems.

In order to produce better results, existing works in performance prediction examine various features (e.g. sociodemographic or academic results), machine learning algorithms models (e.g. classification or clustering), and study cases (e.g. online courses or specific courses). The recent works achieved good prediction results [11, 12, 13]. Yet, latterly domain experts put forward the model's unfairness as the one of the main problems, which arises when building prediction models. The model's unfairness appears when the prediction model does not work similarly for all students, which is explained by their diversity and characteristics, as gender, race, or nationality. For example, Gardner *et al.* in [14] show that the model of prediction dropout in massive open online courses performed worse for female students than for male students. Yu *et al.* in [15] pointed out that when the race data was included in the model, students of several racial backgrounds were inaccurately predicted to perform worse. Ogan *et al.* in [16] indicated that models built using data from learners in the USA, Costa Rica, and the Philippines were more accurate for students from their own country than for students from other countries. To go deeper, we can also mention the difference of students' cultural codes (mentality) which affect their learning strategies and further might cause biases in performance analysis. The best example of this is the study of tolerance of cheating in Russia, Netherlands, Israel, and the USA,

that show that tolerance to cheating for students from Russia and Israel is higher than for students from USA and Netherlands [17].

To prevent the model unfairness the most evident solution is to extend the existing datasets by all possible information about students. However, sooner or later, this theory faces with such obstacles as privacy and ethical principles or technical difficulty to collect this information [18]. Therefore, instead of accumulation of more information to build models, it is important to find another possible way to improve the understanding and quality of existing data. The best option to improve the data quality is data preprocessing, which includes feature selection, data cleaning, filling of missing values, and outlier detection. Thus, in the thesis we consider outlier detection as a tool which might bring the contribution into existing practices of performance prediction.

1.3 Contribution

The main contribution of this thesis consists of investigation of outliers in educational data. Specifically, outlier detection is proposed as a tool for enhancing existing practices of data analysis and prediction in Educational Data Mining and Learning analytics.

Outlier detection in education is a barely investigated topic, which requires more examples and studies for educational real-world data. Therefore, the thesis presents three case studies of outlier detection for different educational contexts and way of data representation (numerical dataset for the German University, numerical dataset for the Russian University, sequential dataset for French nursing schools). For each case study, we proposed the data preprocessing approach based on the dataset peculiarities, e.g. characteristics of the students' measurement in the certain educational environment. The prepared data has been used to detect outliers in the conditions of unknown ground truth. The characteristics of detected outliers have been explored and analysed that allowed extending of knowledge about students' behaviour in a learning process.

Many educational institutions aim to develop essential mechanisms which will help to improve academic results and reduce attrition. Thus, they build the models of performance prediction which are able to detect students with learning problems that need special help. Yet, these models usually work for students with

common characteristics, and they ignore students-outliers. In the thesis we explored the impact of outliers on performance prediction models. We considered the outliers effect in two of the most widespread prediction tasks in the educational domain: the dropout prediction and final score prediction. The outcomes of the experiments show that models can be improved by deleting of outliers from the train data. Thus, outlier detection might be a part of the pipeline of designing prediction models for educational environments.

1.4 Thesis organization

The rest of the thesis is organized as follows:

Chapter 2 is devoted to the detailed discussion and identification of outlier detection. Such aspects of outlier detection as the types of anomalies, application of outlier detection, input data characteristics, various types of supervision, threshold setting problem, the taxonomy of existing outlier detection techniques are explicitly explained. Moreover, the chapter describes the state-of-the-art unsupervised outlier detection algorithms implemented in the thesis.

Chapter 3 observes the progress of educational data mining and learning analytics (EDM and LA). It addresses to types of environments and analysed data in EDM and LA, refers to the peculiarities of educational data and its preprocessing, considers the existing applications, presents the tools and techniques commonly used by practitioners and researchers, and discusses the outlier detection task in the education.

Chapter 4 proposes three case studies of data preprocessing and outlier detection in education. The observed educational datasets vary in terms of the context (students' performance in the German University, students' performance in the Russian University, the in-game behaviour of students from French nurse schools) as well as in terms of the way of data representation (numerical features of students' achievements, records of students-system interaction). Thus, we perform the data preparation methods and outlier detection based on distinctive characteristics of the analysed datasets. Moreover, this chapter investigates the characteristics of outliers detected by different algorithms, and compares their outcomes.

Chapter 5 performs the analysis of the impact of outliers on performance

prediction models. Two types of models are investigated: (i) dropout prediction (ii) final score prediction. The prediction models are compared in terms of different prediction algorithms and the presence of outliers, detected in the chapter 4, in the training data.

Chapter 6 summarizes the thesis and the main results, and concludes with some perspectives and future research directions.

Chapter 2

Outlier detection

Normality is a paved road: It's comfortable to walk, but no flowers grow on it.
— Vincent van Gogh

2.1 Introduction

With the elaboration of new technologies, data collected from the real world become larger in both size and dimensions. The increasing complexity of data induces the appearing of new patterns which do not conform to expected behaviour or deviate from the majority of data objects in terms of some features. The identifying and analysis of interesting or unexpected patterns are very important in many domains since they might point out the valuable and relevant information. As an illustration, unexpected geological activity in nature can be a predecessor of the earthquake or tsunami. These unexpected patterns are called outliers and the process of finding them in data is called outlier detection.

Probably the first definition of outliers was given by Grubbs in 1969: an outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs [19]. Later, Hawkins determined outlier is an observation that deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism [20]. Since then, outliers have been continuously researched and characterized in various studies [21, 22, 23, 24, 25]. Generalizing all definitions, outliers are observations in data, which markedly differ from the norm, defined regarding their features. Furthermore, their number is usually significantly smaller than the proportion of common cases.

Despite the definitions of Grubbs and Hawkins are relevant today, in 2021, the motivation of the outlier detection has changed. Previously, the main goal of finding outliers was data cleansing – removing them from the data, due to the

sensitivity of the existing models to observations that deviated from the majority of observations. However, after 2000, researchers started to pay attention to outliers since they could point interesting features, suspicious events, or abnormal data records.

In the literature, the outliers are also mentioned as anomalies, exceptions, noise, or some peculiar definitions depending on the specific applications (e.g. faults, grey sheep, white crows) [26, 27]. Nevertheless, they always refer to the problem of finding rare patterns that do not conform to common or expected behaviour. Therefore, Golstein and Uchida defined the main important characteristics of outliers as follows [28]:

1. Outliers are different from the norm with respect to their features;
2. Outliers are rare in a dataset compared to normal instances.

The importance of outlier detection is caused by the fact that outliers in a dataset can bring significant and relevant information for application domains. Outliers exist in almost every real data set and might be summoned by a variety of reasons:

- **Human error.** This reason is caused by the human factor, such as data reporting error.
- **Instrument error.** Instruments' defects or wear and tear can induce the appearance of outliers.
- **Natural deviations in populations.** In this case, the outliers are purely natural, due to the variety of individuals (e.g. very tall people). However, their characteristics can bring interesting insights to a domain field.
- **Fraudulent activity.** The dishonest or deceitful activity in some systems, e.g. credit/card fraud, terrorist activity, spam.
- **Changes in systems or faults in systems.** Outliers occur due to the change in the established system or environment. For example, new buying patterns among consumers, genes mutation.

Outlier detection is a widely researched topic, which has an important concept in the field of data analysis. Various applications nowadays, in order to ensure system reliability, try to find patterns in data that do not conform to expected distribution or behaviour. For example, outliers can point to malicious activity in the monitoring of credit card transactions. Traders can use outlier

detection to monitor individual shares or markets and detect novel trends which may indicate buying or selling opportunities. Also, outlier detection can find anomalous objects on the image. Despite the vast variety of applications exist, the key purpose of outlier detection remains the same: to explore the hidden and unseen area to better understanding the processes in systems.

Usually, the outlier detection approach defines the area that represents normal behaviour and after, proclaims all objects which do not belong to this area as outliers. However, there are several factors, which make this task challenging. Figure 2.1 presents the general scheme of outlier detection, where the main aspects are depicted in polygons, and factors impacting them are situated below.

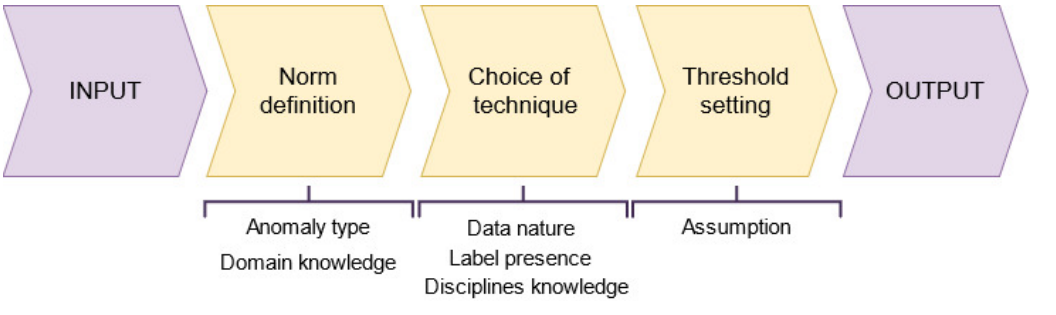


Figure 2.1: The general scheme of outlier detection.

The first aspect of outlier detection an identification of the norm, i.e. defining the area with normal and expected behaviour or characteristics. The norm strongly depends on the context, for example, the temperature +30°C in Toulouse is not anomalous for summer; however, it is abnormally high for winter months. The knowledge of a specific, specialized discipline, profession, or activity help to precise the purposes of outlier detection. Furthermore, it is important to understand the type of anomalies, since the outlier object can be a point with extremely high/low value of the certain feature or an object which belongs to low-density area.

The second aspect of outlier detection refers to choosing the technique or the algorithm. Outlier detection techniques have been explored and developed in various disciplines such as statistics, machine learning, data mining, information theory. Often, the development of an outlier detection algorithm relies on the concept and ideas used from one or more knowledge disciplines. Another factor, which impacts the choice of outlier detection technique is data nature. Data

objects might be in different types such as numeric, binary, categorical or continuous. They can contain various number of features. For instance, the technique based on Euclidean metric can not be applied to categorical data. Therefore, outlier detection technique has to be chosen according to the analysed data. From a data mining perspective, anomaly detection is broadly classified into three following categories: supervised, semi-supervised and unsupervised methods. The main difference between these types of methods is whether a domain expert labelled sample of data as normal and abnormal (outliers and inliers) or not. Here, the fact of presence of labels impact on choice of outlier detection technique.

The third aspect of outlier detection concerns setting the boundary between normal and outlier areas. This task addresses the problem of choosing the threshold. Usually, outlier detection algorithms output an anomaly score for each data object. The objects are considered as outliers if their anomaly score is higher then set threshold. Some algorithms set boundaries of normality during processing and automatically set a threshold. However, these approaches often require user-specified parameters such as the number of clusters. Other approaches need user-defined parameters to define the size or density of neighbourhoods for the outlier. In any case, it is always the task of assumption, where a researcher has to choose some parameters or values about rareness, apartness, or density of outlying objects.

The formulation of an outlier detection problem depends on various factors and the availability or unavailability of some resources. This chapter is devoted to the detailed discussion and identification of these factors. Section 2.2 describes the types of anomalies. Section 2.3 discusses the application of outlier detection. Section 2.4 is dedicated to the consideration of input data characteristics. Section 2.5 explains the difference between various types of supervision. Section 2.6 refers threshold setting problem. Section 2.7 presents the taxonomy of existing outlier detection techniques, while Section 2.8 describes the state-of-the-art unsupervised outlier detection algorithms implemented in the thesis.

2.2 Types of anomalies

As was stated earlier, the main purpose of outlier detection is to define the area of normality and find objects which do not belong to this area. However, there are various scenarios when this task becomes ambiguous. For example, if we consider the height of students in school, 170cm seems pretty normal, however, if we consider this feature in the context of primary school, such height is abnormal. Therefore, the definition of the required outlier is an important factor in the detection process. There are three types of anomalies based on their content and relation to the other data objects.

The first type of anomalies is point anomalies - individual outlying instances, which deviate from the rest of the data instances [24]. The majority of available outlier detection techniques are developed for this type. In Fig.2.2a, a simple point anomaly detection is illustrated, based on the example of a two-dimensional dataset. Looking at this example, the outlying instances O_1, O_2, and O_3 can be immediately identified. Sometimes, with appropriate visualization, outliers are easily detected by humans. However, there is no possibility to visualize more than three dimensions (forth can be represented in plots as item color or size). Therefore, the task of detection of point anomalies becomes more advanced and requires the implementation of outlier detection algorithms.

The second type of anomaly is contextual anomalies. In this case, the point can be seen as normal, but when a given context is taken into account, the point turns out to be an anomaly [28]. This type is probably the most often occurring type when processing real-world data. For example, according to the statistics of student loan debt by age in 2020, from the USA Department of education [29] in Fig. 2.2b, we can see that the major part of dept is distributed between 24-34 y.o. and 35-49 y.o intervals. The standard repayment plan for federal student loans is calculated on a 10-year timeline, but borrowers can choose an income-driven repayment plan which allows them to make smaller payments over 20 years. This fact explains the 51-61 y.o interval, however, the interval more 62 y.o. is definitely abnormal. The integration of the context in outlier detection always requires the domain knowledge.

The third type of anomaly is collective anomalies when a subset of data is outlying with respect to an entire dataset. Each instance from this subset is not necessarily a point anomaly, but only a specific combination of them defines

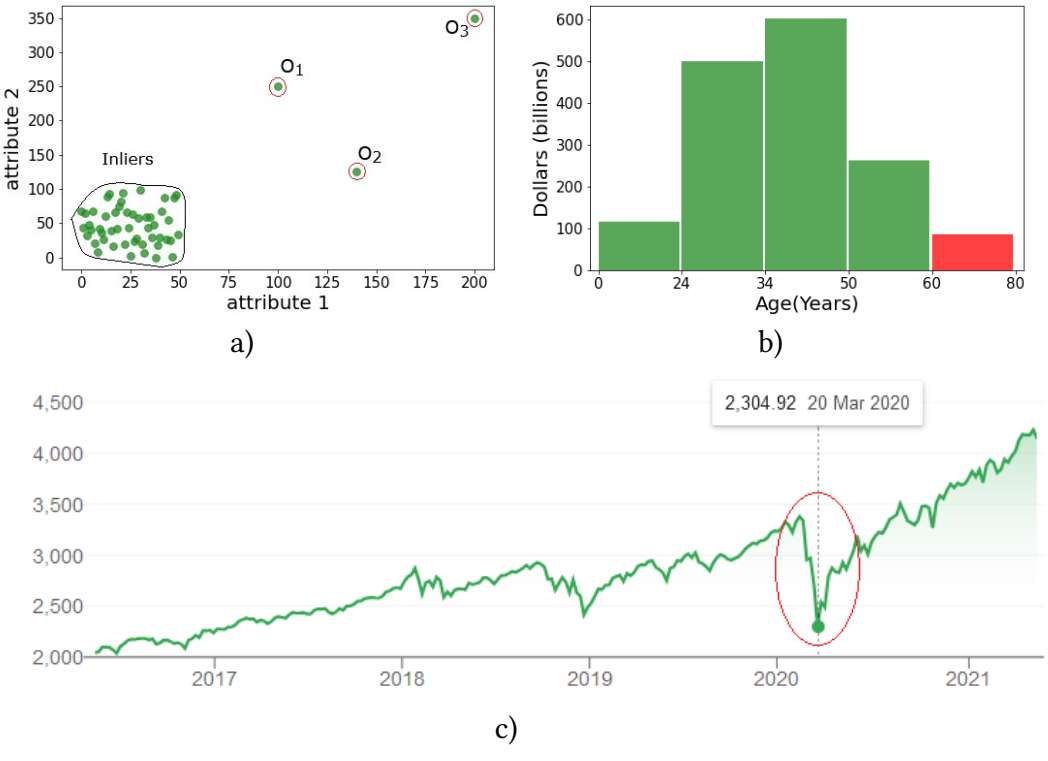


Figure 2.2: Types of anomalies. **a)** Point anomal ies **b)** Contextual anomalies **c)** Collective anomalies.

the anomaly. Theoretically, many complex relationships between data instances exist. However, the most common example of collective anomalies are anomalies with sequential nature, e.g. a drop in stock prices amid the COVID-19 crisis (Fig.2.2c) [30].

2.3 Overview of outlier detection applications

Outlier detection is an actively researched topic with a wide range of application domains. This results in highly diverse literature on outlier detection techniques. As was stated earlier, the domain knowledge plays a key role in a process of outlier detection, especially in the detection of contextual anomalies with specific outlier scenarios. Thus, outliers are also often called intrusions, fraud, exceptions, misuses, novelties, or irregularities depending on the application domain. The most common and well-researched areas for outlier detection are discussed

in detail in [24, 23, 28].

- ***Intrusion Detection.*** In this application domain unauthorized access in computer networks is detected. The Intrusion Detection Systems (IDS) aim at preventing malicious software, hacking attempts or suspicious network activity [31, 32]. There are two types of IDS: host based and network based. Host-based IDS refers to detection anomalies based on data available on a certain computer, whereas network based IDS monitor network traffic, e.g. Internet Protocol (IP) packets.

- ***Fraud Detection.*** This application scenario concerns the detection of malicious activity, which usually appears in the bank systems, credit card systems or insurance agencies. According to [33], the majority of research studies in the fraud detection corresponds to bank transfer area and insurance fraud area (more than 80% of observed papers). However, during the past decades, with developing online stores and mobile telecommunications, such areas as e-commerce fraud detection and telecommunication fraud detection have been widely researched [34, 35, 36].

- ***Medical applications.*** The outlier detection in the medical domain has been applied and used in many various ways. For instance, anomaly detection in medical wireless sensor networks, which are used for remote monitoring of patient vital signs [37]. Also, anomaly detection in image data to detect tumours in digital mammography or brain magnetic resonance [38].

- ***Industrial damage detection.*** Outlier detection in this application domain aims at detecting faults in mechanical units or structural defects through the analysis of sensor data. The outlier detection monitors industrial mechanisms such as motors, turbines, oil flow in pipelines and detects defects that might occur due to wear and tear or other reasons. [39, 40].

- ***Specialized Applications.*** Besides the described above common applications, there are many studies of outlier detection for specific cases. For example, Chandola in [24] defines image processing as an application for outlier detection, which can be used in the medical domain as well as in industrial damage; Goldstein in [28] emphasizes data leakage prevention system application. Furthermore, with the development of domains, where

data collection became possible, the problem of outlier detection arises. For instance, detection of fake news in social media [41]; detection of users who have unusual preferences in recommending systems [27]; search for unusual patterns in the large volume of logs [42]; outlier detection in activities of daily living [43]. This thesis considers an outlier detection in the educational domain, which usually concerns the finding of students with abnormal characteristics [3, 5].

2.4 Data nature

One of the main factors of outlier detection is the nature of the analysed data. First of all, the input data contains *objects (instances)* which are supposed to be characterized as inliers or outliers. These objects can be represented in various forms, e.g. data points, vectors, patterns, or records. Secondly, each data object is described by the set of *attributes (features)*. Each data object can have one attribute (*univariate*) or several attributes (*multivariate*). For instance, in the analysis of the weather changes, timestamps (day, hours, minutes) are data objects, and the temperature value at the certain time is an attribute. If we consider just temperature values – data objects are univariate, however, if to add the precipitation or wind features – data objects become multivariate.

Attributes attached to each data point can be set out in different scales, such as binary, numerical, or categorical. Sometimes data have spatial or sequential nature. Therefore, data prepossessing is an important part of outlier detection. Chapter 4 of the thesis shows the data preprocessing for three data sets, which have different nature: sequential data and data with numerical attributes; where the data transformation and implantation of appropriate similarity metric allow to apply the same outlier detection algorithms.

2.5 Label presence

From a data mining perspective, outlier detection techniques are classified into three following categories: supervised, semi-supervised, and unsupervised methods. The main difference between these categories of methods is the presence of labels (outliers and inliers). Supervised outlier detection requires a full labelled

dataset, semi-supervised assumes availability of labels just for one class, and unsupervised is intended for unlabelled data.

Supervised outlier detection is basically the task of supervised machine learning. The presence of labels normal/abnormal enables to train and test classifier model to predict outliers (Fig. 2.3). The main problem of supervised outlier detection is the unbalanced data: the number of outliers is usually much less than the number of inliers. Therefore, the algorithms which effectively deal with unbalanced training data should be used. Supervised outlier detection is rare due to the following reasons: (i) the labelled data is rarely available in the real-world data collection context; (ii) the behaviour of outliers is often dynamic in nature, new types of outliers might arise, for which there is no labelled training data.

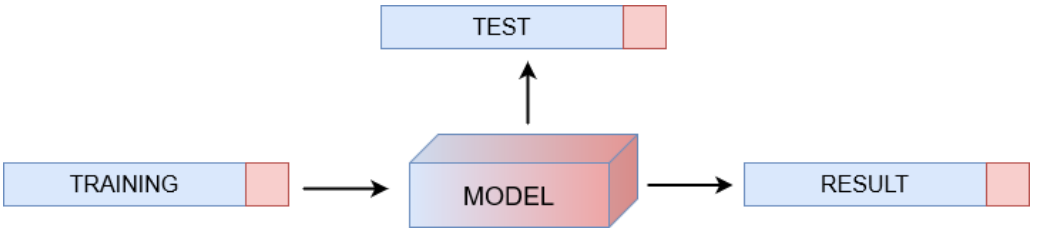


Figure 2.3: Supervised outlier detection.

Semi-supervised outlier detection uses labels for one class since the difficulty of collecting labels for another class. For example, in some fields, an outlier scenario corresponds to extremely rare events, such as a plane accident. In this case, the simplest way to build a model which represents only normal class and points out anomalies that are not normal according to the model (Fig. 2.4). For instance, authors in [44] use outliers for training: they define normal patterns of activity or stable behaviour of a system/process, which needs to be monitored; afterward, generate a set of detectors; then, monitor new observations for changes by continually matching the detectors. Nevertheless, the labels for the normal class are usually easy to gain, because usually, normal behaviour is well-defined. For this reason, semi-supervised techniques which use normal data for training are more popular.

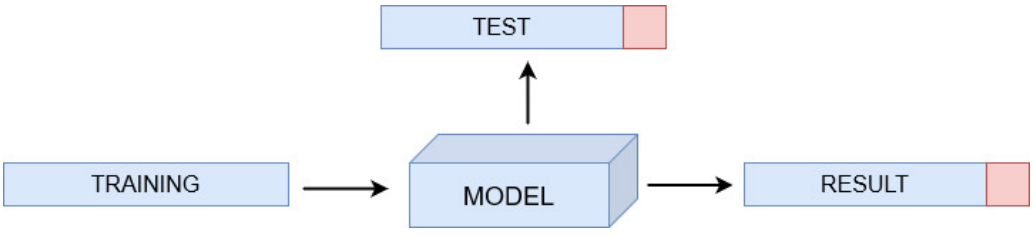


Figure 2.4: Semi-supervised outlier detection. Example of using normal data for training.

Unsupervised outlier detection is the most widely applicable and the most challenging. In this case, data contains as inliers as outliers without any information about labels. These techniques always make assumptions about the data: how many outliers data contains, how far they should be from the normal instances (Fig. 2.5). In the frameworks of unsupervised outlier detection, there are two types of techniques: algorithms for detection of global outliers and algorithms for detection of local outliers. *Global outliers* are data points that are far away from the majority of data points, whereas *local outliers* are anomalous with respect to their local neighbourhood.

Figure 2.5: Unsupervised outlier detection.

In this thesis we detect outliers in conditions of unknown ground truth when information about labels is not available. We consider methods for detection both local and global outliers, however, mainly focus on global outliers since they differ from the majority of observations.

2.6 Threshold setting problem

The problem of the threshold setting appears only in unsupervised outlier detection when the ground truth (which objects are outliers and which are inliers) is unavailable. Unlike supervised outlier detection, when output is a binary class label indicating the normal or anomalous class membership, unsupervised outlier

detection usually has numeric output. This numeric output is the so-called *outlier (anomaly) score*, which defines the degree of outlines for each data instance. The anomaly score is easily converted to binary output by setting the threshold, which is basically the assumption of the domain expert about analysed data. After the calculating anomaly score, the domain expert has to assume which part of the data instances are outliers. As a thumb rule, in the outlier detection field, the partition of the outliers usually does not exceed 10%, however, it depends on the application domain and other circumstances related to the certain context.

2.7 Outlier detection techniques

In the literature, a diverse set of outlier detection techniques and their classifications have been discussed. The most common outlier detection techniques, mentioned in the majority of existing surveys, are statistic-based, distance-based, density-based, and cluster-based techniques [23, 24, 45, 46]. Some studies are focused on the specific data structure or data context. Thus, Akoglu *et al.* consider graph-based techniques [47], whereas Aggarwal addresses to ensemble-based techniques [48]. With developing interest to machine learning, techniques based on learning have occurred [49, 50].

Since many outlier detection techniques in many areas and applications have developed, it is impossible to mention all of them. For this reason, existing taxonomies do not provide a general structure of outlier detection techniques classification and usually mention different types of techniques. Fig. 2.6 contains the types of outlier detection techniques which are described in detail in surveys cited above. All these techniques can be considered in terms of two aspects: discipline knowledge and data nature.

Discipline knowledge aspect concerns algorithm which based on components of a certain discipline. For instance, cluster-based outlier detection techniques use clustering methods, information theory-based techniques utilize theoretical information theory measures such as entropy, information gain, spectral decomposition techniques deal with principal component analysis. The data nature aspect refers to algorithms proposed for data with a specific structure, like time series or sequential pattern mining. The reminder of this section observes the most used outlier detection techniques which were implemented in the thesis.

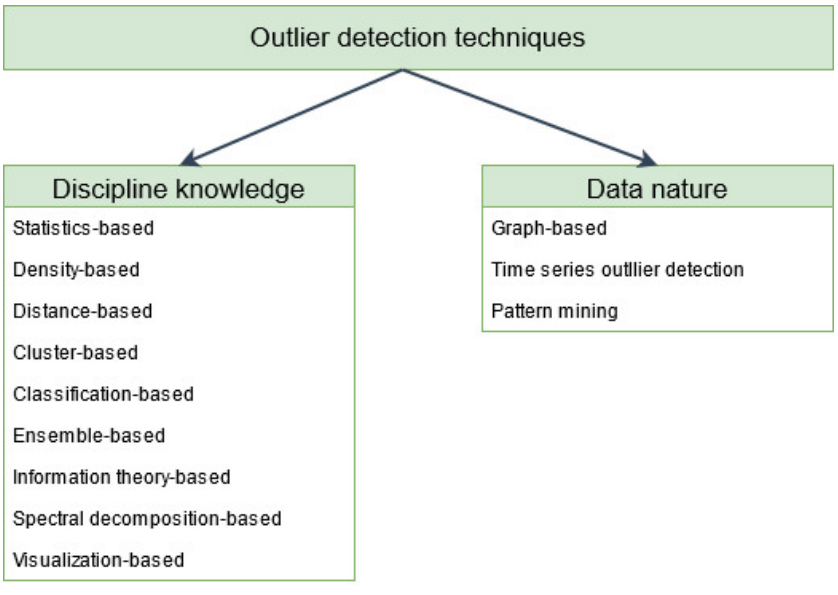


Figure 2.6: The classification of outlier detection techniques.

2.7.1 Statistics-based techniques

The statistics-based techniques assume that the data is generated from a known distribution. Outliers are usually defined according to the probability distribution that indicates the possibility of being an outlier, but sometimes they are specified depending on the relationship with the distribution model. The statistic-based techniques are the earliest approaches that were used for outlier detection. For instance, well-known technique Z-value proposed by Grubbs [19] in 1969, which is basically the difference between the mean value and the data object value divided by the standard deviation, where the mean and standard deviation are calculated from all feature values including the data object value.

Statistical outlier detection approaches are usually classified into two major groups – parametric and non-parametric methods. The parametric techniques assume that analysed data generated from the known distribution. The non-parametric techniques do not have any assumption about distribution.

The most prevalent parametric statistical approaches used for outlier detection are Gaussian models. These models assume that data is normally distributed. The training step usually involves estimating the mean and variance for the distribution using the Maximum Likelihood Estimate. For the testing step, some statistical tests are applied. The simplest test is the box-plot test which visual-

izes data distribution as a box, where borders of the box correspond to maximal and minimal values, which data can take. If a data object is beyond these borders, it is defined as an outlier [51]. Besides Grubbs's test mentioned earlier, such tests as Rosner test [52] and Dixon test [53] are widely utilized. Some studies propose Gaussian mixture model methods, which assume the different distribution for inliers and outliers [54, 55, 56]. Another group of parametric techniques is based on regression. These methods are the most straightforward: at the training step, the regression model is constructed; at the testing step, every data instance is evaluated against the model. In this case, an outlier is a data point with a remarkable deviation of the actual value from the anticipated value obtained by regression model [45, 57].

Nonparametric techniques do not make assumptions about data distribution. The first group of nonparametric statistical outlier detection techniques concerns histograms. Such approaches create bins that refer to the frequency of occurring in data instances. The outlier object is usually defined as a point that belongs to a bin with rare frequency [58]. Another group of non-parametric methods is kernel density estimation methods. The underlying principle of these methods is a comparison of each point's local density with its neighbour's local density [59, 60].

The statistics-based techniques are mathematically acceptable and easily implemented. However, some shortcomings can be pointed out. Firstly, parametric methods require the assumption about distribution. This makes obtained results unreliable for real-life situations when the knowledge about distribution is not available. Therefore in scenarios with unknown distribution, nonparametric techniques are usually used. Furthermore, a lot of statistics-based techniques are applied to univariate feature space and are limited in multidimensional scenarios. Some studies try to adopt statistical methods to increasing of dimensionality which usually leads to the processing time increase [46].

2.7.2 Distance-based techniques

The distance-based technique utilizes the distance between each pair of instances. The general idea presumes that the neighbourhood of each object is defined by the distance threshold. In other words, if the neighbourhood of an object O contains few elements, then an object O is regarded as an outlier.

Depending on distance threshold definition, outliers can be identified as:

- the object O is an outlier, if at least p of objects lies greater than distance D from O [61];
- the object O is an outlier if it belongs to top n objects whose maximal distance to their k-th nearest neighbour are the greatest [62];
- the object O is an outlier if it belongs to top n objects whose average distance to their k-th nearest neighbour are the greatest [63];

k-nearest neighbour methods (kNN) are the most commonly used for distance-based outlier detection since they work well for the detection of global outliers. Algorithms proposed by Knorr and Ng [61] and Ramaswamy *et al.* [62] have been adopted and enhanced in terms of complexity and implementation. For instance, Angiulli *et al.* proposed algorithm which detects top outliers in unlabelled dataset [64]. Ghoting *et al.* developed Recursive Binning and Re-Projection (RBRP) that improved computational speed for high-dimensional data [65]. Huang *et al.* presented a Rank-Based Detection Algorithm (RBDA) algorithm, which ranks the neighbours. This method ensures that the nature of high-dimensional data becomes meaningful.

2.7.3 Cluster-based techniques

Cluster-based anomaly detection techniques operate on the output of clustering algorithms. Clustering methods are unsupervised since they do not require prior information about labelling. Firstly, they split data objects into groups (or clusters) applying any clustering algorithm, where each group contains similar objects. Hence outliers are defined as objects which either belong to a small sparse cluster or do not belong to any cluster (belong to monoclusters).

At the step of clustering objects, any clustering algorithm can be chosen. Agglomerative and divisive clustering algorithms are splitting the complete dataset into clusters in a hierarchical order. Along with them, expectation-maximization (EM) based algorithms are widely implemented, for example, the well-known k-means algorithm. EM algorithms iterative compute cluster assignment and centroid (also medoid) calculation, which is usually faster than hierarchical approaches.

Clustering-based techniques are commonly used in an unsupervised scenario when ground truth is unknown. Some of them intend to detect global outliers as Cluster-Based Local Outlier Score (CBLOF) [66], or unweighted Cluster-Based Local Outlier Score (uCBLOF) [67]. Other cluster-based algorithms were designed for the detection of local outliers such as the Local Density Cluster-Based Outlier Factor (LDCOF), which is an extension of uCBLOF, and the Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise (DBSCAN) [68].

Since cluster-based methods are unsupervised, they are robust to different types of data. However, they rely on the specification of parameters by a user, e.g. number of clusters.

2.7.4 Density-based techniques

The density-based techniques are based on the determination of sparse regions in the data in order to identify outliers. For one-dimensional cases, histogram-based or grid-based methods can be used. However, the definition of density becomes more challenging with increasing dimensionality.

The idea of density-based is to compare the density around an object with the density around its local neighbours. The basic assumption of density-based outlier detection methods is that the density around an inlier object is similar to the density around its neighbours, while the density around an outlier object is significantly different from the density around its neighbours.

The density-based methods intend to find local outliers since they consider outlines of objects according to their neighbourhood. The simplicity and effectiveness of density-based methods have made them widely adopted to detect outliers. The most applicable density-based outlier detection technique is Local Outlier Factor (LOF) [69], which estimates local reachability density for each data object. Another well-known density-based algorithm proposed by Jin *et al.*, which measures local outliers based on symmetric neighbourhood relationships. [70]. These methods served as a baseline for several approaches [71, 72].

Despite some density-based methods perform well performance, they are sensitive to parameter settings such as in determining the size of the neighbours.

2.8 State-of-the-art unsupervised outlier detection algorithms

2.8.1 k-nearest-neighbor

The k nearest-neighbour outlier detection algorithm finds outliers in data relative to their neighbourhood. The basic idea is that outliers are data points which are distant from their neighbours or which have sparse neighbourhoods. Based on the distances to the neighbours, the outlier score can be computed in two ways: 1) the distance to the k^{th} -nearest neighbour (only a single one) is computed [62], 2) the average distance to all the k-nearest-neighbours is used as a score [73].

Firstly, a k-nearest-neighbour graph is defined, where every vertex has k edges to the k nearest vectors. The weight of the edge is the distance between vectors. Afterward, according to a predefined threshold for edge weight (distance between points), instances are detected as outliers if at least one point in their k-neighbourhood is further than the threshold.

kNN algorithm works well for detecting global outliers. The task of creating kNN graph has O(n^2) complexity due to computing the distance between every point. The results of the algorithm are very simple to understand and equally easy to interpret. However, it becomes more difficult, when the number of analysed features increases. Furthermore, the algorithm requires predefined parameters: k – number of nearest neighbours. In practice, the value for k should not be below 10 so that it is assured that a minimum neighbourhood is used for estimating the local density. On the other hand, too large values may lead to similar scores of all instances. Usually, k should not be larger than 50 in practice. When working with real-world datasets, a small change on the k usually does not dramatically change the results.

2.8.2 Cluster-based local outlier factor

The process of merging similar objects into groups is known as clustering. Cluster-based outlier detection utilizes the output of clustering algorithms. They assume that outliers usually belong to sparse and small clusters. The initial step followed by these algorithms is to classify the clusters as outliers and inliers clusters. Cluster-based local outlier factor labels objects as outliers according to both

the size of the cluster the object belongs to and the distance between the object and its closest cluster as described in [66].

The detection algorithm begins with clustering of all points in data set D according to a predefined distance and clustering algorithm. Hence, resulting clusters are divided into large and small clusters sets according to the following conditions:

$$\begin{cases} |C_1| + \cdots + |C_b| \geq |D| \cdot \alpha \\ |C_b|/|C_{b+1}| \geq \beta \end{cases} \tag{2.1}$$

Where $|C_i|$ is number of objects in the cluster C_i, $|D|$ is a number of objects in dataset D, α and β are predefined parameters. Then, the set of large clusters is LC $= [C_i$, if $i \leq b]$, and the set of small clusters is SC $= [C_i$, if $i > b]$.

This cluster-based algorithm is based on two conditions (1) that are applied to the clustering results. Clustering can be conducted with implementation of any algorithm for clustering of objects and predefined metric between objects.

2.8.3 Histogram-based outlier score

Histogram-based outlier score (HBOS) is a combination of univariate methods that allows identifying outliers in dataset as objects, which are different from the majority of the data. The algorithm starts with constructing univariate histogram for each analysed feature d. If the feature we are looking at is categorical, then each bin of the histogram corresponds to a category and the height corresponds to the relative frequency. For numerical features, either static bin-width histograms or dynamic bin-width histograms can be used [58]. Afterwards computed histograms are normalized such that the maximum height is 1.0. This normalization step makes each analysed indicator have equal weight. Finally, the HBOS of every instance p is calculated using the corresponding height of the bins where the instance is located:

$$HBOS(p) = \sum_{i=0}^{d} \log\left(\frac{1}{hist_i(p)}\right) \tag{2.2}$$

According to predefined threshold, observations are divided into outliers and inliers.

This algorithm can be compared with discrete Naive Bayes probability model, which uses multiplication of probabilities. Here the sum of the logarithms is

equivalent to multiplication ($\log(a \cdot b) = \log(a) + \log(b)$). However HBOS is less sensitive to errors due to floating point precision in extremely unbalanced distributions.

HBOS is a fast and simple to interpret algorithm. It works in linear time O(n) in case of fixed bin width or in O($n \cdot \log(n)$) using the dynamic bin width which can lead to a significant decrease of running time, especially on large data.

2.8.4 Local outlier factor

The local outlier factor (LOF), proposed by Breunig, et al. in [69], is based on the concept of local density, where locality is given by the k-nearest neighbours algorithm, whose distances are used to estimate the density. By comparing the local density of an object to the local densities of its neighbours, LOF can identify regions of similar density. Points that have a substantially lower density than their neighbours are considered outliers.

First, the k-nearest neighbour graph $N_k(x)$ is built for each instance x. The $k - distance$ of x is the distance from x to the k^{th} object in the set. Then the reachability distance $reach\_dist(x, y)$ is computed as the maximum of either the distance from x to y or the $k - distance$ of y. Then the local reachability density is identified for each instance as:

$$LRD_k(x) = 1/\left(\frac{\sum_{o \in N_k(x)} reach\_dist_k(x, o)}{|N_k(x)|}\right) \tag{2.3}$$

Finally, local outlier factor is computed as:

$$LOF(x) = \frac{\sum_{o \in N_k(x)} \frac{LRD_k(o)}{LRD_k(x)}}{|N_k(x)|} \tag{2.4}$$

The normal instances have a score 1.0, while outliers have a larger score. Scores below 1.0 indicate that instances are in very dense areas. Due to the local approach, LOF can identify outliers in a data set that would not be outliers in another area of the data set. For example, a point at a "small" distance to a very dense cluster is an outlier, while a point within a sparse cluster might exhibit similar distances to its neighbours. However, the resulting values are hard to interpret. A value of 1 or even less indicates a clear inlier, but there is no clear rule for when a point is an outlier. In one data set, a value of 1.1 may already be an

outlier, in another dataset and parameterization (with strong local fluctuations) a value of 2 could still be an inlier. These differences can also occur within a dataset due to the locality of the method. Breunig, et al. [69] mentioned that k parameter is important for meaningful results, and LOF scores are only stable when $k \geq 10$.

2.9 Conclusion

Outlier detection is an important topic that is widely researched in many applications. The key purpose of outlier detection is to find objects in data that deviate from the norm. This definition induces the first, the most important task of outlier detection - determining the normality and abnormality spaces and the border between them. Obviously, this task is more complex for unsupervised scenarios, when data objects are not labelled. Nevertheless, labelling of data is usually made by a domain expert, who has to decide which objects are outliers. In this chapter, we plunged into general outlier detection and considered the theoretical aspects of the topic.

In Section 2.2, three types of anomalies are presented. The type of anomaly is closely related to the norm definition, especially in the task of detecting contextual anomalies, when data objects are abnormal in a certain context. The simplest and most applicable type of anomaly is point anomalies - individual points which deviate from the rest of the data. These anomalies can occur in a wide range of cases, such as simple 2D visualization or students' performance data, where the points correspond to students. The last type of anomaly is collective anomalies when a subset of data is outlying with respect to an entire dataset. Such anomalies are typical for time series.

The Section 2.3 overviews the applications of outlier detection. Since outliers appear in almost every real-life dataset, many applications of outlier detection exist. This thesis is devoted to the educational domain of outlier detection, which mostly concerns finding students with characteristics that significantly deviate from characteristics of the majority of students.

The Section 2.4 points out the data nature aspect in the outlier detection process. Firstly, the input data objects can be described by one or several attributes. Secondly, it can be represented in different modes: points, records, vectors, etc.

This refers to the data prepossessing task which is observed in detail in Chapter 4.

In Section 2.5 the problem of label availability is observed. Concerning label presence, there are three modes of outlier detection: supervised, semi-supervised, and unsupervised. In real-life scenarios, the labelled data is usually unavailable, therefore unsupervised models are the most applicable. At the same time, they are the most comprehensive, because they are always based on the assumptions of which part of data can be considered as outliers, and require predefining of parameters. For this reason, in Section 2.6, the threshold setting problem is discussed.

The Section 2.7 overviews the outlier detection techniques. Many outlier detection techniques in many areas and applications have been developed, they can concern various disciplines (e.g. clustering, information-based theory) or data structure (e.g. graph-based, time series). The section provides a detailed description of such outlier detection techniques as statistics-based, distance-based, cluster-based, and density-based techniques since they are mostly used in outlier detection field. Section 2.8 represents the state-of-the-art unsupervised outlier detection algorithms implemented in this thesis.

Chapter 3

Educational data mining and learning analytics

Study the past if you would define the future.
— Confucius

3.1 Introduction

Data science impacts many aspects of our life. It has been involved in many areas, and education is not an exception. The increase of e-learning resources, instrumental educational software, the use of the internet in education, and the establishment of state databases of student information have created large repositories of educational data [10]. The analysis of this data can point out drawbacks in educational systems. This induces the task of developing techniques and data-based models which aim to achieve the essential teaching and learning.

In the educational domain there are two main communities, which share the same interests, however, differ fundamentally:

> **Educational Data Mining (EDM)** is the application of data mining (DM) techniques to this specific type of dataset that come from educational environments to address important educational questions [1].

> **Learning Analytics (LA)** is measurement, collection, analysis, and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs [2].

Both, EDM and LA are two areas that tend to enhance educational practices. Yet, LA is more focused on the educational challenge whereas EDM pursues the technological challenge. EDM concerns developing new algorithms and models,

finding new descriptive patterns and predictions that characterize learners behaviors and achievements, domain knowledge content, assessments, educational functionalities, and applications. Meanwhile, LA embeds existing technical solutions to further decision-making: it suggests actionable insights, allowing education institutions not only to better know what happens and understand what has happened, but also to predict with a high level of confidence what will happen or to recommend actions on how make something happen [74].

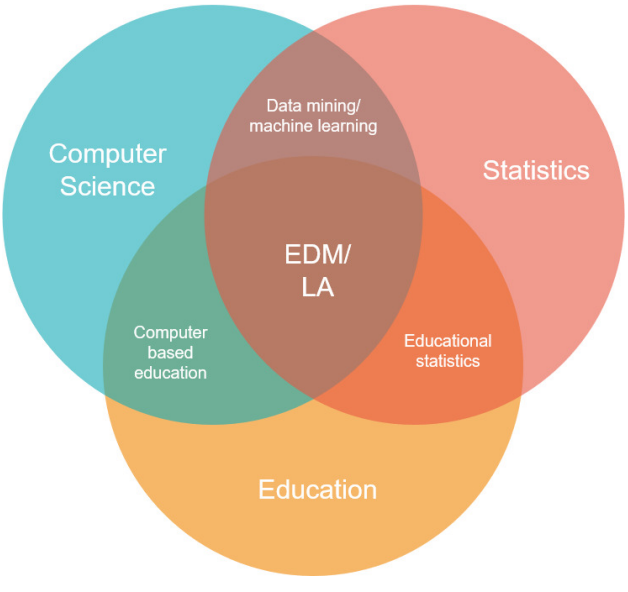


Figure 3.1: Combination of disciplines which form Educational Data Mining and Learning Analytics.

EDM and LA involve the implementation of techniques from various disciplines, such as visual data analytics, recommender systems, information retrieval, etc. Generally, EDM and LA can be described as a combination of three fields: education, statistics, and computer science (Fig. 3.1). Furthermore, the intersections of these areas form subareas which closely linked with EDM and LA. The intersection of computer science and education generates computer-based education (CBE), which implies using computers for education. The best example of CBE is massive open online course (MOOCs) - web-based classes designed to support many students. The intersection of education and statistics forms educational statistics, which is collecting, analysing, interpreting, and presenting educational data using statistical instruments. Finally, the intersection

of computer science and statistics constitutes data mining and machine learning areas, which aim to provide complex analysis or models for big data in many domains. Additionally, with development of educational systems, environments, and strategies, new branches of EDM and LA have appeared, such as Teaching Analytics [75], Serious Game Learning Analytics [76], Data-Driven Education and Data-Driven Decision-Making in Education [77].

In the thesis EDM and LA are considered jointly since they both aim to discover knowledge that can be used in evaluation to assist educators, establish a pedagogical basis for decisions, or modifying an environment or teaching approaches. The Fig.3.2 depicts the components of extraction of this knowledge.

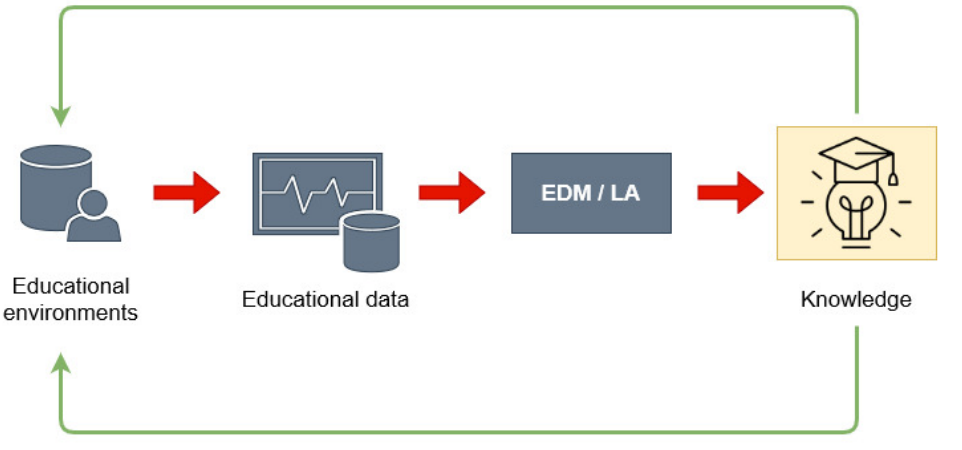


Figure 3.2: The Educational data mining/Learning analytics knowledge extraction process.

Firstly, the educational data is produced by educational environments. Depending on the type of educational environment different kinds of data can be collected. For instance, traditional classroom education can provide the data about students' performance, interaction with other students, while MOOCs can yield data about the time of interaction with the platform, or discussions at forums. The next step of knowledge extraction is applying EDM and LA techniques, which include data prepossessing, analysis, and interpretation of results. Finally, extracted knowledge is used to enhance existing educational environments or creation of new ones.

The utilization of EDM and LA allow to analyse a lot of information (e.g. learning objectives, learning activities, learning preferences, participation, com-

petencies, performance, and achievements) and convert it to essential decisions or recommendations, which might improve the quality of educational systems and learning process. This chapter observes the progress of EDM and LA. The remaining of this chapter is organized as follows: Section 3.2 addresses to types of environments and analysed data in EDM and LA, Section 3.3 refers the peculiarities of educational data and its preprocessing, Section 3.4 considers the existing applications, Section 3.5 presents the tools and techniques commonly used by practitioners and researchers, Section 3.6 observes the existing studies devoted to outlier detection in the education.

3.2 Environments and analyzed data

Educational data can be gathered from various sources with different formats. It comes from three types of learning environments:

Traditional education. Traditional classrooms for primary, secondary, higher education, etc.

Computer-based education. Learning management systems, intelligent tutoring systems, computer supported collaborative learning, serious games, test and quiz systems, etc.

Blended learning. The combination of traditional face-to-face education and with computer-based education. Hybrid learning, mixed-mode instruction, etc.

The collected data is highly variable depending on the type of environment and can contain such information as:

- performance data (grades, number of passed/failed exams);
- administrative data (school, faculty, teacher information);
- demographic data (gender, age);
- student affectivity (motivation, emotional states)

Bousbia and Belamri in [78] propose other features of analysed data in EDM, such as:

- data availability (recorded data, generated during experiments data, data in benchmark repositories);
- collection sources (manually collected data, digitally collected data, mixed of manual and digital approaches);
- the educational described level (the keystroke level, the answer level, the session level, the student level, the classroom level, the teacher level, and the school level).

Collection and integrating of data are non/trivial tasks. Sometimes data is available in an inappropriate for analysis format, and it is necessary to convert it. Furthermore, to answer the certain research questions, certain data has to be collected and analysed (e.g. features of students' grades in one course can not fully show students' motivation in study). Educational data has some peculiarities which should be taken into account while data preparing for further analysis. Thus, the following section is devoted to peculiarities of educational data and its preprocessing.

3.3 Peculiarities of educational data and its preprocessing

Nowadays, the immense data growth evokes the processes of gathering useful information or organized knowledge to be understood or extracted automatically. This is why Data Mining (DM) has recently evolved and implemented in many domains. DM is the process of discovering interesting patterns and knowledge about them from large amounts of data [79]. DM is also known as a synonym of the Knowledge Discovery in Databases (KDD) process; however, sometimes DM is considered as an essential step in the process of knowledge discovery [80]. One of the most important steps of DM or KDD is data preprocessing. Usually, gathered data is performed in multiplied forms and comes from different sources. It is not cleansed, changed, or transformed. Therefore, it may include a lot of errors or missing values. This can lead to wrong interpretation of results of data analysis and further biased decision making. Data preprocessing allows transforming data to a suitable form for resolving problems by data mining techniques. Basically, the better data is preprocessed, the more useful information can be extracted.

The data preprocessing step requires a lot of time and manual work. According to Pyle [81], data preparation consumes 60-90% of the time needed to mine data - and contributes 75-90% to the mining project's success. The tasks of data preprocessing usually seek either to find and change imperfections in datasets (e.g. dealing with missing values, cleaning duplicates) or to transform data to appropriate for data mining format (e.g. feature selection, data rescaling). Fig. 3.3 shows the main steps of data preparation process. The first step is devoted to data gathering, which brings together all available data from different sources, e.g. log data, quiz/test data, portfolio, etc. Then the data which came from different sources integrated or aggregated into a database. At the cleaning step erroneous or irrelevant data is detected and discarded. Usually, this step includes detection of point outliers and working with missing data. Feature selection, also known as feature reduction or attribute selection, chooses an optimal subset of relevant attributes according to a certain criterion. The criterion must be defined consistent with the purposes of feature selection, e.g. subset of attributes that gives maximal predictive accuracy. Finally, data transformation creates new attributes from already available attributes. This allows to better interpret information, and sometimes reduce calculation complexity. The most commonly used types of transformation are normalization, discretization, derivation, and format conversion.

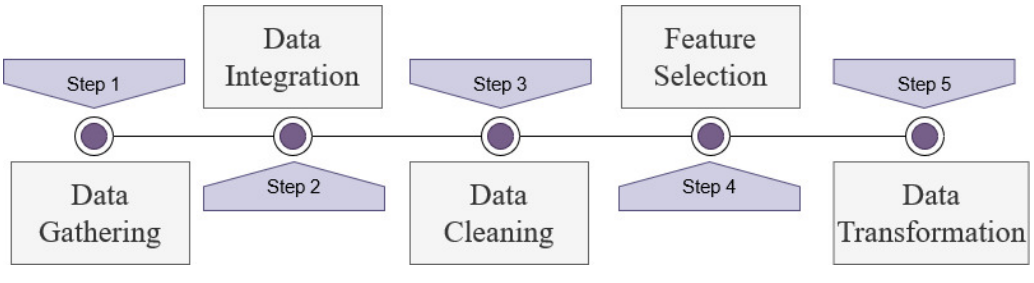


Figure 3.3: Main data preprocessing steps.

The data preprocessing for educational data, generally, has the same tasks as data preprocessing for other domains. However, the educational domain has some peculiarities which distinguish it from others. Romero *et al* in [82] pointed out specific tasks of data preprocessing in educational data:

1. Educational systems provide a huge amount of student information generated daily from different sources of information.

Educational data is usually gathered from three sources: (i) Log files - records all students-system interaction; (ii) quiz/test - information about quiz/test utilization; (iii) portfolio - information about students.

The log files contain information about students' interaction with the environment, which usually record events in software occurring after mouse clicks. The log data easy to record and convenient for storing different information. For example, information about students' actions and choices while interacting with serious games environment, time of their occurrence, information about students' ID, and some extended knowledge about these actions such as virtual places which learner visited while exploring the game environment [83, 84]. Quiz/test data is usually collected in the $Score \times Question$ form, where each column is related to the certain question or task, and a row performs score for this task [85]. Finally, the portfolio data includes row data about each student. This data may contain as achievements of students during learning process (e.g. attended courses and scores) as personal information (e.g. educational program, age, gender) [86].

Many educational institutions collect data from different sources which are aggregated into relational databases, e.g. databases with students' portfolios, quiz data, and information about learning supporting sources. These databases have large and complex structures. Therefore, the task of collection, aggregation, and integration of educational data is very important. Optimized data storage helps quickly extract valuable information for flexible analytical processing.

2. Normally, all students do not complete all the activities, exercises, etc. In consequence, there is often missing and incomplete data.

   Missing data is general issue which occurs in data analysis process. There are many possible solutions to deal with this: delete rows with missing values, fill in the missing value by constant (e.g. fill missing values by 0.0), use substitute value (e.g. mean or mode), use linear interpolation or linear regression.

   This task requires deep domain and data knowledge since wrongly replaced missing values can significantly impact data analysis outcomes. For instance, in German universities, the best score for the passed exam is 1.0

while 5.0 score means a fail. If one replaces missing score values by 0.0 and apply a data mining technique based on a numerical scale (e.g. clustering), it will show that students without performance are similar to students with excellent performance, instead of students with a bad performance.

3. The user identification task is not normally necessary.

   The user identification distinguishes the users as individuals. This can be done using IP addresses, cookies, and/or direct authentication (login/password). The identification of users and their sessions is not specific for education, since the data can be stored as with the user ID as a primary key. It is noteworthy to mention that usually some information about students/educators can not be included to database due to privacy issues [18].

4. There are usually a great number of attributes available about students and a lot of instances at different levels of granularity. So, it is necessary to use attribute selection and filtering tasks in order to select the most representative attributes and instances that can help to address a specific educational problem.

   Feature selection is a very important task that helps to handle many practical situations. The most common application is to reduce the number of attributes to improve the accuracy of prediction models or avoid overfitting. Additionally, many attributes increase the complexity of models, which are difficult for interpreting. The selection of important attributes for a certain task allows simplifies computations and interpretation.

5. Some data transformation tasks, e.g. attribute discretization, can be normally applied for improving the comprehensibility of the data and obtained models.

   Data transformation of educational data can be performed in different forms for different tasks. It can be normalization or standardization of numerical attributes, aggregation of several attributes into one, encoding attributes, etc.

3.4 Applications EDM and LA

There is a wide range of EDM/LA objectives and it is too difficult to create a taxonomy which will cover all possible tasks. For example, Romero and Ventura in [1] proposed to classify EDM/LA objectives according to their final user:

Learners. Supporting learners in their study process, providing adaptive feedback or recommendations, improving performance and preventing dropout, etc.

Educators. Improving learning process, providing new teaching methods, analysing pedagogical strategies, analysing pedagogical materials, etc.

Researchers. Developing and comparison of algorithms to be able to provide a recommendation for specific educational task, data prepossessing, etc.

Administrators. Organization of institutional resources (human and material) and their educational offer.

This classification shows the advantages of EDM/LA applications for each final user. However, there are cases when objectives are related to more than one user category. Therefore, some studies propose classification of EDM/LA applications based on general goals of the fields [78, 10, 87]. The remainder of this section observes the applications considered in the thesis.

3.4.1 Evaluation and monitoring of students' learning

The evaluation and monitoring of students' performance is an important aspect of education. Assessment of learning provides valuable information that might help all stakeholders of the educational process (students, instructors, administrators, or policy-makers) to make decisions. The modern data mining techniques allows to monitor student performance and discover hidden information in educational systems.

Applying EDM and LA to the performance data, researchers identify students' behaviour and their learning strategies [88], perform student profiling [89], and predict student achievement [90].

3.4.2 Dropout and retention

Many educational institutions seek to decrease the dropout rate. The increasing rate of dropouts negatively affects social aspects, such as reducing the number of

people with higher education, and economic aspects such as financing students who do not complete their studies. This leads to a growing interest in examining factors that might impact a dropout [91].

Although many studies about dropout prediction exist, there is no consensus about the best ways to understand the changing nature of this phenomenon regardless of the pedagogical style or activity used in the course [87]. To produce better results, existing works in dropout prediction examine various features (e.g. sociodemographic or performance), machine learning algorithms models (e.g. classification or clustering), and study cases (e.g. online courses or specific courses). For example, in [86, 11] authors predict dropout in German and US universities respectively through analysis of students' performance after the first semester. In [92, 93] authors predict student's dropout in MOOCs.

3.4.3 Game learning analytics

Educational applications aim at both improving learning quality in general and enhancing understanding of the learning process. In order to follow these two highlights, new types of educational environments have been developed in recent years. One of these environments that allows improving students' decision-making skills and performance is game [94]. The use of a game with purposes apart from entertainment context is called serious game (SG) [95, 96].

Through serious games players learn from their own experiences [97]. They are designed to educate people about a certain subject, amplify concepts, reinforce development, or help learners to gain learning skills or change an attitude. Thus, serious games are relevant instruments for teaching a lesson, delivering a message, or acquisition knowledge through interactivity, motivation, and engagement. The interaction players with SG produces the data, which can be used to evaluate the game's outcomes and their positive impact. Therefore, from the educational data mining and learning analytics fields, which focus on education generally, the Game Learning Analytics (GLA) builds up. (GLA) is defined as the collection, analysis, and extraction of information from data collected from serious games [76].

GLA has many applications, which impacts different aspects of SG:

GLA can predict game's impact. The data collected through playing can provide measures and features which can help to estimate the game's effec-

tiveness. Some studies assess effectiveness through engagement, motivation, and usability [98, 99, 100], others examine the acquisition of knowledge and skills through learning [101, 102, 103, 104].

GLA can help form student profiling. The creation student's profile can improve learning, including targeted feedback and adaptive learning experiences. The student profiling can be executed with analysis of characteristics, in-game behaviors[105, 106], or clustering [107, 108, 109, 110].

GLA data can validate serious game design. GLA can help to obtain insights and improve serious game design and implementation. Using GLA some studies validated serious game design [111, 112].

3.5 Data mining techniques

The majority of techniques implemented in EDM/LA are acknowledged as universal, such as visualization, clustering, prediction and so on. The most applicable EDM/LA techniques are as follows:

1. **Causal mining**. Investigating causal relationship or causal effect in data. For example, finding features of students' behaviour evoke dropout, investigating features that impact on success in a serious game.

2. **Clustering.** Grouping a set of similar objects in the same group (cluster). For example, merging students to clusters based on similar performance or learning strategies.

3. **Statistics**. Collection, analysis, interpretation, and presentation of data by using statistical methods. For example, exploration of factors impacting success or investigation dependencies between participation of courses.

4. **Recommendation.** Creation of the rating or preference of users. For example, recommendations for students, which courses to attend, based on their preferences and problems with courses taken before.

5. **Visualization.** Graphical representation of data. For example, visualization of large amount of information by representing the data in some visual display (histogram, scatterplot, etc.).

6. **Sequential pattern mining.** Discovering the relationships between occurrences of sequential events. For example, investigation of students learning strategies through their actions in online courses progress or serious game playing.

7. **Prediction.** The process of supervised learning that predicts values or class of one variable through a combination of other variables (e.g. classification or regression). For example, dropout prediction, the final score prediction.

8. **Text mining.** Extraction a high-quality information from the text. For example, parsing the contents of forums, chats, web pages, and documents.

9. **Outlier detection.** Detection of objects with significantly different/deviating characteristics. For example, detection of students with irregular learning process.

3.6 Outlier detection in education

In the educational field, outlier detection techniques mostly focus on analysing abnormal behaviour or noisy reactions [113]. Despite the relevance of outlier detection, it is not well researched in the educational field. According to [87], only 2.25% of investigated papers considered an outlier detection. This section observes existing studies of outlier detection in education and their contribution.

Carneiro *et al.* in [3] perform outlier detection for student assessment in distance learning programs (e-learning) which is based on face-to-face exams. Authors define outliers as students, which do not use resources from the learning platform but still pass face-to-face exams, and detect them applying the isolation forest technique.

Abu Tair and El-Halees in [4] use unsupervised outlier detection algorithms (k-nearest neighbours and local outlier factor) as part of educational data mining to improve graduate students' performance and overcome the problem of low grades of graduate students. Their findings show that detected outliers are students with excellent results in some degrees.

Ueno in [5] describes a system that supports an online outlier detection using Bayesian predictive distribution. This system identifies learners with irregular learning processes using learners' response time data for the e-learning contents.

Oeda and Hashimoto in [6] propose outlier detection as a method for predicting dropouts. Using clustering of log-data, authors compare the time of active behaviour in programming lessons. The obtained clusters show the three main trends of students' outlying behaviour: (i) a high number of inputs (ii) a low number of inputs (iii) suddenly increasing number of inputs, which can be predicted.

3.7 Conclusion

This chapter observes the progress in Educational Data Mining and Learning Analytics. EDM and LA are two interdisciplinary communities that have developed past decades. Both seek to enhance educational and learning practices through the collection, analysis, and mining of data from educational environments.

Section 3.2 considers educational environments and types of data used in EDM and LA. Data in the educational domain can be gathered from various sources, such as traditional classrooms or computer-based systems. Furthermore, the collected data can be represented in inappropriate for further analysis formats. Thus, the domain experts should take into account the peculiarities of educational data and its preprocessing, which are discussed in Section 3.3. Section 3.4. presents the applications of EDM and LA. Since there is a wide variety of tasks in these fields, the section is mostly focused on the application considered in the thesis: evaluation and monitoring of students' learning, dropout and retention, and game learning analytics. In Section 3.5 the taxonomy of mostly applicable techniques in EDM and LA is presented. The majority of techniques implemented in EDM and LA can be used in other fields, such as visualization, clustering, prediction. Section 3.6 observes the existing studies devoted to outlier detection in education. Despite the relevance, outlier detection is not a well-explored field that needs more experimental outcomes to form the concept.

In the first two chapters, we discussed the theoretical aspects and the state-of-the-art of the outlier detection and EDM/LA fields. The following chapters of the thesis are devoted to the contribution and experimental results, which point out the importance of detection of outliers in the educational domain.

Chapter 4

Outlier detection in educational data

By failing to prepare, you are preparing to fail
— Benjamin Franklin

4.1 Introduction

Modern institutions of higher education aim to enhance the educational systems to improve students' learnability. Usually, any changes in the educational process are based on decisions that were made by analysing common and frequent patterns while infrequent patterns can contain valuable hidden information for domain experts. In the educational field outlier detection is usually mentioned as a data preprocessing step, namely, cleaning data by removing inconsistent instances [82, 87]. However, some studies considered in Section 3.6, explored the outlying behaviour of students in different contexts. Despite this fact, outlier detection in education remains a barely investigated topic, which requires more examples and studies. This drives us to detect outliers in educational real-world data in order to extract relevant information which might help better understand students' behaviour in a learning process.

As various potential applications exist, a great number of outlier detection algorithms, both supervised and unsupervised, were developed in the past decades. Supervised algorithms are more restricted than unsupervised methods as they need to be provided with a labelled dataset. Supervised outlier detection algorithms require a set of predefined outlier/inlier data from which they train their model. Nevertheless, usually in real life, we do not have labelled data readily available. The ground truth sometimes is not implicit due to the strong dependence between outliers' definition and their context and type. Moreover, the

behaviour of outliers is often dynamic in nature, for instance, new types of outliers might arise, for which there is no labelled training data [24]. Additionally, even the existing studies devoted to unsupervised outlier detection use labelled data to estimate algorithms' performance [28, 114]. These reasons motivate the need for a detailed benchmark bringing together unsupervised techniques on unlabelled real-world datasets.

In this chapter unsupervised outlier detection is considered as a data mining technique, which requires preprocessed data. We propose three case studies for two types of educational data: numerical data and sequential data. For each case, we consider the peculiarities of data and analyse the characteristics of detected outliers. The anatomy of the chapter is as follows: Section 4.2 refers to the description of the basics of numerical and sequential data; Section 4.3 is devoted to the data preprocessing and outlier detection in numerical data, namely, the performance of students in Russian and German universities; Section 4.4 is dedicated to data preprocessing and outlier detection in sequential data which was collected while students from French nurse schools played the serious game.

4.2 Basics of data types

Since the analysed datasets have two different ways of representation (numerical data and sequential data), in this section we consider the main features of each data type.

Numerical data

Numerical data is a data type performed in numbers, rather than natural language description. Numerical data differentiates itself from other number form data types with its ability to carry out arithmetic operations with these numbers. For example, score or number of passed exams are features presented in the numerical type.

Usually numerical data satisfies the following characteristics:

- Quantitativeness: numerical data has quantitative nature.

- Arithmetic operation: one can perform arithmetic operations like addition and subtraction on numerical data. Due to its quantitative character, almost all statistical analysis is applicable when analysing numerical data.

- Estimation and enumeration: numerical data can both be estimated and enumerated. In a case whereby the numerical data is precise, it may be enumerated. However, if it is not precise, the data is estimated. When computing the GPA of a student, for instance, a 4.495623 GPA is rounded up to 4.50.

- Interval difference: the difference between each interval on a numerical data scale are equal. For example, the difference between 5 minutes and 10 minutes on a wall clock is the same as the difference between 10 and 15 minutes.

- Analysis: numerical data is analysed using descriptive and inferential statistical methods, depending on the aim of the research. Some descriptive-analytical methods include: mean, median, variance, etc.

- Data visualization: numerical data may be visualized in different ways depending on the type of data being investigated. Some data visualization techniques adopted by numerical data include: scatter plot, dot plot, stacked dot plot, histograms, etc.

The numerical data in the education domain usually includes students' performance during their studies, in the $Score \times Question$ form, where each column is related to the certain question or task, and a row performs score for this task. Sometimes datasets may contain other numerical features such as the number of passed exams or the number of attempts to pass exams. The majority of data mining techniques work for numerical data, and it often is easily interpreted. However, educational numerical data has some peculiarities, which has to be considered before applying data mining techniques. For example, if data contains information about students' performance who studied different educational programs, then their features have to be normalized over the program. In the Section 4.3, we present the approach of data preprocessing for numerical educational data, which has been implemented for two case studies: German and Russian institutions of higher education.

Sequential data

Sequential data allows measuring/monitoring some phenomena over the time (time series data) or in a given order (sequential events) without a concrete notion of time. Basically, whenever the points in the dataset are dependent on the other points in the dataset the data is said to be sequential data. For example, sensor data or stock market prices where each point represents an observation

at a certain time point, or gene sequences where there is no dependence over time.

A broad range of real-world applications uses data in a sequential form. **Sequence** is an ordered list of events. An event can be represented in a symbolic value, a numerical real value, a vector of real values, or a complex data type [115].

Let O be considered domain, then a sequence S of length n is a mapping of the index set $I_n = 1, 2, ..., n$ into a domain O:

$$S : I_n \to O$$

- The set of all sequences of length n is $O^n = O^{I_n} = \{I_n \to O\}$

- The set of all sequences over domain O is $O* = \{I_n \to O \mid n \in N_0\}$

- Sequences can be classified by their domain O: categorical values (nominal values, alphabets), and continuous values (real numbers).

Given two sequences $\alpha = \{a_1, a_2, \ldots, a_n\}$ and $\beta = \{b_1, b_2, \ldots, b_m\}$, α is called a **subsequence** of β, denoted as $\alpha \subseteq \beta$, if there exist integers $1 \leq j_1, j_2, \ldots, j_n \leq m$ such as that $a_1 \subseteq b_{j_1}, a_2 \subseteq b_{j_2}, \ldots, a_n \subseteq b_{j_n}$.

Sometimes, the exploring of sequences allows to find out dependencies between events or find similar order of several events, e.g. synchronous drop of prices on the stock market or the same order of similar words in a text. Therefore, a lot of studies are dedicated to sequential pattern mining. **Sequential pattern** is a sequence of itemsets (or events) that frequently occurred in a specific order [116]. Thus, sequential pattern mining is the process of discovering useful, novel and/or unexpected patterns in databases.

Despite the majority studies in this field are focused on investigation of frequently occurring patterns, the research community has witnessed the significance of rare patterns (or outliers) in many domains (e.g. inimical drug reactions can be identified by some rare responses to medications in the field of biology) [117]. In the sequential data field, outlier detection algorithms are mostly developed for time series or for transnational data, where each event can be considered as an independent transaction and their order is not kept (e.g. Apriori or FP-growth algorithms). Therefore, if the data does not have timestamps and the order of events has to be saved, the data preprocessing is needed. In the Section 4.4 we propose the data preparing approach for such case.

4.3 Outlier detection for numerical data

4.3.1 Methodology

Data collection and preprocessing

Two datasets of students' performance were collected through their studies in the university. The datasets are performed in a similar form, where the row corresponds to a certain student, and the columns determine the achievements or characteristics of this student. Fig. 4.1 presents the general scheme of data preprocessing for this kind of educational data.

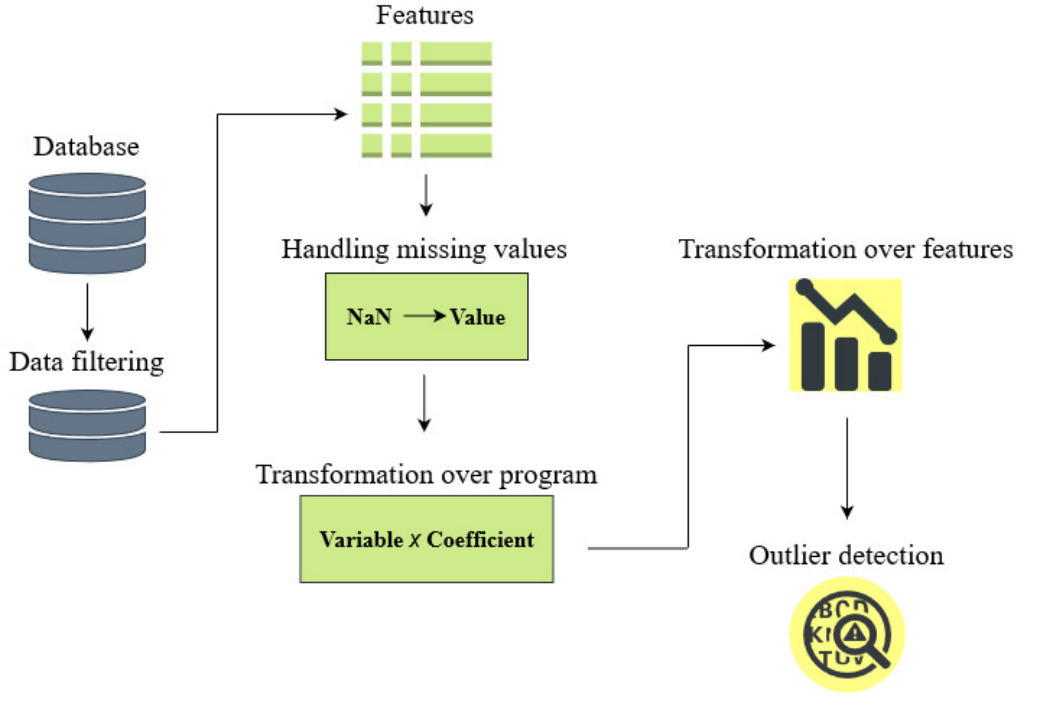


Figure 4.1: Visualization of numerical data prepossessing.

After the data collection, domain experts should define the main goals of the research and filter data according to these goals. For example, if the goal is to investigate the impact of the students' age on their scores, the data used for the analysis has to contain information about age and scores.

Afterward, the representative and easily interpreted features have to be formed. For instance, in the task of investigating students' engagement in a computer-based educational environment, such features as time spent on reading

additional materials might show students' involvement in the learning process.

As was mentioned earlier in the Section 3.3, usually students do not complete all activities and tasks, which evokes the problem of missing values. The missing values should be handled according to the context and type of the data. For example, the missing grades for students who have not passed the exam can be transformed to the value of the worst score, or it can be transformed to a certain value. Then, in the first case, the students who tried to pass an exam, but failed and students who did not try to pass an exam will be at the same level, while in the second case they will be split.

Usually, educational data is collected for students who study different educational programs, which have a different curriculum or scales of measuring the results. Therefore, if data contains students from different educational programs, then their features have to be normalized over the program, by utilizing a coefficient, e.g. the dividing test scores by maximal possible value show the proportion of the correct answers.

Finally, to implement data mining techniques the set of considered features should be transformed to the union scale. The easiest way to do this is to normalize or standardize data.

The analysed datasets were prepared according to the proposed method (Fig. 4.1). Despite the similar general scheme of numerical data preprocessing, each case study has the peculiarities which are described in detail in sections 4.2.2.1 and 4.2.3.1.

Outlier detection and analysis of outlying characteristics

The outlier detection topic is not well investigated in educational domain [87]. Some researchers try to clean data by removing the abnormal values, others do not mention how they handle with outliers. Another issue of outlier detection in educational data is absence of ground truth when the information about which objects are abnormal is non-evident. Therefore, first of all, we identified and explored abnormal values for considered features.

In statistics, if a data distribution is approximately normal then about 68% of the data values lie within one standard deviation of the mean and about 95% are within two standard deviations, and about 99.7% lies within three standard deviations. Thus, abnormal objects were defined as values deviating from the mean at three standard deviations (less than $mean - 3 * std$ or more than $mean + 3 * std$). Unfortunately, the majority of real-world data is not normally distributed. Consequently, interquartile range (IQR) was considered as an additional way to investigate abnormal values. To calculate the IQR, the data which represents one feature is divided into quartiles, or four rank-ordered even parts via linear interpolation, where the first quartile Q1 corresponds to the 25^{th} percentile, the second quartile Q2 corresponds to the median, and the third quartile Q3 corresponds to the 75^{th} percentile. The interquartile range IQR is defined as the difference between Q3 and Q1 ($IQR = Q3 - Q1$). Here, outliers are observations that fall below $Q1 - 1.5 * IQR$ or above $Q3 + 1.5 * IQR$. The interquartile range can be clearly visualized by the box on a box plot, and $[Q1 - 1.5 * IQR; Q3 + 1.5 * IQR]$ interval is visualized as whiskers.

The definition of outliers strongly depends on the context of the analysis. However, outliers always correspond to the two characteristics: (i) they are different from the norm with respect to their features; (ii) they are rare in a dataset compared to normal instances [28]. Therefore, such aspects as the norm and rareness should be determined.

To specify the norm, the similarity metrics between objects have to be chosen. We defined outliers as students who differ from the majority of students according to the set of numerical features. The ground truth indicating which students are outliers is absent and non-evident. Hence, only unsupervised algorithms can be used. Thus, we considered four unsupervised algorithms to detect outliers: distance-based k-nearest neighbours (kNN), cluster-based local outlier

factor (CBLOF), histogram-based outlier score (HBOS), and density-based outlier factor (LOF). The algorithms were compared in terms of relation outlier score value *vs* its rank and a robustness to the change of the parameter. The relation outlier score *vs* its rank has been observed with help of visualization, whereas the robustness to the change of the parameter has been examined by calculating the Spearman's correlation coefficients between the outlier scores yielded with different values of the parameter. Spearman's correlation coefficient has been chosen because it is based on the ranks, which are more indicative than numerical outcomes in the task of outlier scores comparison.

Concerning the rareness aspect, we investigated three assumptions: (i) 3% of students are outliers; (ii) 5% of students are outliers; (iii) 10% of students are outliers. The outliers drawn from the detection algorithms were ranked, and 3%, 5%, or 10% of the students with top ranks were considered as outliers.

To understand the nature and the kind of the detected outliers, k-means clustering has been run, and the number of clusters has been chosen according to the elbow curve and the silhouette criteria. The mean values of features for each cluster have been examined and compared, which allowed emphasizing the outlying characteristics for each case study.

The additional computational details for outlier detection with their explanation are presented in Appendix A in the table A.1. Further, we consider two case studies of outlier detection for numerical data and explore the main characteristics of identified student-outliers.

4.3.2 Case study 1: German institution of higher education

4.3.2.1 Data collection and preprocessing

Student Advice dataset (SA) has been collected in the frameworks of the research project Students Advice - A Data-Driven Approach, based in Berliner Hochschule für Technik, Berlin, Germany [118]. The project aims to exploit the data that universities have about the academic achievements of their current and past students, to devise algorithms and to build tools to: (i) identify the different paths that students follow in their curriculum; (ii) better understand how these paths influence their progress; (iii) use this knowledge to help students who are in difficulty by providing informed personalized advice.

The dataset contains information about students' performance collected dur-

ing their studies in a six-semester bachelor's degree program. The students started their studies from winter 2012 to summer 2019. The data records comprehend courses taken by each student including the earned grade and the respective semester as well as the semesters of graduation or dropout for each student.

The data includes students from three educational programs, who have dropped out or graduated. Each degree program has a curriculum that contains a list of planned courses for each semester. Students may follow this curriculum or not. For instance, they can enrol in courses from the 1st semester when they are in the 2nd semester, and vice versa. The grade scale for passing a course is [1.0; 1.7; 2.0; 2.3; 2.7; 3.0; 3.3; 3.7; 4.0], where the best grade is 1.0 and the worst is 4.0. If a student fails the examination, the grade is 5.0. Students may enrol in courses without taking the exam. To graduate, students must complete all mandatory courses and a program-specific number of electives with a maximum of three exam attempts.

We considered features such as average grades and numbers of courses passed, failed, and enrolled per semester, similar to [11, 119]. In terms of the number of courses passed, we made a further distinction: whether students passed them as planned or earlier, or later than planned. Since the best score is 1.0, while 5.0 score means fail, the empty values that have occurred during aggregation have been handled as follows: if no grade was obtained in the respective semester, the value of the grade is set to 6.0; if no course was passed, failed and/or enrolled in, the respective total is set to 0. The number of planned courses is program-specific (e.g. 5 planned courses in S2 for program 1, and 6 planned courses for program 2), the features related to courses were converted to proportions by division by the respective number of planned courses.

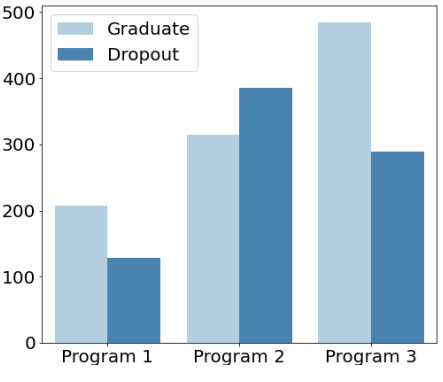
Outlier analysis has been conducted for students after their 1st and 2nd semesters of study. Fig. 4.2a depicts the analysed features and their descriptive statistics. The codes of the features are presented as follows: **S1** and **S2** correspond to the 1st and 2nd semesters, **Av_grade** is the mean semester grade for passed courses, **F_ex** is the proportion of failed exams, **En_ex** is the proportion of courses which students enrolled in but did not take the exam, **P_ex_p, P_ex_d, P_ex_a** are the proportions of passed exams according to the plan, with delay, and in advance respectively.

In contrast to the data after the 1st semester, data after the 2nd semester reveals the development of performance over time, e.g. whether more or fewer

courses were taken or whether exam grades improved. We removed students who dropped out after the 1st semester due to the absence of data for them in the 2nd semester. The considered dataset includes 1 809 students, among them, 1 007 students are labelled Graduate, and 802 students are labelled Dropout. The distribution of labels according to the educational program is shown in Fig. 4.2b.

	MIN	MEAN	STD	MAX
Av_grade S1	1	2.8	1.3	6
Av_grade S2	1	3.2	1.5	6
F_ex S1	0	0.067	0.13	0.8
F_ex S2	0	0.075	0.13	0.83
En_ex S1	0	0.17	0.28	2.8
En_ex S2	0	0.22	0.32	1.6
P_ex_p S1	0	0.7	0.38	1
P_ex_p S2	0	0.5	0.4	1
P_ex_d S1	0	0.041	0.13	1
P_ex_a S1	0	0.078	0.45	5.4
P_ex_a S2	0	0.074	0.16	3.5

a)



b)

Figure 4.2: Data description for SA dataset: a) descriptive statistics of analyzed features; b) distribution of labels Dropout/Graduate for three educational programs.

4.3.2.2 Outlier Detection

The investigation of deviations from the mean ($\notin [mean - 3 * std; mean + 3 * std]$) shows the abnormal values for the following features: a high number of failed exams in both semesters, a high number of enrolments for each semester, a high number of passed exams with delay and in advance for both semesters. Observations fell beyond the whiskers detected for the same features. However, their number is higher than the number of values detected with deviations from the mean. Fig. 4.3 depicts the investigation of abnormal values for analysed features. Here,the box corresponds to the IQR, the horizontal line inside the box shows the median, the whiskers denote the $[Q1 - 1.5 * IQR; Q3 + 1.5 * IQR]$ interval, the circle white point shows mean value, and the colored background denotes the $[mean - 3 * std; mean + 3 * std]$ interval. This means that the majority of students prefer to follow the curriculum, avoid enrolments without passing the exams, and try to avoid fail of exams.

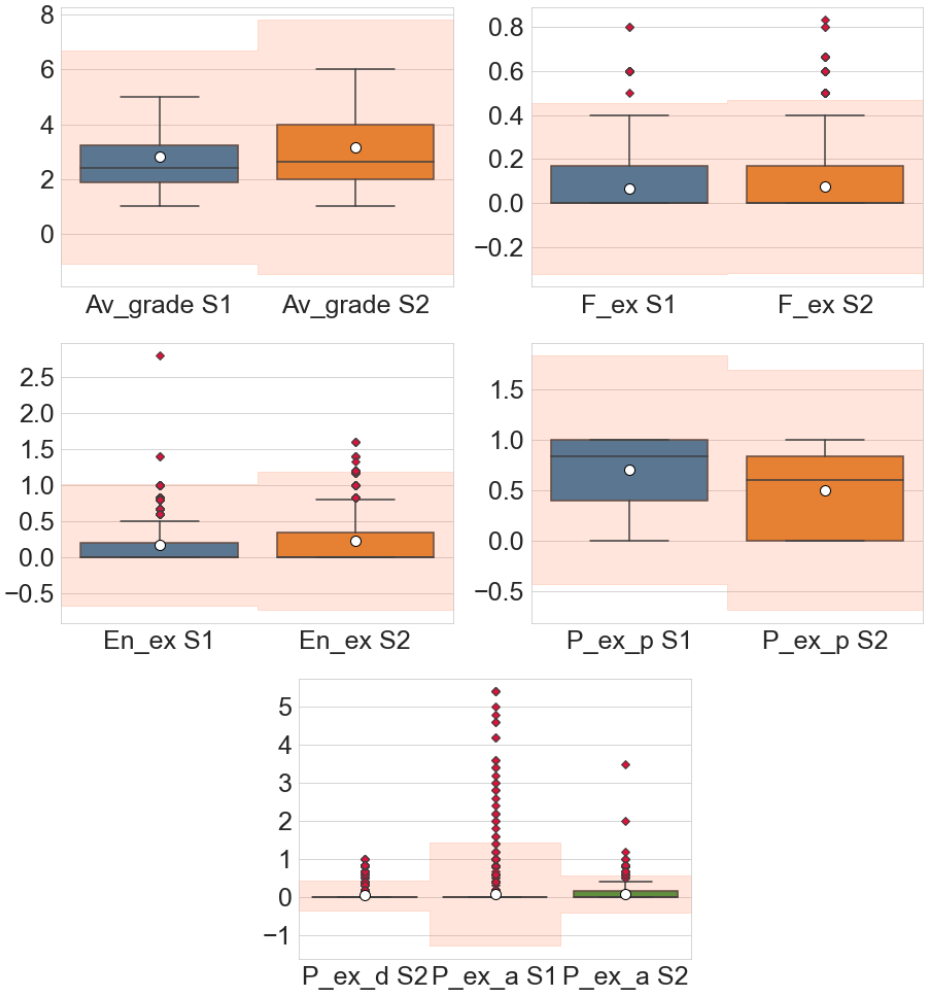


Figure 4.3: The investigation of abnormal values for for analysed features of SA dataset.

Before detecting students-outliers by four unsupervised outlier detection algorithms, their parameters have to be predefined. Fig. 4.4 presents the relation between outlier score and its rank according to different values of predefined parameters: k - the number of nearest neighbours for algorithms kNN and LOF ($10 \leq k \leq 50$), k - the number of clusters for algorithm CBLOF ($6 \leq k \leq 11$), and b - number of bins for algorithm HBOS ($40 \leq b \leq 50$). Algorithms show the considerable difference between outlier scores for different predefined parameters when the rank $\in [0\%; 5\%]$.

To investigate the difference between outlier scores, which were obtained by changing the parameter value, the Spearman's correlation coefficients were

estimated (Fig. 4.5). The outlier scores calculated by kNN, CBLOF, and HBOS algorithms with different parameter values are highly correlated (all $corr.coef > 0.9$), whereas outlier scores for LOF are not robust to the change of the parameter: when $k < 30$ the correlation coefficients are low, however, when $k \geq 30$ the correlation coefficients increase. Therefore, for further outlier detection, the following parameters have been chosen: $k = 50$ for kNN algorithm, $b = 43$ for HBOS, $k = 7$ for CBLOF, since these algorithms are robust to the change of the parameter, and $k = 50$, since the algorithm becomes robust when $k \geq 30$. Additionally, in the case of CBLOF, the parameter k - number of clusters has been chosen with respect to the elbow curve and silhouette criteria.

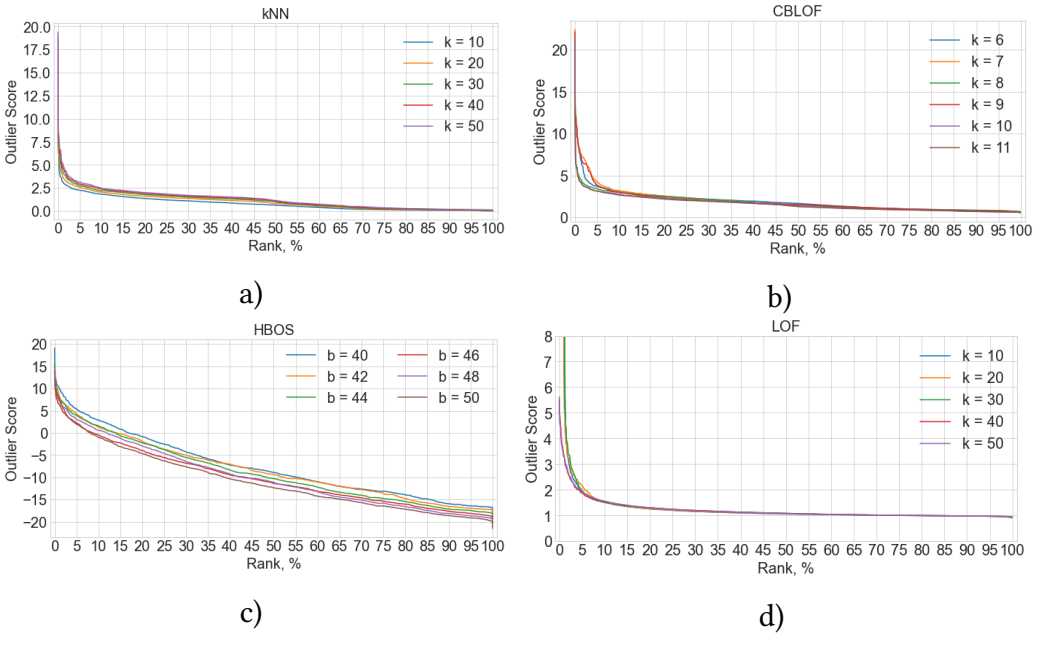


Figure 4.4: Outlier scores detected by unsupervised outlier detection algorithms with different parameters for dataset SA *versus* Rank of outlier score.

The Fig. 4.4 shows that outlier scores for each algorithm swiftly drop until the rank 3%, then substantially decrease until the rank 5%, then considerably go down until the rank 10%, and after slowly decline until the rank 100%. Thus, in the task of outlier detection, we made three assumptions, that 3%, 5% or 10% of students are outliers. Table 4.1 shows intersection rates for outliers detected by different algorithms with different assumptions. The overlap between outliers detected by algorithms kNN and CBLOF is more than 75% for all assumptions.

The intersection rate between LOF and the algorithms kNN and CBLOF is low (11%) and slightly increases with the increasing of the assumption percentage. The HBOS almost does not overlap with other algorithms for 3% and 5% assumptions, while an intersection rate for 10% assumption at the level 27-31%.

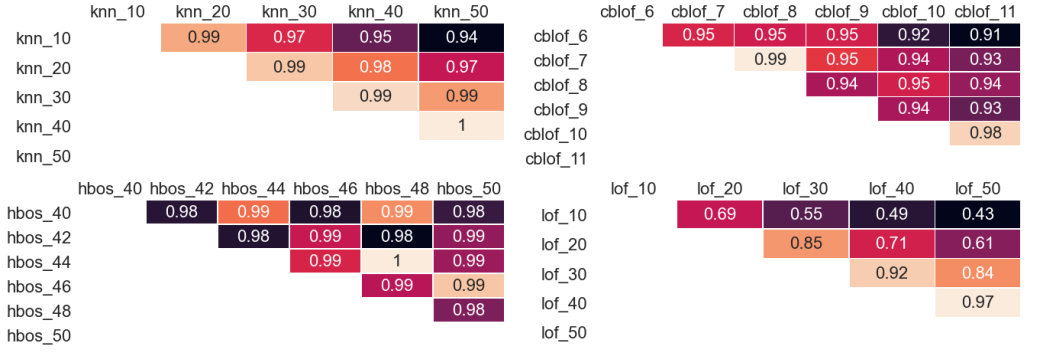


Figure 4.5: Triangular correlation matrices for outlier scores detected by four unsupervised algorithms with different parameter values for SA dataset.

Table 4.1: Triangular matrices with intersection rate for outliers detected by unsupervised algorithms in SA dataset.

	kNN 3%	CBLOF 3%	HBOS 3%	LOF 3%
kNN 3%	1.00	0.75	0.07	0.11
CBLOF 3%		1.00	0.02	0.11
HBOS 3%			1.00	0.00
LOF 3%				1.00
	kNN 5%	CBLOF 5%	HBOS 5%	LOF 5%
kNN 5%	1.00	0.88	0.12	0.30
CBLOF 5%		1.00	0.09	0.24
HBOS 5%			1.00	0.02
LOF 5%				1.00
	kNN 10%	CBLOF 10%	HBOS 10%	LOF 10%
kNN 10%	1.00	0.87	0.31	0.30
CBLOF 10%		1.00	0.27	0.31
HBOS 10%			1.00	0.08
LOF 10%0				1.00

The distribution of labels of detected outliers by each algorithm for each as-

sumption is presented the Table 4.2. The majority of outliers in the case of 3% assumption detected by kNN and CBLOF have labelled Graduate, while the increasing of the assumption percentage increases the part of dropouts. The majority of outliers detected by the LOF algorithm have the label Graduate for all assumptions. Meanwhile, the main part of outliers detected by the HBOS algorithm dropped out in each assumption case.

Table 4.2: Labels distribution of detected outliers in SA dataset.

Alg.	Dropout	Graduate
kNN_3	40%	60%
CBLOF_3	29%	71%
HBOS_3	80%	20%
LOF_3	16%	84%

Alg.	Dropout	Graduate
kNN_5	47%	53%
CBLOF_5	42%	58%
HBOS_5	73%	27%
LOF_5	21%	79%

Alg.	Dropout	Graduate
kNN_10	62%	38%
CBLOF_10	59%	41%
HBOS_10	71%	29%
LOF_10	23%	77%

To find out the main characteristics of detected students-outliers, they have been clustered by k-means algorithm, where k parameter - number of clusters has been defined with the help of the elbow curve and the silhouette criteria. We focus on the clusters gained with the 10% assumption since they include outliers from 3% and 5% assumptions. The obtained clusters for outliers detected by the kNN algorithm are presented in the Table 4.3. The results of clustering for other outlier detection algorithms are performed in the Appendix B. The tables include information about the number of outliers in each cluster (N), their labels (Dropout/Graduate), the number of outliers that are also detected as outliers in 3% and 5% assumptions (N 3% and N 5% respectively), and mean cluster values for each feature. Although the features for outlier detection and clustering were standardized, the tables include average values for non-standardized features of the clusters to help the interpretation.

Clusters of outliers detected by kNN:

Cluster 1 - Intense S1 and missed S2. Students with an extremely high number of passed exams in S1, but without any passed exams in S2: the propor-

tion of passed exams in S1 according to the plan is 0.64, while the proportion of passed exams ahead of the plan is 2.39. Meanwhile, the average grade in S2 is 6.0, which points out the fact that students from this cluster have not passed any exams in S2.

Table 4.3: Characteristics of clusters of outliers detected by kNN algorithm with 10% assumption for SA dataset.

Cluster	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
N	14	22	2	33	35	29	25	21
Graduate	3	22	2	17	17	2	3	3
Dropout	11	0	0	16	18	27	22	18
N 5%	8	22	2	21	17	5	8	8
N 3%	7	21	2	9	6	2	3	5
Av_grade_S1	2.31	2.28	4.58	5.85	2.28	4.08	4.01	4.69
Av_grade_S2	6.00	2.04	2.50	2.61	2.09	4.65	3.13	3.45
F_ex_S1	0.06	0.01	0.00	0.04	0.03	0.14	0.49	0.13
F_ex_S2	0.00	0.02	0.00	0.07	0.02	0.51	0.09	0.14
En_ex_S1	0.24	0.08	0.00	0.17	0.24	0.15	0.14	0.87
En_ex_S2	0.20	0.08	0.00	0.04	0.20	0.37	0.61	0.36
P_ex_p_S1	0.64	0.99	0.00	0.00	0.31	0.34	0.34	0.05
P_ex_p_S2	0.00	0.08	1.00	0.04	0.22	0.07	0.19	0.05
P_ex_d_S2	0.00	0.01	0.83	0.71	0.11	0.04	0.22	0.33
P_ex_a_S1	2.39	2.99	0.20	0.00	0.55	0.10	0.03	0.02
P_ex_a_S2	0.00	0.66	2.75	0.02	0.41	0.05	0.11	0.06

Cluster 2 - Intense S1. Students with an extremely high number of passed exams in S1: the proportion of passed exams according to the plan is 0.99 and passed exams ahead of the plan is 2.99. Unlike cluster 1, this cluster contains students who have performance in S2. The majority of students from this class has the label Graduate.

Cluster 3 - Intense S2. Students with an extremely high number of passed exams in S2. The proportion of passed exams according to the plan is 1, the proportion of passed exams ahead of the plan is 2.75, the proportion of passed exams behind the plan is 0.83. In this small cluster, all students have the label Graduate.

Cluster 4 - Procrastination in S1. Students with a bad performance in S1,

which try to reestablish in S2: they pass exams from S1 in S2 with average marks. The labels of students in this cluster are mixed.

Cluster 5 - Average performance and intention to pass exams in advance. Students with average grades during all semesters, who prefer to pass exams ahead of the plan in each semester instead of exams that correspond to the plan. The labels of students in this class are mixed.

Clusters 6-8 - Various types of bad performance. These students belong to different clusters, and have various characteristics: some of them have low grades in all semesters, some have a high number of failed exams, others have a high number of enrolments without attending the exam. The majority of them have a Dropout label.

Clusters of outliers detected by CBLOF:

Cluster 1 - Intense S1. Students with an extremely high number of passed exams in S1: the proportion of passed exams according to the plan is 0.99 and passed exams ahead of the plan is 3.24. The majority of students from this class has the label Graduate.

Cluster 2 - Intense S2. Students with an extremely high number of passed exams in S2. The proportion of passed exams according to the plan is 1, the proportion of passed exams ahead of the plan is 2.75, the proportion of passed exams behind the plan is 0.83. All students from this cluster have the label Graduate.

Cluster 3 - Procrastination in S1. Students with a bad performance in S1, which try to reestablish in S2: they pass exams from S1 in S2 with average marks (the proportion of passed exams with delay in S2 is 0.7). The labels of students in this cluster are mixed.

Cluster 4 - Average performance and intention to pass exams which do not correspond to the plan. Students with average grades during all semesters, who prefer to pass exams ahead and behind of the plan in each semester along with passing exams that correspond to the plan. The majority students from this cluster has the label graduate.

Cluster 5 - Average performance and intention to pass exams in advance. Students with average grades during all semesters, who prefer to pass exams ahead of the plan in each semester instead of exams that correspond to the plan. The labels of students in this class are mixed.

Clusters 6-8 - Various types of bad performance. These students belong

to different clusters, and have various characteristics: low grades in all semesters, a high number of failed exams, a high number of enrolments without attending the exam. The majority of them have a Dropout label.

Clusters of outliers detected by HBOS:

Cluster 1 - Students with average performance and intention to pass exams in advance. Students from this cluster have the average score for each semester. They intend to pass exams in advance along with exams which correspond to the plan. Students from this cluster tend to graduate.

Cluster 2 - Procrastination in S1. Students with a bad performance in S1, which try to reestablish in S2: they pass exams from S1 in S2 with average marks (the proportion of passed exams with delay in S2 is 0.44). The majority of students from this cluster dropped out.

Cluster 3 - Average performance. Students with average scores per each semester. Students from this cluster prefer to pass exams according to the plan. The labels are mixed.

Cluster 4 - The performance degradation in S2. Students, with a degradation of performance after S1: they have a high number of enrolments in S2 (0.88), and low proportion of passed exams (P_ex_p S2 = 0.01). All students from this cluster have the label Dropout.

Cluster 5 - Average performance with intention to enrol. Students with average scores per each semester, who enrolled courses but did not pass them. The labels are mixed.

Cluster 6 - 8 - Various types of bad performance. These students belong to different clusters, and have various characteristics of bad performance. For example, students from cluster 6 are characterized by a high number of failed exams and enrolments. Students from cluster 7 and cluster 8 tried to pass exams in both semesters; however, they failed the part of them (the proportion of failed exams in S1 for cluster 7 is 0.42, the proportion of failed exams in S2 for cluster 8 is 0.38). The majority of them have a dropout label.

Clusters of outliers detected by LOF:

Cluster 1 - Average performance. Students who follow the plan and pass exams with good marks. This cluster is the biggest one (120 students). The majority of students graduated their studies.

Cluster 2 - Average performance and intention to pass exams in advance. Student from this cluster have average scores and intention to pass exams in advance in both semesters. The labels are mixed.

Cluster 3 - Intense S1 and missed S2. Students with an extremely high number of passed exams in S1, but without any passed exams in S2: the proportion of passed exams in S1 according to the plan is 1.0, while the proportion of passed exams ahead of the plan is 4.10. Meanwhile, the average grade in S2 is 6, which points out the fact that students from this cluster have not passed any exams in S2. The labels are mixed.

Cluster 4 - Intense S1. Students with an extremely high number of passed exams in S1: the proportion of passed exams in S1 according to the plan is 0.99, while the proportion of passed exams ahead of the plan is 2.73. Unlike the cluster 3, students from this cluster have performance in S2. They tend to graduate.

Cluster 5 - Procrastination in S1. Students with a bad performance in S1, which try to reestablish in S2: they pass exams from S1 in S2 with average marks. The labels of students in this cluster are mixed.

Cluster 6 - Bad performance in S1 and missing S2. Students with a bad performance in S1: the average score in S1 is 4.05 and the proportion of enrolments is 1.8. Meanwhile, the average score for S2 is 6.0, which means that students have not passed any exams in S2. Students from this cluster dropped out.

Cluster 7-8 - Bad performance. These two mono-clusters contain dropped out students and have low scores, a high proportion of failed exams, and enrolments.

4.3.2.3 Summary

In this subsection the performance of students in the German institution of higher education was analysed. The investigation of the main tendencies of outlying behaviour showed:

1. The abnormal values were found for the certain features, such as the number of failed exams in both semesters, the number of enrolments for each semester, the number of passed exams with delay and in advance for both semesters. All detected abnormal values correspond to the high values of each feature, which means that majority of students prefer to follow the

curriculum, avoid enrolments without passing the exams, and try do not to fail exams.

2. The unsupervised outlier detection algorithms were explored in terms of the robustness to the change of parameters. Algorithms kNN, CBLOF, and HBOS are robust to the changing of the parameters, while LOF becomes moreless robust when $k \geq 30$.

3. The outlier detected by unsupervised algorithms shows different results. The algorithms for detection of global outliers kNN and CBLOF have similar performance (the overlap for all assumptions > 0.75), while histogram-based HBOS and density-based LOF almost do not overlap with others. The part of outliers with the label Dropout increases with the increasing of percentage assumption for kNN and CBLOF algorithms. The majority of outliers detected by HBOS dropped out for all assumptions. Meanwhile, the LOF mostly detected graduated students as outliers for all assumptions.

4. The clusters of detected outliers by four algorithms revealed the different types of outlying behaviour in the considered dataset. Since kNN and CBLOF outliers have a high intersection rate, their clusters have similar characteristics: students with extremely high number of passed exams; students, who try to reestablish in S2, students with bad performance. The clusters with extremely high number of passed exams can be explained by the recognition of courses that have been passed before enrolling formally (e.g. students who changed their educational program). The clusters of outliers detected by HBOS mostly contain students with different types of average (average performance with intention to pass exams in advance, average performance with high number of enrolments) and bad performance (degradation of performance in S2, high number of failed exams). The clusters of outliers detected by LOF show the various types of average/good performance: the biggest cluster contains 120 students-outliers, who passed exams according to the plan with good marks, the next big clusters (Cluster 2 and Cluster 5) are also characterized by good average marks and high number of passed exams according to/in advance the plan.

4.3.3 Case study 2: Russian institution of higher education

4.3.3.1 Data collection and preprocessing

Tomsk Polytechnic University (Tomsk, Russia) dataset (TPU) contains 6 features that estimate learning progress in the first semester for 1 075 students from 6 engineer faculties in higher mathematics course. Students started their studies in autumn 2018. Names of considered features and their brief descriptions are depicted in the Table 4.4.

Table 4.4: The set of analysed features for TPU dataset.

Feature	Description
ETmath	The score for the enrolment mathematics test
ETph	The score for the enrolment physics test
ETch	The score for the enrolment chemistry test
AT1	The score for the first attestation test of higher mathematics course
AT2	The score for the second attestation test of higher mathematics course
Exam	The score for the examination test of higher mathematics course

When students enter the university, they pass the enrolments tests for three subjects, which are important for the engineering programs: mathematics, physics, and chemistry. These tests include tasks similar to tasks from the union government exam, which students pass for school graduation. After, they attend the course of higher mathematics which has two attestation tests. At the end of the course, students pass an exam that includes tasks similar to tasks from attestation tests. Each test is measured in different numerical scale, for instance the max possible score for ETch is 24, whereas for Exam the max possible score is 5. The descriptive statistics of the feature is depicted in Table 4.5. The test is considered as failed if the percentage of correct answers is 0%- 54%; as passed with an average score if the percentage of correct answers is 55%- 69%, as passed with a good score if the percentage of correct answers is 70%- 89%, as passed with an excellent score if the percentage of correct answers is 90%- 100%. If students fail an enrolment test, the university provides the additional courses by the respective subject in order to help to fulfil the lack of knowledge. If students

fail attestation tests, they continue their studies, then pass the exam. The failed exam must be always repassed to continue the study and finally graduate. If a student does not pass an exam before starting the next semester - he/she loses the scholarship. If a student does not pass an exam after 1-2 semesters - he/she can be expelled.

Table 4.5: The descriptive statistics of the features of TPU dataset.

Feature	**Min**	**Mean**	**Std**	**Max**
ETmath	0.00	2.77	1.17	5.00
ETph	0.00	7.99	4.49	20.0
ETch	0.00	10.57	5.49	24.00
AT1	0.00	6.82	2.93	14.72
AT2	0.00	2.02	1.07	4.62
Exam	0.00	2.89	1.27	5.00

Russian academic system unlike the majority of European systems has the strict curriculum regulations and an intensive workload. However, like european universities the high rate of students attrition happens after the first course of their studies [120]. Therefore, the monitoring of students' performance during their studies, especially after the first semester, may prevent the dropout. The features ETmath, AT1, AT2, Exam show the continuous learning progress in mathematics, however, sometimes the tasks of higher mathematics tests are considered in the physical or chemical applications, therefore the enrolment physics and chemistry tests were included to the set of analysed features. Furthermore, their performance is positively correlated with Exam scores (Table 4.6).

Table 4.6: Correlation matrix for features of TPU dataset.

corr. coef.	ETmath	ETph	ETch	AT1	AT2	Exam
ETmath	1.00	0.58	0.43	0.62	0.59	0.47
ETph	0.58	1.00	0.27	0.45	0.47	0.37
ETch	0.43	0.27	1.00	0.38	0.35	0.36
AT1	0.62	0.45	0.38	1.00	0.71	0.62
AT2	0.59	0.47	0.35	0.71	1.00	0.63
Exam	0.47	0.37	0.36	0.62	0.63	1.00

According to the data preprocessing scheme (Fig. 4.1), the next step after forming the set of features is handling missing values. All analysed features represent the scores for the passed tests, where the higher score corresponds to the higher grade. The missing values in the dataset are related to the situation when students have not passed a test. In this study case, we do not need to separate students who tried to pass the exam but obtained 0.0 scores, and students who have not come to pass a test, since consistent with Russian educational system they have equal chances to drop out. Thus, all missing values have been transformed to 0.0.

The enrolments tests and the program of higher mathematics course are unique and obligatory for all students who study their first semester in engineering educational programs. Therefore, the analysed features do not need any transformation over an educational program. The transformation over features was performed by applying standardization.

4.3.3.2 Outlier Detection

The first step of unsupervised outlier detection is to define abnormal values. The distribution of each feature is presented in Fig.4.6. Here,the box corresponds to the IQR, the horizontal line inside the box shows the median, the whiskers denote the $[Q1 - 1.5 * IQR; Q3 + 1.5 * IQR]$ interval, the circle white point shows mean value, and the colored background denotes the $[mean - 3 * std; mean + 3 * std]$ interval. The abnormal values defined as deviations from the mean (less than $mean - 3 * std$ or more than $mean + 3 * std$) have not been found. The observations which fell beyond the whiskers have been found for features AT1 and Exam. Where abnormal values for AT1 correspond to the highest scores, while abnormal values for Exam corresponds the lowest score. This means that the majority of students do not pass the first attestation tests with excellent scores, but they tend to pass the exam.

To investigate the robustness of unsupervised outlier detection algorithms to the change of the parameters, the outlier scores obtained with different parameters have been compared: $10 \geq k \geq 50$ - the number of nearest neighbours for kNN and LOF algorithms, $6 \geq k \geq 11$ the number of clusters for CBLOF algorithm, $30 \geq b \geq 40$ for HBOS algorithm. Fig. 4.7 presents the relations of outlier scores and their ranks. Curves of outlier scores detected by four algorithms with

different values of parameters have the similar declining shapes. We can emphasize the swift drop of outlier scores on the interval 0%- 3%, significant decrease until 5%, and gradual decline on the interval 10% - 100%. Therefore, 3%, 5%, and 10% of students with the highest outlier score have been considered as outliers.

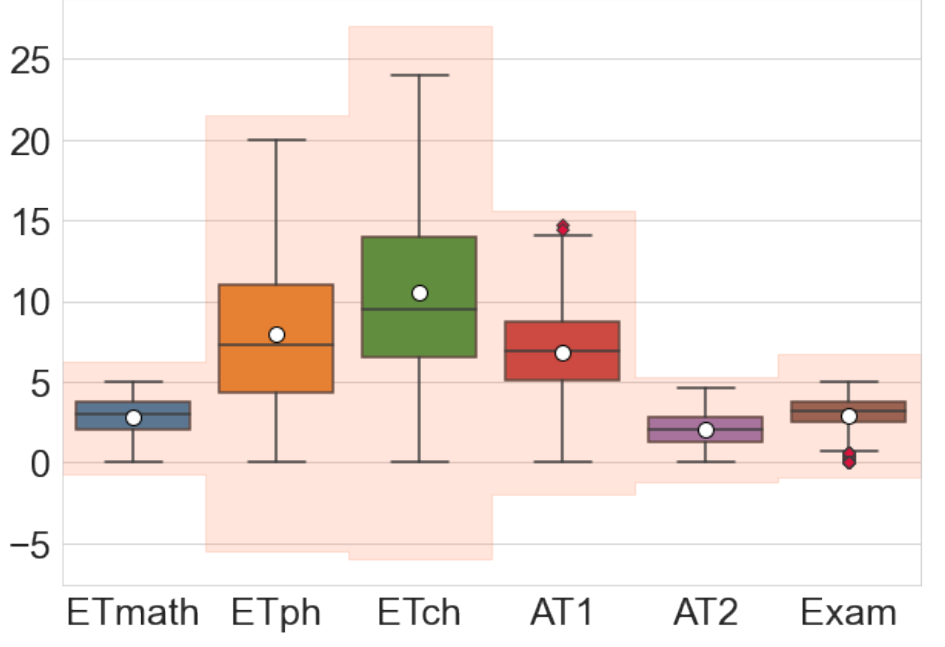


Figure 4.6: Boxplots for features of TPU dataset.

The correlation matrices of outlier scores gained by four outlier detection algorithms (Fig. 4.8) with different parameters show: kNN algorithm is robust to the change of the parameter k-nearest neighbours (corr. coef. ≥ 0.94); HBOS algorithm is robust when the number of bins $b < 40$ (corr.coef. ≥ 0.86), while outlier scores detected with the parameter $b = 40$ are less correlated with others; LOF algorithm is robust when the number of nearest neighbours $k \geq 20$, whereas the outliers scores obtained with $k = 10$ are less correlated with others; the outlier scores estimated by the CBLOF are not robust to the change of the parameter, therefore for further outlier detection the number of clusters has been chosen regarding the elbow curve and the silhouette criteria. Finally, the outliers detected with the following parameters have been considered: $k = 50$ for kNN and LOF, $k = 6$ for CBLOF, and $b = 33$ for HBOS.

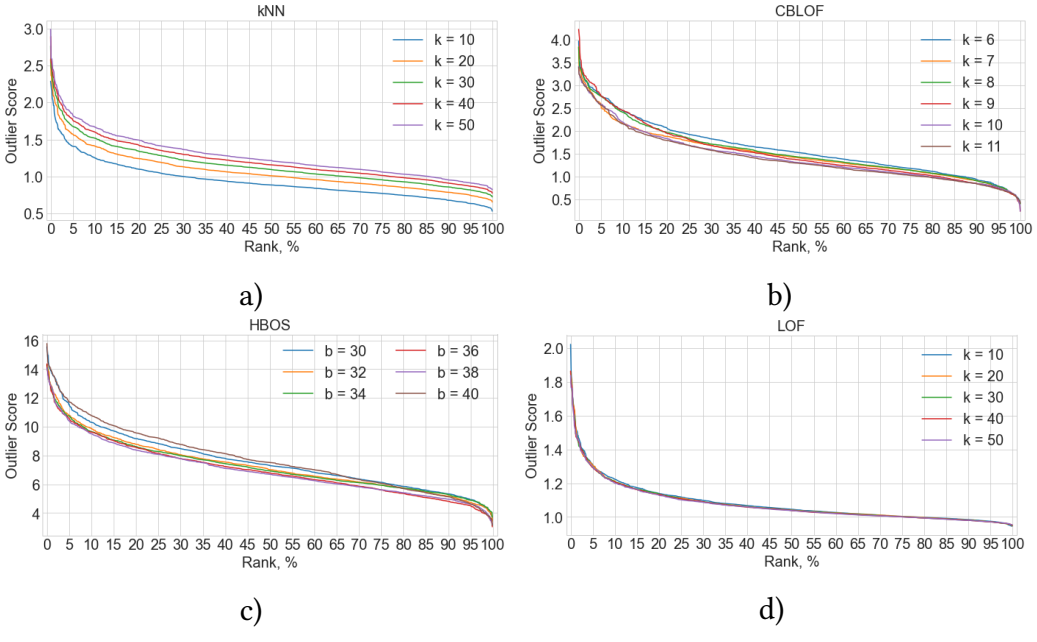


Figure 4.7: Outlier scores detected by unsupervised outlier detection algorithms with different parameters for dataset TPU *versus* rank of outlier score.

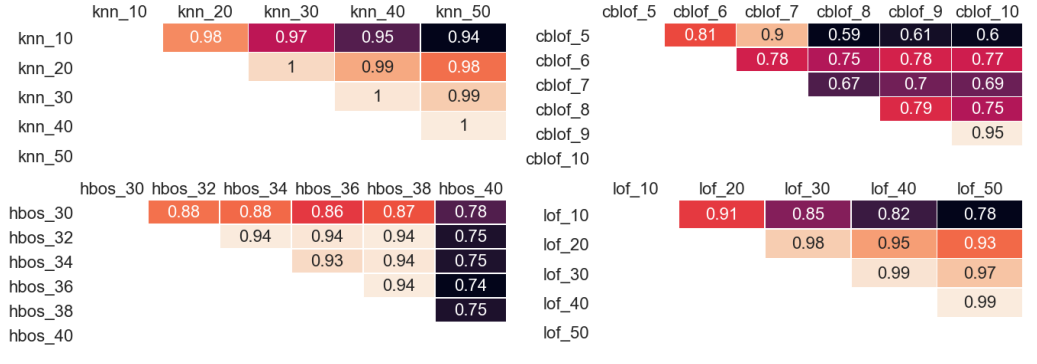


Figure 4.8: Triangular correlation matrices for outlier scores detected by four unsupervised algorithms with different parameter values for TPU dataset.

The intersection rates of outliers identified by four unsupervised algorithms with three assumptions that 3%, 5%, and 10% of students are outliers performed in the Table 4.7. Outliers detected by kNN, CBLOF, and LOF have a high rate of intersection, which increases with the increasing of percentage assumption. Yet, outliers detected by histogram-based HBOS almost do not overlap with outliers detected by other algorithms. This can be explained by the various nature

of HBOS and other algorithms. Despite kNN, LOF, and CBLOF are completely different, all of them use the same distance matrix between objects (Euclidean distance), while HBOS uses the histogram values.

Table 4.7: Triangular matrices with intersection rate for outliers detected by unsupervised algorithms in TPU dataset.

	kNN_3	CBLOF_3	HBOS_3	LOF_3
kNN_3	1.00	0.64	0.00	0.67
CBLOF_3		1.00	0.00	0.42
HBOS_3			1.00	0.00
LOF_3				1.00

	kNN_5	CBLOF_5	HBOS_5	LOF_5
kNN_5	1.00	0.72	0.00	0.78
CBLOF_5		1.00	0.00	0.61
HBOS_5			1.00	0.00
LOF_5				1.00

	kNN_10	CBLOF_10	HBOS_10	LOF_10
kNN_10	1.00	0.79	0.15	0.82
CBLOF_10		1.00	0.10	0.74
HBOS_10			1.00	0.11
LOF_10				1.00

To find out the main tendencies of outlying behaviour of students in the TPU dataset, the detected outliers have been clustered by k-means algorithm, where k - number of clusters has been chosen with the help of the elbow curve and the silhouette criteria. Analogically to the case study with the SA dataset, we focus on the clusters obtained with the 10% assumption since they include outliers from 3% and 5% assumptions. The clusters of outliers detected by the kNN algorithm are presented in the table 4.8. The results of clustering for other algorithms are performed in the Appendix B . The tables contain information about the number of outliers in each cluster (N), the number of outliers that are also detected as outliers in 3% and 5% assumptions (N 3% and N 5% respectively), and mean cluster values for each feature. Although the features for outlier detection and clustering were standardized, the tables include average values for min-max normalized (where values of features $\in [0; 1]$) features of the clusters to help the interpretation. The values of features 0-0.54 mean a fail; 0.55 - 0.69 correspond to the average score, 0.7 - 0.89 indicate good scores, and 0.9-1 determine the excellent scores.

Clusters of outliers detected by kNN:

Cluster 1 - Good course performance with a good background in physics and mathematics. Students with good scores for enrolment tests in physics (0.75) and mathematics (0.76), who passed attestation tests with average scores and exam with good scores (0.85).

Cluster 2 - Average exam score despite failed tests. Students from this cluster have low scores for all enrolment tests and attestation tests. However, after they passed the exam with an average score (0.64).

Cluster 3 - Problems with attestation tests. Students with a high score for all enrolment tests, which passed the exam with an average score (0.63). However, they failed both attestation tests.

Cluster 4 - Failed performance. Students who failed all tests.

Cluster 5 - Failed course performance. Students from this cluster have average scores for enrolment mathematics test (0.65), however, after they failed attestation and examination tests.

Cluster 6 - Good course performance with good background in chemistry and mathematics. Students with excellent scores for chemistry enrolment test (0.97) and good scores for mathematics enrolment tests (0.72) as well as for attestation and exam tests.

Table 4.8: Characteristics of clusters of outliers detected by kNN algorithm with 10% assumption for TPU dataset.

Cluster	**1**	**2**	**3**	**4**	**5**	**6**
N	9	10	11	31	39	8
N 5%	2	3	6	18	22	3
N 3%	1	2	3	12	14	1
ETmath	0.76	0.34	0.75	0.44	0.65	0.72
ETph	0.75	0.14	0.8	0.37	0.44	0.26
ETch	0.18	0.17	0.76	0.39	0.46	0.97
AT1	0.67	0.55	0.51	0.04	0.49	0.64
AT2	0.69	0.49	0.42	0.1	0.22	0.62
Exam	0.85	0.64	0.63	0.33	0.16	0.82

Clusters of outliers detected by CBLOF:

Cluster 1 - Problems with attestation tests. Students with average scores for all enrolment tests and the exam. However, they failed both attestation tests.

Cluster 2 - Failed course performance. Students from this cluster have average scores for enrolment mathematics test (0.67), however, after they failed attestation and examination tests.

Cluster 3 - Failed performance. Students who failed all tests.

Cluster 4 - Average exam score despite failed tests. Students from this cluster have low scores for all enrolment tests and attestation tests. However, after they passed the exam with an average score (0.63).

Cluster 5 - Good course performance with good background in chemistry and mathematics. Students with average scores for chemistry and mathematics enrolment tests as well as for attestation and exam tests.

Clusters of outliers detected by HBOS:

Cluster 1 - Good performance. Students with good scores for all tests.

Cluster 2 - Good course performance with a good background in chemistry and mathematics. Students with good scores for all tests, except ETph, who passed exams with excellent score (0.95).

Cluster 3 - Good course performance with good background in physics and mathematics. Students with good scores for all tests, except ETch, who passed exams with excellent score (0.9).

Clusters 4-5 - Failed performance. Students who failed all tests.

Clusters of outliers detected by LOF:

Cluster 1 - Average exam score despite failed tests. Students from this cluster have low scores for all enrolment tests and AT1, however, after they tried to reestablish and passed AT2 and exam with average scores.

Cluster 2 - Failed performance. Students with failed performance for all tests.

Cluster 3 - Problems with attestation tests. Students with average scores for all enrolment tests and the exam. However, they failed both attestation tests.

Cluster 4 - Good course performance with good background in physics and mathematics. Students with good scores for mathematics and physics enrolment tests as well as for attestation and exam tests.

Cluster 5 - Failed course performance. Students from this cluster have average scores for enrolment mathematics test (0.69), however, after they failed attestation and examination tests.

In this section the students-outliers have been detected for the educational dataset collected in a Russian institution of higher education. The results obtained with 4 unsupervised outlier detection algorithms have been compared. The main tendencies of outlying behaviour have been investigated. The main outcomes of the analysis are as follows:

1. The statistically abnormal values have been found for the following features: AT1 and Exam. These values correspond to the students who passed AT1 tests with the maximal score, and failed Exam. The rareness of excellent scores in AT1 test might be explained by the difficulties in adaptation. When students just enter the university, they need some time to adapt to more complex schedule and subjects. Meanwhile, the majority of students tends to avoid failure of the Exam, since they can lose the scholarship or even be expelled.

2. The students-outliers have been detected by four outlier detection algorithms, which require the defining of the parameters. The analysis of the robustness to the change of the parameter value shows that outliers scores detected by kNN with different parameters are highly correlated, outlier scores detected by LOF with different parameters are highly correlated when $k \geq 20$, while outlier scores detected by HBOS with different parameters are highly correlated when $b < 40$, the outliers scores detected by CBLOF with different parameters correlated less than others.

3. The intersection rate for outliers detected with three assumptions shows the high overlap between algorithms kNN, CBLOF, and LOF. Outliers detected by HBOS algorithm almost do not overlap with outliers detected by other algorithms. This can be explained by the fact, that kNN, CBLOF, and LOF use the same similarity matrix, while HBOS is based on histograms' scores.

4. The clusters of detected outliers present the main tendencies of outlying behaviour. Each algorithm detected students who failed all tests as outliers, or who passed enrolment tests but after failed attestation and exam tests. These students have problems with learning at the beginning of their study, which can probably lead to the drop out of the study. Another example of outlying behaviour show students who passed enrolment tests with good scores, then had problems with attestation tests, but after passed the exam

with average scores. Furthermore, all algorithms detected students with good or even excellent performance for all tests, or for all tests except one enrolment test.

4.4 Outlier detection for sequential data

4.4.1 Case study 3: Serious game for French nurse schools

In this section, the method of preprocessing sequential data is proposed for the serious game in order to find out students-outliers with outlying sequences of actions. The dataset has been collected from the game-based simulation, which is called Clinical Organizer Nurse Education (CLONE). The serious game CLONE is a real-time Digital Virtual Environment for Training. It targets students from Nursing Schools where they have to manage various real-life-like professional situations. The game provides a big number of cases studies where a nurse-student plays the role of a nurse. This digital environment includes game mechanisms and interactive features such as task scheduling or shifting and decision-making (Fig. 4.9). The designing process of CLONE contained three steps: the domain analysis, the human activity modelling, and the scenario, which are described in [121]. Before considering the preprocessing method and outlier detection in detail, we briefly describe the game process in the following section.

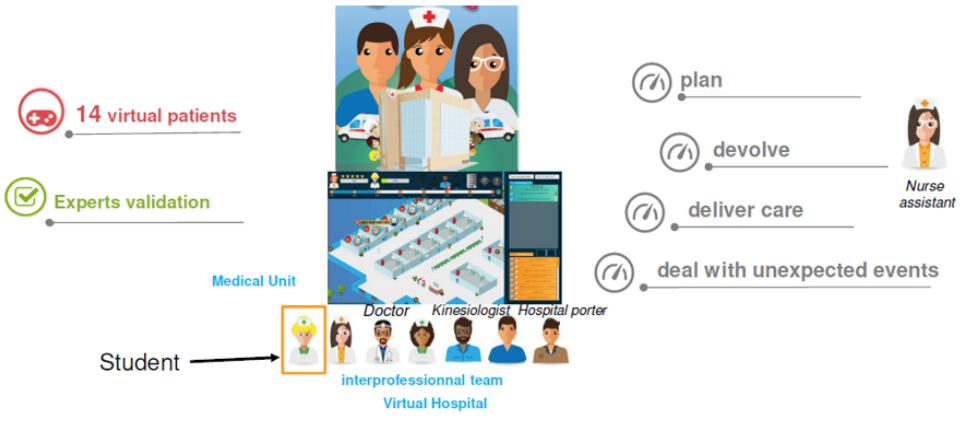


Figure 4.9: The visualization of the main CLONE characteristics.

The player chooses a case study from the library of educational scenarios, which shortly describes the actual and expected situations. A case study provides interactions that allow the players to complete the mission. Each proposed case studies contains patients, actors involved in the medical team and medical dynamic events predictable or not. The mission includes locks (educational and playful), which intend to prevent the player to succeed. The player has to man-

age patients' diseases to deliver care and to delegate tasks to the nurse assistant. To complete the mission, the medical care team (nurse and nurse assistant) has to provide the required care for each patient according to their pathology. Afterwards, outcomes are compared to expected objectives, and results of the game are immediately displayed on the dashboard. Fig. 4.10 presents the graphical user interface.

Figure 4.10: The graphical user interface of the serious game CLONE.

The playing process includes the following steps:

1. **The briefing.** At this step a student reads information about the mission of the game. The game briefly describes the actual situation and expected situation at the end.

2. **The communication with the night shift.** Here, a student receives information from the night shift about the current situation when they shift at 6:30 a.m.

3. **The scheduling.** This step is devoted to developing a care plan for all patients. At this step, a student inspects patients' records and organizes their daily activities.

4. **The care delivery.** The main goal of this step is to provide drugs, organize medical examination, professional phone calls, patient discharge or arrival.

5. **The communication with the afternoon shift and debriefing.** The next shift is informed about the current situation.

A list of soft and hard constraints is attached to each patient. A hard constraint must be satisfied at all costs. For example, distributing an anti-inflammatory treatment according to a doctor's prescription is a hard constraint. A soft constraint refers to a desirable practice that might be violated in order to generate a workable solution. If the player breaks a soft constraint, the patient's health is not strongly affected. If a hard constraint is broken, the player loses a star. A set of stars is associated with each scenario and represents the maximum of violations allowed.

4.4.1.1 Methodology

Data Extraction, Cleaning, and Encoding

The students from 11 French Nursing Schools played the serious game during the 2018-2020 time period. The data collected through playing is kept as a SQL database, which contains information about the actions of students during game sessions and their timestamps, actions' type, the chosen scenario, the personal information of students (school, sex, etc.). Therefore, the first step of data preprocessing is extraction the needed dataset from database (Fig. 4.11).

To compare students' in-game behaviour and detect outlying sequences of actions, the data has been extracted for the same scenario. The explored scenario includes 5 patients' profiles, who require a low-level of care. According to the scenario, students had to attentively analyse patients' profiles, doctor prescriptions, and schedule care plan for all patients. Moreover, the set of established constraints should be satisfied for each patient. In case of breaking of hard constraints, students accumulate critical errors, which can bring to the failure of the game session. The maximal allowed number of critical errors for Patient 1 and Patient 2 is 7, for Patient 3 and Patient 4 is 3, and for Patient 5 is 4. Thus, the collected data contains in-game actions for 353 game sessions which were made by 222 students. Among considered in this study 353 game sessions, 261 are lost, and 92 are successful. The game saves modifications of the schedule panel form previous sessions, therefore repeated sessions can be considered as continuous games. For example, a student may devote the first session to the inspection of patients' records and scheduling, and the second session - to deliver care, which

makes a comparison of sessions incorrect. To avoid biased results, repeated sessions for one student were merged into one. Hereby, for further analysis, we obtained 222 sequences of actions for 222 players. Further, *game session* will denote merged games.

The serious game CLONE contains complex solution paths (open-ended), and allows players to make various actions to achieve the goals. For example, sometimes students can repeat the same actions several times in a row or the sequence of continuous mouse clicking can show the one action. This can evoke difficulties in the task of students' in-game behaviour tracing because it can not properly reflect the real situation. For this reason, duplicate actions have been deleted and in-game actions and sets of actions were encoded.

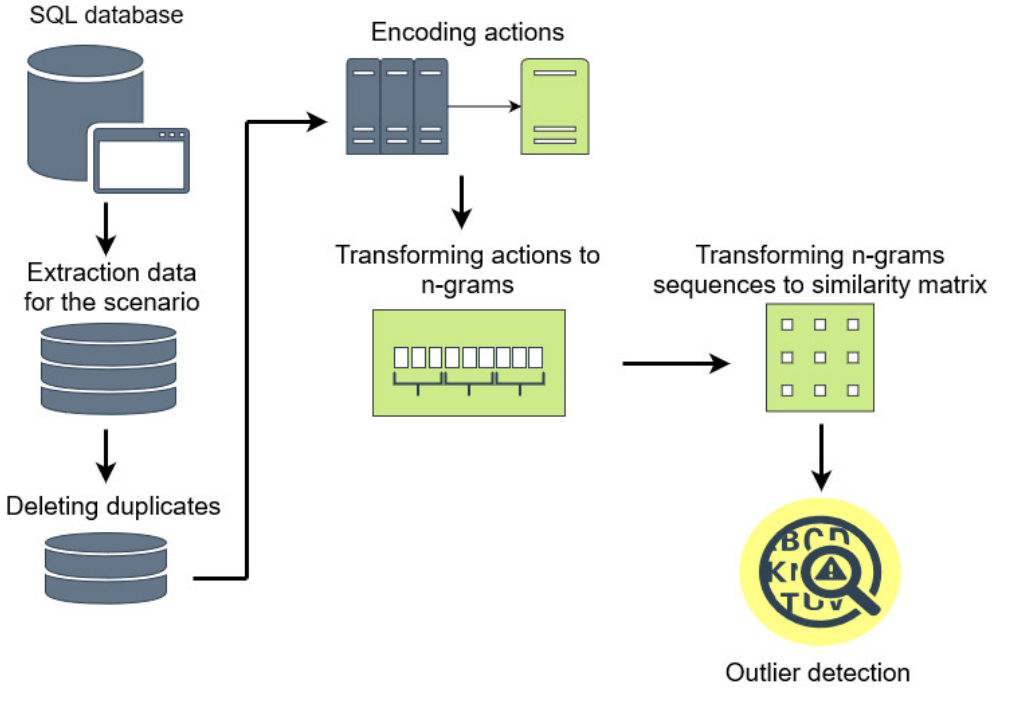


Figure 4.11: Scheme of sequential data preprocessing for outlier detection.

The actions or sets of actions were denoted as activity types and after were coded. The code for each action has the following representation: **Type_Info_Time**, where **Type** corresponds to activity type, **Info** refers to the Patient ID or extended information about activity, and **Time** corresponds to the virtual time at schedule panel. The scheduled panel contains six time gaps with one-hour duration and one time gap with thirty minutes duration, where the

start time is 7:00, and the end time is 13:30, thereby the code for gap 7:00-8:00 is 1, and code for gap 13:00 – 13:30 is 7. Depending on the activity, the code may consist of one, two, and three parts. For instance, the code of such activity as Communication is C, the code of planning of the breakfast Pm_bkf, the code of planning an activity pertaining to personal care for a Patient 4 and 8:00-9:00 time gap is P_4_2. The full list of codes and its descriptions are represented in Table 4.9.

Sequences Analysis

The data extracted from the game represents sequences of actions, which students make during the game session. It is worth mentioning that we do not focus on time series and ignore the time for each action. However, we concentrate on continuous sequences of actions and their row in the game session. In other words, we do not compare data objects according to the time, when they happened, but analyse their order.

For a symbolic sequence, the simplest way is to treat each element as a feature. However, the sequential nature of sequences cannot be captured by this transformation. To keep the order of the elements in a sequence we exploited the n-gram models. N-gram approach is commonly used in computational linguistics (for instance natural language processing) and computational biology (e.g. DNA sequencing or protein sequencing). An n-gram is a contiguous subsequence of n elements from considered sequence. For instance, for a sequence of elements {1, 2, 3, 4} we can form three 2-grams {12, 23, 34} and two 3-grams {123, 234}. Given a set of n-grams, a sequence can be performed as a vector of the presence and the absence of the n-grams or as a vector of the frequencies. These vectors form map matrix, which afterward might be a basis for further analysis such as clustering or classification [122]. At the same time, n-grams can be used for efficient approximate matching [123].

The sets of n-grams were utilized for computing a similarity measure between action sequences. We applied a metric based on Jaccard index, which is ratio Intersection over Union.

$$J(A, B) = A \cap B / A \cup B \tag{4.1}$$

Here the numerator is the number of n-grams intersection and the denominator

Table 4.9: Types of actions in a serious game playing process.

Type	Description	Code
Care action	Deliver care to a patient	A_1/.../A_5
Related activity	Perform tasks related to the patient's care	Af_1/.../Af_5
Relaxation	Perform actions in the relaxation room	R
Delegate tasks	To a nurse assistant To a nurse To a nurse and nurse assistant	D_AS D_IDE D_IDE-AS
Planning Micro	Set / modify / delete a task in the patient's care plan	P_1_1/ .../P_5_7
Planning Macro	Move/ delete a task in the global care plan	Pm_1_1/ .../Pm_5_7
Arguments	Argue for the choice	Ar_1/.../Ar_5
Open patients' records	Open patient's administrative record and click on the mask to discover hidden information Open patient's medical record and click on the mask to discover hidden information Open patient's nursing record and click on the mask to discover hidden information	O_adm O_med O_nurse
Communication	Communication with the next or previous shift team	C
Help	Get information about the nursing practices in this medical unit	H
Inspection of global planning	Inspect scheduled global planning without making any actions	PI
Inspection	Inspect a patient's care planning without making any actions for a patient Inspect administrative record Inspect nurse assistant record Inspect all information Inspect global care plan for all patients Inspect medical reports Inspect nurse service and nurse assistant service Inspect nurse service Inspect prescriptions for injections Inspect prescriptions per os Inspect other requirements Inspect other information	I_1/.../I_5 I_adm I_assist I_info I_all I_med-rep I_n-as I_nurse I_pr-inj I_pr-os I_pr I_service

is the number of n-grams union. The Jaccard index ranges from 0 to 1, where 1 – means 100% similarity and 0 – means 0% similarity. Therefore, distance based on Jaccard index for two sets of n-grams $s1, s2$ can be represented as follows:

$$dist(s1, s2) = 1 - J(s1, s2)$$

Outlier Detection

In the context of analysed data, we assume that the games where students had abnormal and rare sequences of actions or students with rare behavioural game strategy are called outliers and the process of finding them is outlier detection. Among the data science techniques for the exploration of behavioural strategies, cluster techniques are widely used. Cluster analysis helps researchers to group data based on the similarity of the data points.

There are numerous options for clustering methods, but all attempt to merge similar objects in the same cluster, and split dissimilar objects into different clusters. For instance, authors in [106] applied fuzzy feature cluster analysis to identify key features of student performance in log data collected from a mathematical game for sixth grade students; [108] used Expectation-Maximization and K-means clustering approaches for students' performance in a game simulation of the biological room; [124] performed hierarchical clustering in order to investigate student success study of fractions.

Thus, we implemented a clustering method with a distance matrix based on the Jaccard index in order to find out groups of students with similar action sequences and reveal their characteristics. Our main assumption was founded on the presumption that clusters with a small number of members contain outliers and their characteristics are abnormal. Since we do not use any predefined parameters except distance matrix, we do not utilize the partitioning methods of clustering. Eventually, we applied agglomerative clustering as a linkage method, namely Ward's method, which is based on minimizing variance. Afterward, to confirm the assumption about small clusters, we considered distance-based outlier detection algorithm k-nearest-neighbours (kNN) [73], and compared the detected outliers.

4.4.1.2 Results

To examine the students' activity during the game, 222 sequences of actions were parsed into n-grams. We examined three models where parameter n = [4,5,6]. The models, where $n < 4$ were not considered because they can not show the main tendencies in game strategies of students. The game does not have any restrictions in terms of continuous progress, which means students can commit any action at any time and repeat it several times. For instance, students can plan an activity related to care for Patient 1 for 7:00-8:00 time gap, then for 10:00 – 11:00 time gap, and afterward modify 7:00-8:00 time gap. 2-grams and 3-grams models contain a lot of noisy patterns, which have high frequency but do not reflect any behavioural features.

In selecting an appropriate model for further analysis, we focused on the comparison of the following model's characteristics: number of unique grams and relative frequency. The Relative Frequency RF for the gram G_i is a ratio of the gram's frequency $N(G_i)$ (how many times the gram appears in data) and the number of existing grams in the data N:

$$RF(G_i) = N(G_i) \times 100\%/N \tag{4.2}$$

With the increasing n parameter of the model, the number of unique grams rapidly growing due to the expanding the number of possible combinations of actions (Table 4.10). Consequently, the part of grams which occur in data once is increasing. Therefore, for further analysis, the 4-grams model was chosen, due to the optimal values of considered characteristics.

Table 4.10: Characteristics of n-gram models.

The model	**4-grams**	**5-grams**	**6-grams**
N of unique grams	22 041	32 126	40 395
RF of 100 most frequent grams	21%	14%	9%
RF for grams occurred in data once	26%	42%	56%

Patterns consisting of 4 actions can determine tendencies in players' behaviour during the game session. For example, the 4-gram {P_1_2,P_1_3,P_1_4,P_1_5} shows continuous planning strategy and the 4-gram {I_1,P_1_1,I_1,P_1_2} refers planning with inspection. Consistent with analysis of obtained frequent grams,

the evident behavioural strategies in game sessions procedure were highlighted (Table 4.11). In particular, the distinguished strategies concern the following activities: scheduling personalized care plan (S_P), delivering care (S_A), opening medical records (S_O), and inspection (S_I). Moreover, general strategies pertaining to inspection, opening medical records, and scheduling, were specified consistent with patient index. Patterns devoted to delivering care are merged into union strategy S_A, due to the small values of RFs for patterns related to a certain patient.

Table 4.11: Behavioural strategies in the playing process.

Strategy	**Description**	**Examples**
S_P_1, S_P_2, S_P_3, S_P_4, S_P_5	Planning a schedule for the certain patient	{P_1_2,P_1_3,P_1_4,P_1_5} {P_4_3,P_4_4,P_4_5,P_4_6} {P_5_1,P_5_2,P_5_1 P_5_2}
S_A	Delivering cares for patients	{A_1,A_2,A_4,A_5}
S_O_1, S_O_2, S_O_3, S_O_4, S_O_5	Opening medical reports and planning a schedule for the certain patient	{O_adm,O_med, P_1_2,P_1_3} {O_nurse,P_5_2,O_med,P_5_3}
S_O	Continuous opening medical reports	{O_adm,O_med,O_nurse,I_info}
S_I_1, S_I_2, S_I_3, S_I_4, S_I_5	Inspection and planning a schedule for the certain patient	{I_all,I_1,P_1_1,I_all} {I_2,P_2_1,P_2_2,P_2_3} {I_all,I_5,I_info,P_5_1} {I_2,P_2_1,I_2,P_2_2}
S_I	Inspection of additional information	{I_pr-inj,I_pr,I_med-rep,I_n-as} {I_pr-os,I_pr-inj,I_pr,I_n-as}

According to analysis of patterns' frequency, the majority of patterns are devoted to the planning of schedule for patients. The scenario conditions imply focusing more on one group of patients and less on others. This strongly depends on the number of hard constraints for each patient and their discharge or arrival (according to the scenario, Patient 3 has late arrival). The distribution of patterns among strategies of planning, as well as strategies of inspection

and reading medical reports have the same characteristics: the highest RFs correspond to Patient 1 and Patient 2 and the lowest – to Patient 3 (Fig.4.12). The figure concerns the most frequent 1281 4-grams with RF>0.01%. Here strategies related to inspection, opening medical records, and scheduling are specified for each patient and represented like **S_Action_n**, where n is the index of the patient. WS denotes patterns without strategy.

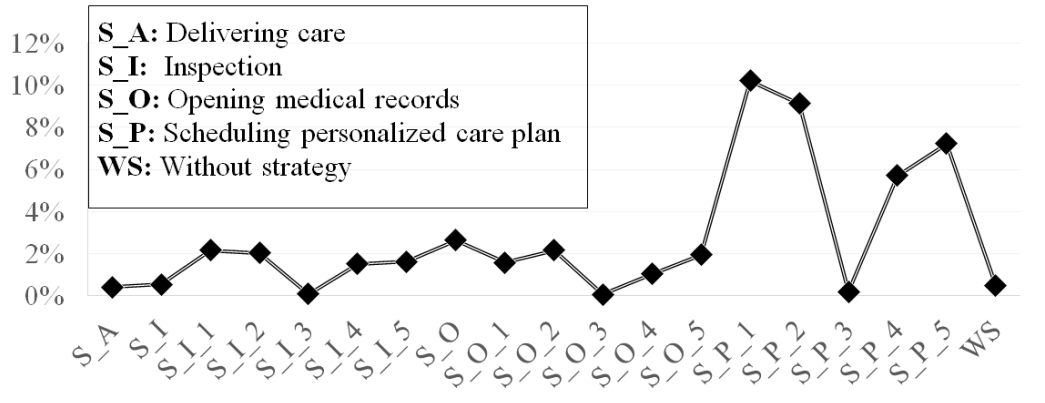


Figure 4.12: Cumulative Relative Frequency of 4-grams according to distinguished strategies.

Table 4.12: General characteristics for clusters.

Cluster	N	Min	Average	Max
C1	67	109	302	467
C2	18	1	61	313
C3	26	77	227	386
C4	111	52	307	708

In order to emphasize the similarities between sequences of actions, which students made during a game session, we exploited a clustering. Agglomerative hierarchical clustering (Ward's method) was implemented by using computed earlier Jaccard distance matrix for sequences of 4-grams. The hierarchical dendrogram divided sessions into four major clusters. The main characteristics of clusters are presented in Table 4.12, where N denotes the number of game sessions in the cluster, Min and Max reflect the minimal and maximal length of 4-gram sequence in a cluster, Average determines the average number of 4-

grams per session in a cluster. The biggest cluster C4 contains 111 game sessions, whereas the smallest includes 18 game sessions.

To investigate prevailing strategies for clusters, we considered the distribution of patterns occurring in every cluster depending on their strategies (Fig. 4.13a). Distributions of clusters C1, C3, and C4 have similar shapes: patterns mostly distributed between planning, inspection, and reading medical records strategies. Meanwhile, the major part of patterns from cluster C2 has either an inspection strategy either does not have any strategy. More detailed distributions of patterns according to their strategies and clusters are represented in the Fig.4.13b (patterns without strategy are excluded). Here we also can see the dependence between the number of hard constraints and the frequency: the most frequent patterns correspond to Patient 1 and Patient 2, whereas the less frequent patterns correspond to Patient 3.

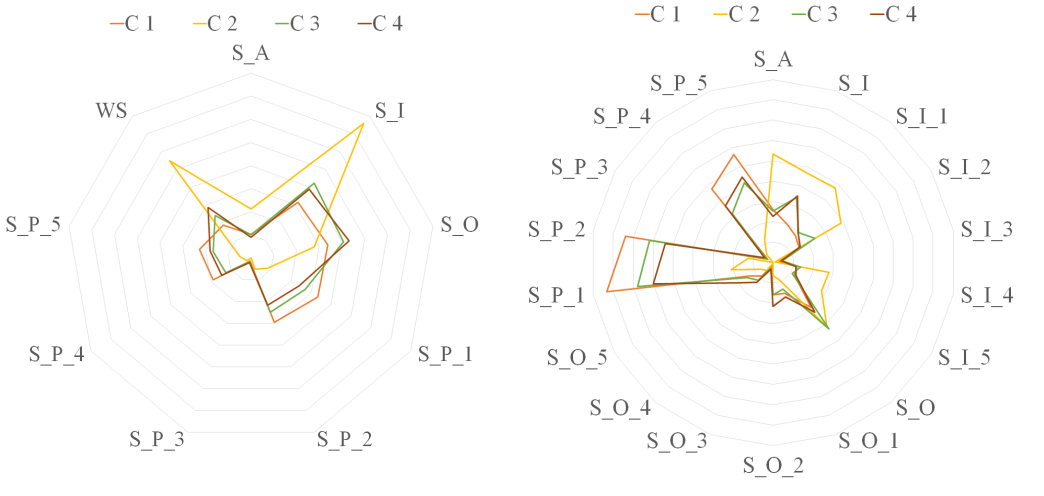


Figure 4.13: Distribution of 4-grams by clusters. a) Distribution of patterns according to their strategies. Here WS denotes RF of patterns without strategies. b) Detailed distribution of patterns according to their strategies. Patterns without strategy are excluded.

Cluster C2 notably contrasts with other clusters in terms of patterns distribution by strategies. That means there are sessions, where players' behavior differs from the general behavior of the majority of players. To investigate the nature and significance of these differences, we used unsupervised out-

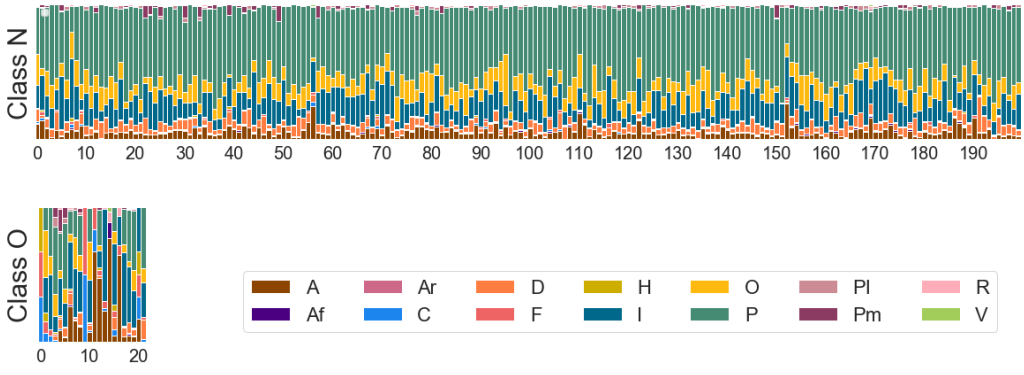


Figure 4.14: Distribution of types of actions according to the class, where Class N contains inlier session, and Class O contains outlier sessions.

lier detection. 22 game sessions were detected as outliers by applying the kNN method with parameter $k = 10$ and assumption that 10% of students are outliers ($222 * 10\% = 22$). Among them, 4 sessions are from cluster C4 and 18 sessions are from cluster C2. Thus, the entire cluster C2 is outlined.

The distribution of action types for both classes of sessions - Inliers (Class N) and Outliers (Class O) is depicted in Fig. 4.14. Game sessions for Class N have the homogeneous spreading of actions, which determine the main behavioral tendencies for the majority of players. These game sessions are mostly devoted to planning schedules, reading medical records, and information inspection. During these game sessions, players use the delegation of tasks and deliver care. Contrary to this, sessions-outliers do not have any evident tendencies. However, we can emphasize several reasons for their arising:

1. **Delivering cares before planning.** Among detected outliers, there are games where students do not plan a personalized schedule for each patient before delivering care. As a result of this activity, a student makes critical errors that lead to failure.

2. **Inconsistent planning.** Elaborating a global care plan is one of the most important goals of the game. The majority of students try to consistently plan the daily care for one patient: inspecting information, reading medical records, setting or modifying personalized care, and afterward, they do the same actions for another patient. Furthermore, they effort to continuously move from the first time gap to the last. This group of detected outliers is

characterized by inconsistent planning when students planned schedule just for one patient or they incoherently moved from one time gap to another.

Inconsequent planning interferes students to achieve the goal of the scenario, however, sometimes this tactic leads to success. Among detected outliers, there is one successful session, where the student often interrupted the inconsistent planning by inspection or delegation, that did not break hard constraints.

4.5 Conclusion

This chapter is devoted to the investigation of outlier detection in educational data. Two approaches for educational data prepossessing have been proposed: (i) for numerical data; (ii) for sequential data. Then, the students-outliers have been detected by unsupervised outlier detection algorithms and their outlying characteristics have been compared.

Outlier detection for numerical data.

Concerning outlier detection in numerical data, two cases were studied: (i) students-outliers in the data of two semesters performance in the German university (SA dataset); (ii) students-outliers in the data of the high mathematics course performance during their studies at the first semester in the Russian university (TPU dataset). First of all, statistically abnormal values were investigated. For both cases, abnormal values correspond to the extremely high or low points for certain features. In the SA case, these features are related to the number of failed, enrolled, passed in advance/delay courses. This means that the majority of students prefer to follow the curriculum and try to avoid fails and enrolments without passing exams. In the TPU case, the features with abnormal values are related to the high scores for the first attestation test, and low scores for the exam. The rareness of excellent scores in AT1 test might be explained by the difficulties in adaptation. Meanwhile, the rareness of low exam scores explains the desire to avoid the loss of the scholarship or being expelled.

Then outliers have been detected in terms of the considered set of features. Since the absence of the ground truth, the state-of-the-art unsupervised outlier detection algorithms have been implemented. Each chosen algorithm presents the different type of outlier detection techniques: distance-based kNN, cluster-based CBLOF, histogram-based HBOS, and density-based LOF. The algorithms

were compared in terms of the robustness to the change of the parameters and intersections of the outcomes. The change of the parameter k in kNN does not dramatically change in both study cases. According to Goldstein *et al.* [28] the parameter k should be more than 10 and less than 50. Spearman's correlation coefficients between outlier scores obtained with k $\in [10; 50]$ are more than 0.94 for both cases. For anomaly detection with CBLOF based on k-means, the setting of k does not pay too big role. If k is overestimated, more than one centroid share an actual cluster. This leads to slightly incorrect scores within the cluster, but the scores of the anomalies stay approximately the same [125]. This is approved by outlier detection in SA dataset, while for TPU dataset the outliers score detected with different k are not highly correlated. Thus, the choice of the parameter k was mainly determined by the elbow curve and the silhouette criteria. HBOS algorithm is robust to the change of the parameter, however, the identified outliers almost do not overlap with the outliers found by other algorithms. This can be explained by the fact, that kNN, CBLOF, and LOF are based on the same distance matrix with Euclidean metric, while the HBOS is focused on the histogram values. The LOF algorithm for both cases is more robust when $k \geq 10$ which corresponds to the [69] statement. Despite LOF, kNN, and CBLOF algorithms are based on the same distance matrix, the outliers found by LOF intersect less than outliers detected by kNN and CBLOF. This can be interpreted by the fact that, unlike CBLOF and kNN which detect outliers as objects which significantly differ from all objects, LOF identifies outliers as objects which significantly differ from their local neighbourhood.

The outliers have been detected by four unsupervised algorithms (kNN, CBLOF, HBOS, and LOF) with three assumptions (3%, 5%, or 10% of students are outliers) for two study cases. Despite the datasets have a similar form $Score \times Question$, the outlying characteristics of students are fundamentally different. As was stated earlier, the definition of an outlier strongly depends on the context. Thus the characteristics of the detected outliers are related to the considered features in each situation: (i) two semesters performance in the German university, (ii) one course performance in the Russian university. However, we can generalize some tendencies of outlier behaviour in students' performance data:

1. Outliers that are characterized by an abnormally high or low value of the certain features. For instance, clusters of outliers with a high number of

passed exams in advance for SA data, or students with problems in attestation tests for TPU data.

2. The improvement/degradation of performance. For example, outliers that are characterized by procrastination in S1 for SA data, or outliers who passed well enrolments test and after failed attestation tests in TPU data.

Outlier detection for sequential data.

The second part of this chapter is devoted to detection students-outliers in sequential data, namely, the students-outliers in the process of playing the serious game in French nurse schools. The detection of outlying patterns is important topic which recently has been emphasized by the research community [117]. However, the developed algorithms are mostly used for time series or transnational data. In the case, when the order of the events has to be kept, the data preprocessing is needed. Thus, the method of sequential data preprocessing has been proposed and applied.

The examining frequent action patterns revealed the main in-game behavioural strategies, which students stand by. The majority of students stand by the same in-game strategy, which implies being aware of the patient's pathology, doctors' prescriptions, and level of required care. The majority of frequent patterns are devoted to scheduling and inspection. In real life this situation is usually the opposite: nurses spend most of the time delivering care and abstain the scheduling and checking the workload. As a consequence, the patient-to-nurse ratio is growing which leads to professional dissatisfaction or burns out [126, 127]. The obtained outcomes showed that the game impacts students' awareness and highlights the importance of scheduling and work organization.

The outliers were detected with two algorithms (Clustering and kNN). The outcomes of these two algorithms are strongly overlapped. Within outlier detection, we exposed the main reasons for arising outliers: delivering care before planning and inconsistent planning. Outliers can occur due to a bad/wrong understanding of the game processor's unawareness of good practice guidelines. The game allows players to override the good practice, for example, minimize the scheduling step and jump to the delivering step. The outliers pointed out the students, who were more engaged in the gaming process and not in the learning process. They do clicks and try to progress without using previous courses, knowledge, and skills.

Chapter 5

The impact of outliers on prediction models

The consequences of our actions are always so complicated, so diverse, that predicting the future is a very difficult business indeed.
— J.K. Rowling

5.1 Introduction

Student performance analysis and performance prediction are two widely explored research topics in the education field [128]. Although they have different objectives, they both seek to design effective mechanisms that improve academic results, enhance the learning process, and avoid dropout. The necessity of developing these mechanisms is caused by social and economic aspects. For example, with the increasing commercialized education environment, education institutions need to become more efficient, provide a better quality service to deliver exceptional student experience [129]. Another illustration is the increasing dropout rate, which reduces the number of people with higher education and wastes the resources spent on financing students who do not complete their studies [11]. Thus, the development of models which predict student performance might help to detect students with problems in learning at an early stage and produce specific advice for them.

In the literature, the performance prediction models usually have two types: the prediction of dropout (or attrition) and the prediction of the results (e.g. examination score or GPA). The prediction of dropouts is always the task of classification, where models have two classes - dropout and graduate. In the case of results prediction, there are more options: score as value, score as a binary class (Pass/Failed labels), score as multi-class (e.g. letter grades). Here, the class

prediction is the classification task, while the value prediction is the regression task.

To produce better results, existing studies about prediction examine various features (e.g. sociodemographic or performance), machine learning algorithms models (e.g. classification, regression or clustering), and study cases (e.g. online courses or specific courses). For instance, Aulck *et al.* [11] predict graduation and re-enrolment of students in a US university. The authors use data of students after the first calendar year and achieve the best results with logistic regression. Furthermore, this study shows that including sociodemographic features in the prediction model hardly improves the results. Berens *et al.* [12] develop an early detection system to predict students' dropout in German universities. AdaBoost is implemented to improve the results of logistic regression, random forest, and neural networks. Baneres *et al.* [119] apply naive Bayes, decision trees, k-Nearest Neighbours, and support vector machines to performance features of students from a fully online university in order to identify at-risk students as soon as possible. Sandoval *et al.* [130] use background and academic records of students to predict a final score with implementation of linear regression, robust linear regression and random forest algorithms. In [131] authors apply principal component regression to features related to internal assessment and video viewing to predict final academic performance.

Classical models usually work for the majority of students with common and consistent characteristics. However, they ignore the students who are not aligned with the majority - outliers. Students-outliers may show significant information to domain experts and affect the prediction models. The effect of the impact of outliers on the prediction models has been investigated in different domains. For example, in [132] authors predict the residential building energy use. They proved that removing outliers detected by the standard deviation and quartile methods improves the performance in 20% of homes with outliers. In [133] authors explore the effects of handling outliers on the performance of bankruptcy prediction models. In [134] authors propose a training algorithm to detect outliers and reduce their negative impact for financial time series prediction. In the educational field, the impact of outliers on prediction models is not well researched. Furthermore, the existing papers devoted to prediction in education barely mention how they deal with outliers. To fill this gap, in this chapter the impact of outliers on prediction models has been investigated.

The chapter performs the analysis of the impact of outliers on two types of models: (i) dropout prediction (ii) final score prediction. For each case we compare outcomes of the models trained with and without outliers, detected in the sections 4.3.2 and 4.3.3. The remainder of the chapter is as follows: Section 5.2 observes the main outlines of machine learning; Section 5.3 performs the analysis of the impact of outliers on the dropout prediction models; Section 5.4 investigates the impact of outliers on the score prediction models.

5.2 Machine learning outlines

The prediction of the performance is the task of supervised learning, which is commonly used by experts in educational domain. The methods of supervised learning vary depending on the type of target variable that needs to be predicted. The model that aims to predict the class of the variable is the task of classification. The model which targets numeric values is the task of regression. As was stated earlier, the prediction of performance in education usually has two types: dropout (attrition) prediction and results prediction. The dropout prediction model is the task of the classification, since the target variable is presented in binary class (Dropout/Graduate). In the case of prediction numerical final scores regression is required.

The main idea of each supervised learning algorithm is to train the model for existing data which will make accurate predictions for new, unseen before, data with the same characteristics as train data. The model learns to make predictions based on this training data, so the more training data the model has access to, the better it gets at making predictions. With training data, the outcome is already known. The predictions from the model and known outcomes are compared, and the model's parameters are changed until they align.

If a model able to adapt properly to new, previously unseen data, drawn from the same distribution as the train data, it has a good generalization. The goal of training is to develop the model's ability to successfully generalize. If a model trained too well on training data, it will be unable to generalize. It will make inaccurate predictions for given new data. This is called overfitting. The inverse situation is called underfitting. Underfitting happens when a model has not been trained enough on the data. In the case of underfitting, it makes the model not

capable of making accurate predictions, even with the training data. The possible solutions which prevent overfitting and underfitting are presented in the Table 5.1.

Table 5.1: The possible solutions to prevent overfitting and underfitting.

Overfitting	**Underfitting**
- To simplify the model by selecting one with fewer parameters - To gather more training data - To reduce the noise in the training data	- Selecting a more powerful model, with more parameters - Feeding better features to the learning algorithm - Reducing the constraints on the model (e.g. reducing the regularization hyper-parameter)

To understand how model generalizes to new data the best way is to split data on train and test sets, where model is trained on the train data and after being tested on the test data. It is common to use 80% of the data for training and keep 20% for the test [135]. The evaluating of the model implies just using test set. However, the possible problem that can arise in this case is that we adapt the model for the particular test set, and if we change it, we might have inaccurate predictions. The best and common solution for this problem is to implement cross-validation. The cross-validation consists of splitting train set into random k folds, and using $(k-1)$ folds to train the model, predicting the remaining k^{th} fold (which is considered 'new' data), and calculating the prediction error metrics. This process is repeated k times to make predictions for each fold (by changing the training and prediction folds).

The evaluation of a model's outcomes depends on the supervised learning task: classification or regression, let us consider them in detail.

Classification performance measures

The classifier evaluation contains several important performance metrics which are based on the confusion matrix values (Fig. 5.1). The confusion matrix shows the number of times when class Positive classified as class Negative, and vice versa.

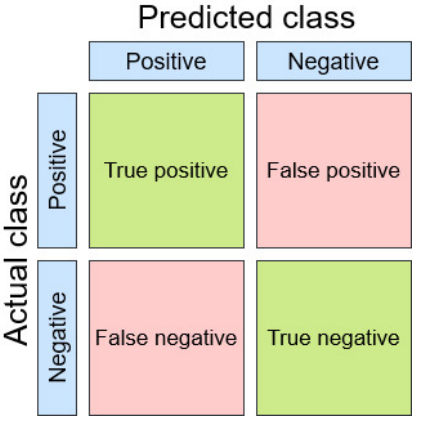


Figure 5.1: The confusion matrix.

The metrics which are usually used for evaluating binary classification models are recall, precision, accuracy, and area under the receiver operating characteristic curve, which are estimated as follows:

$$Recall = \frac{TP}{TP + FN} \tag{5.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{5.3}$$

The area under the receiver operating characteristic curve (auc roc) is created by plotting the true positive rate ($TPR = recall$) against the false positive rate ($FPR = 1 - TPR$) at various threshold settings.

The evaluating the classifier model is always tricky task, where the most important metric should be chosen consistent with the train data nature and the task of the prediction. For example, accuracy is not preferred performance measure when the data is skewed (when the number of objects with one label is notably higher (lower) than the number of objects with other labels). Furthermore, in some contexts you mostly care about precision, and in other contexts you really care about recall. In the case of dropout prediction, the most preferable metric is recall which shows the rate of correctly predicted dropouts.

Regression performance measures

Regression refers to the predicting a numerical value. To evaluate such models, metrics specifically designed for this task must be used.

Mean absolute error (MAE), which is a measure of errors between paired observations actual value and predicted value.

$$MAE = \frac{\sum_{i=1}^{N} |y_{act}^{(i)} - y_{pred}^{(i)}|}{N} \tag{5.4}$$

Root mean square error (RMSE), which gives an idea of how much error the system typically makes in its predictions, with a higher weight for large errors.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_{act}^{(i)} - y_{pred}^{(i)})^2} \tag{5.5}$$

A perfect errors value is 0.0, which means that all predictions matched the expected values exactly. Yet, this almost never happens in a real-life scenario.

Prediction models

Nowadays, a wide variety of supervised learning algorithms exists. In this subsection, we briefly discuss the supervised learning algorithms, which have been used in this chapter, and give the references, where they distinctly described.

Linear regression.

The linear regression model is probably the most commonly used and famous prediction model, which is described in many references [136]. It is used to predict the numerical values. Linear regression builds the model by computing the weighted sum of the input features plus a constant called intercept term. During the training, the model tries to find the optimal values of the parameters which minimize the cost function (error).

Logistic regression

Logistic regression is a commonly used algorithm to estimate the probability that an instance belongs to a particular class. If the estimated probability is greater than 50%, then the model predicts that the instance belongs to positive class or else it predicts that it does not. Like a linear regression logistic regression computes the weighted sum of the input features, but instead of outputting the

result as number, it outputs the logistic of this result [135]. The logistic regression is used for classification task.

Lasso regression.

Lasso regression is a regularized version of linear regression. It adds the regularization term to the cost function (l_1 norm of the weight vector). This forces the learning algorithm to not only fit the data but also keep the model weights as small as possible [135].

Support vector regression

Support Vector Regression gives the flexibility to define how much error is acceptable in the model and finds an appropriate line (or hyperplane) to fit the data. It seeks to find a hyperplane in an n-dimensional space that distinctly classifies the data points. The data points on either side of the hyperplane that are closest to the hyperplane are called Support Vectors. In contrast to linear regression which aims to minimize the squared error, SVR tries to minimize the l2-norm of the coefficient vector. The error term is set less than or equal to a specified margin, called the maximum error, ϵ (epsilon). Epsilon can be tuned to gain the desired accuracy of the model [137].

Decision tree classifier

Decision trees are built of a set of internal nodes where each node is labelled with an input feature. The output of each node represents a test resulting of the branch based on a certain threshold value. Actual branch feature and the threshold value are computed using an optimization procedure. Individual branches represent the outcome and lead to child nodes with subsequent tests, a target class label in the case of a leaf node [135].

Random forest

Random forest is an ensemble of decision trees. This method involves random feature selection for building individual and different trees. The random forest algorithm can be implemented for both classification and regression tasks. The final result is computed using an aggregating (voting) scheme in case of classification and averaging over the members for regression problems [138].

AdaBoost

AdaBoost is a boosting method. The boosting refers to the ensemble method which combines several weak learners into strong learner. AdaBoost, firstly trains a base classifier (e.g. Decision Tree) to make predictions on the training set. Then the relative weight of misclassified training instances is increased. A

second classifier is trained using the updated weights and again it makes predictions on the training set, weights are updated, and so on [139].

5.3 Impact of outliers on dropout prediction

5.3.1 Methodology

To investigate the impact of outliers on the dropout prediction models we have considered dataset SA since it has labels for each student (Graduate and Dropout). Using the data from the three undergraduate programs, we built cross-program models, which predict dropout, where independent variables are features (Fig. 4.2a) and the dependent variable is the feature with Dropout/Graduate labels. In order to avoid overfitting, methods of feature selection has been applied. Chi-squared stats show the significant dependence ($p < 0.05$) between each independent feature and class feature, except the feature related to the proportion of passed exams with delay in the second semester (P_ex_d S2). However, an extra tree classifier for extracting the importance of features shows that the importance score for P_ex_d S2 is higher than for the proportion of passed exams in advance in the first semester (P_ex_a S1). The removing of features from the train set does not significantly change the prediction performance, so we left all eleven features.

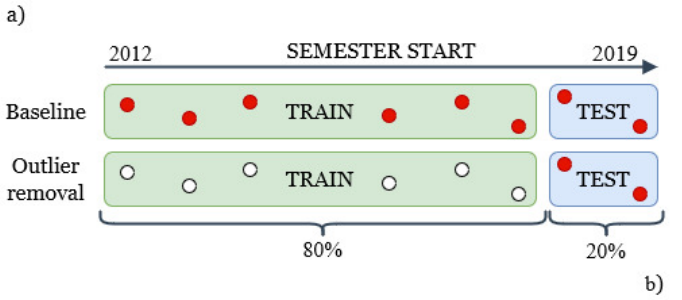
The models based on two different approaches to handle the outliers detected in the section 4.3.2: a. keep outliers, b. remove outliers from training data. The prediction models are based on different types of prediction algorithms: logistic regression (LR) and decision tree (DT) to train interpretable models and random forest (RF) and AdaBoost (AB) as ensemble methods that usually perform well. The analysis has been implemented using the Python scikit-learn library [140].

The dataset that keeps the outliers in the training data (approach a) served as the baseline to evaluate the outlier handling approaches. For approach b, regarding each outlier detection algorithm (kNN, CBLOF, HBOS, LOF) and assumption (3%, 5%, 10%) outliers have been removed from train set.

Analogically to [141] time-aware nested cross-validation has been used to build the models to predict drop out: firstly, we sorted the students' data by their program start semester, then split without shuffling into training and test disjoint sets (80/20%), so that the test set contains the most recent students. The

hyper-parameter tuning has been implemented for each prediction algorithm for baseline model. The 10-fold cross-validation has been used for all training sets to define the best model in each case to predict dropout. The computational details for dropout prediction models are presented in Appendix C Table C.1. The test set remained the same for each model and contained both inliers and outliers since outliers cannot be excluded from the prediction in a real-life scenario.

The detailed outlines of dropout prediction are depicted in Fig.5.1. The Fig. 5.1a presents the scheme of the dropout prediction, where baseline case corresponds to the approach a. keep outliers in the training data, and outlier removal case refers to the approach b. remove outliers from training data. The Fig 5.1b shows the confusion matrix for the task of dropout prediction, where correctly predicted dropouts form true positive class (TP), correctly predicted graduates form true negative class (TN), wrongly predicted dropouts form false positive class (FP), and wrongly predicted graduates form false negative class (FN).



b)

		Actual Class	
		Dropout	Graduate
Predicted Class	Dropout	True positive TP	False positive FP
	Graduate	False negative FN	True negative TN

Figure 5.2: Dropout prediction outlines: a) time-aware train/test splits b) the confusion matrix for dropout prediction.

It is noteworthy to mention that in the dropout prediction task the recall metric is the most preferable, since it returns a ratio of correctly predicted dropouts and the actual dropouts. Thus, the higher the recall score, the more dropouts are correctly predicted. The precision value returns a ratio of correctly predicted dropouts and all dropouts predicted by the model (including wrongly predicted dropouts). The accuracy returns the percentage of correctly predicted labels.

5.3.2 Results

The results of dropout prediction models are presented in Table 5.2. The rows correspond to each considered dataset, where dataset 1 is the baseline model, when all data has been kept for training. Datasets 2 to 13 are related to models without outliers (w/o) in the training set. The columns of the table correspond to the prediction metrics and set size for each prediction algorithm (logistic regression (LR), decision tree (DT), random forest (RF), and AdaBoost (AB)). The bold numbers show the best value of each metric among considered datasets for each prediction algorithm. The colored italic numbers depict the best values of the metric across all models. The arrows show the improvement of metric value in relation to the metric value of the baseline model (dataset 1 All data).

The training models without outliers improves the performance of the models for each algorithm. In terms of prediction algorithms the best recall value is approached by random forest (84.06%), the best value of precision is acquired by AdaBoost (98.53%), the best values of other metrics have been obtained with logistic regression: accuracy is 87.02%, and auc is 89.63%. The best improvements for the metric are as follows: 3.58% for recall (RF cblof 3%), 2.76% for accuracy (DT hbos 10%), 5.19% for precision (DT hbos 10%), and 5.01% for auc roc (DT hbbos 10%).

The performance of logistic regression models has been improved by removing outliers detected by HBOS algorithm with all threshold assumptions. In the case of dropout prediction by decision tree algorithm, almost every model trained without outliers improves at least one metric (exceptions are cblof 3% and lof 10%). Similar to decision tree models, random forest models without outliers in train data improve at least one metric, which is usually the recall (except lof 3%, where the performance of the model is similar to the baseline model, and lof 10%). AdaBoost models trained without outliers detected as abnormal values, by kNN and CBLOF algorithms improve the recall metric, while models trained without outliers detected by HBOS improve precision and roc auc.

Table 5.2: Results of dropout prediction models.

Alg	Dataset	Rec	Acc	Prec	AUC	Test size	Train size
LR	1 All data	80.88%	85.64%	98.07%	88.64%	362	1447
	2 w/o knn 3%	80.48%	85.36%	98.06%	88.44%	362	1414
	3 w/o knn 5%	80.88%	85.64%	98.07%	88.64%	362	1384
	4 w/o knn 10%	79.68%	84.81%	98.04%	88.04%	362	1312
	5 w/o cblof 3%	80.88%	85.64%	98.07%	88.64%	362	1412
	6 w/o cblof 5%	80.48%	85.36%	98.06%	88.44%	362	1384
	7 w/o cblof 10%	79.68%	84.81%	98.04%	88.04%	362	1314
	8 w/o hbos 3%	81.27% ↑	85.91% ↑	98.08% ↑	88.84% ↑	362	1401
	9 w/o hbos 5%	81.27% ↑	85.91% ↑	98.08% ↑	88.84% ↑	362	1369
	10 w/o hbos 10%	**82.87%** ↑	***87.02%*** ↑	**98.11%** ↑	***89.63%*** ↑	362	1300
	11 w/o lof 3%	80.88%	85.64%	98.07%	88.64%	362	1401
	12 w/o lof 5%	80.48%	85.36%	98.06%	88.44%	362	1373
	13 w/o lof 10%	80.08%	85.08%	98.05%	88.24%	362	1301
DT	1 All data	80.88%	81.49%	91.44%	81.88%	362	1447
	2 w/o knn 3%	81.27% ↑	82.32% ↑	92.31% ↑	82.98% ↑	362	1414
	3 w/o knn 5%	79.28%	82.60% ↑	94.76% ↑	84.69% ↑	362	1384
	4 w/o knn 10%	81.67% ↑	82.87% ↑	92.76% ↑	83.63% ↑	362	1312
	5 w/o cblof 3%	80.08%	80.94%	91.36%	81.48%	362	1412
	6 w/o cblof 5%	79.68%	80.94%	91.74% ↑	81.73%	362	1384
	7 w/o cblof 10%	82.47% ↑	83.98% ↑	93.67% ↑	84.93% ↑	362	1314
	8 w/o hbos 3%	80.48%	82.32% ↑	93.09% ↑	83.48% ↑	362	1401
	9 w/o hbos 5%	80.48%	83.98% ↑	95.73% ↑	86.18% ↑	362	1369
	10 w/o hbos 10%	80.08%	**84.25%** ↑	**96.63%** ↑	**86.89%** ↑	362	1300
	11 w/o lof 3%	**83.67%** ↑	83.98% ↑	92.51% ↑	84.18% ↑	362	1401
	12 w/o lof 5%	77.29%	79.56%	91.94% ↑	80.99%	362	1373
	13 w/o lof 10%	79.68%	79.56%	89.69%	79.48%	362	1301
RF	1 All data	80.48%	84.25%	96.19%	86.64%	362	1447
	2 w/o knn 3%	82.07% ↑	85.64% ↑	96.71% ↑	87.88% ↑	362	1414
	3 w/o knn 5%	80.88% ↑	83.7%	94.86%	85.48%	362	1384
	4 w/o knn 10%	81.27% ↑	85.64% ↑	**97.61%** ↑	88.39% ↑	362	1312
	5 w/o cblof 3%	***84.06%*** ↑	**86.74%** ↑	96.35% ↑	**88.43%** ↑	362	1412
	6 w/o cblof 5%	81.27% ↑	84.25%	95.33%	86.13%	362	1384
	7 w/o cblof 10%	83.27% ↑	84.25%	93.3%	84.88%	362	1314
	8 w/o hbos 3%	81.27% ↑	85.64% ↑	**97.61%** ↑	88.39% ↑	362	1401
	9 w/o hbos 5%	79.68%	83.98%	96.62% ↑	86.69% ↑	362	1369
	10 w/o hbos 10%	81.27% ↑	84.81% ↑	96.23% ↑	87.03% ↑	362	1300
	11 w/o lof 3%	80.48%	84.25%	96.19%	86.64%	362	1401
	12 w/o lof 5%	80.48%	85.08% ↑	97.58% ↑	87.99% ↑	362	1373
	13 w/o lof 10%	79.28%	83.15%	95.67%	85.59%	362	1301
AB	1 All data	80.08%	85.08%	98.05%	88.24%	362	1447
	2 w/o knn 3%	81.27% ↑	**85.36%** ↑	97.14%	87.93%	362	1414
	3 w/o knn 5%	81.27% ↑	84.53%	95.77%	86.58%	362	1384
	4 w/o knn 10%	80.88% ↑	82.87%	93.55%	84.13%	362	1312
	5 w/o cblof 3%	81.27% ↑	**85.36%** ↑	97.14%	87.93%	362	1412
	6 w/o cblof 5%	81.67% ↑	83.98%	94.47%	85.43%	362	1384
	7 w/o cblof 10%	81.27% ↑	82.6%	92.73%	83.43%	362	1314
	8 w/o hbos 3%	79.68%	85.08%	98.52% ↑	88.49% ↑	362	1401
	9 w/o hbos 5%	80.08%	**85.36%** ↑	***98.53%*** ↑	**88.69%** ↑	362	1369
	10 w/o hbos 10%	80.08%	84.53%	97.1%	87.34%	362	1300
	11 w/o lof 3%	79.68%	84.53%	97.56%	87.59%	362	1401
	12 w/o lof 5%	78.88%	83.7%	97.06%	86.74%	362	1373
	13 w/o lof 10%	80.08%	83.98%	96.17%	86.44%	362	1301

5.4 Impact of outliers on the final score prediction

5.4.1 Methodology

The study of the impact of outliers on the score prediction has been conducted for the TPU dataset. Using the data of students' performance from 6 engineering faculties, we built the prediction models, where predictors variables are scores for enrolment tests and attestation tests (ETmath, ETph, ETch, AT1, AT2) and response variable is exam score (Exam). To prevent overfitting, methods of feature selection have been implemented. Univariate linear regression test returning F-statistics shows the significant dependence between all predictors variables and response variable. Thus, all features have been kept for building models.

Analogically to the dropout prediction, two approaches to handle outliers detected in the section 4.3.3 have been applied: a. keep all outliers, b. remove outliers from training data. The baseline model (approach a) keeps the outliers in the training data. For approach b, regarding each outlier detection algorithm (kNN, CBLOF, HBOS, LOF) and assumption (3%, 5%, 10%) outliers have been removed. The prediction models are based on the following prediction algorithms: linear regression (LnR), lasso regression (LsR), linear support vector regression (SVR) as commonly used algorithms that train interpretable models, and random forest (RF) as ensemble method which usually performs well. The analysis has been implemented using the Python scikit-learn library [140].

The data has been split in train and test disjoint sets (80/20%) randomly with shuffling. The hyper-parameter tuning has been implemented for each baseline model. The 10-fold cross-validation has been used for each training set. The computational details of hyperparameter tuning for final score prediction models are presented in the Appendix C Table C.2. The test set remained the same for each model and contained both inliers and outliers since outliers cannot be excluded from the prediction in a real-life scenario.

5.4.2 Results

The results of the prediction final score are presented in the Table 5.3. The rows correspond to the models trained with different datasets by different prediction algorithms (linear regression (LnR), lasso regression (LsR), support vector regression (SvR), and random forest (RF)), where dataset 1 keeps all data in the train set, and datasets 2 to 13 were trained without outliers detected by various algorithms with different threshold assumptions. The columns contain the error values (mean absolute error MAE and root mean square error RMSE) and test/-train size for each model. The arrows show the reducing error value in relation to the error of baseline model (dataset 1 All data).

Removing outliers from training data improves the prediction performance and reduces the errors for every prediction algorithm. In the case of linear regression, MAE has been improved for all datasets without outliers, except hbos 10% and lof 3%, where the minimal error is 0.6557 for cblof 10% dataset; the minimal root mean squared error is 0.9114 for knn 10% dataset. Prediction of the exam score by lasso regression without outliers in train data has decreased at least one error for each dataset, where the smallest value of MAE is 0.6559 for cblof 10%, and the smallest value of RMSE is 0.9108 for knn 10%. Support vector regression outcomes have been improved by removing outliers almost in any case (except hbos 10% and lof 3%), the smallest values for both errors are obtained for knn 10% dataset. The performance of ensemble random forest has been improved for each case with removing outliers except hbos 10% dataset. The smallest values of the errors are for dataset lof 10%.

In terms of prediction algorithm, the absolute leader is support vector regression, which has the smallest values of errors for the model trained without outliers detected by kNN with 10% assumption.

Table 5.3: Results of final score prediction models.

Alg	Dataset	MAE	RMSE	Test size	Train size
LnR	1 All data	0.6816	0.9218	215	860
	2 w/o knn 3%	0.6715 ↓	0.9192 ↓	215	833
	3 w/o knn 5%	0.6658 ↓	0.9164 ↓	215	812
	4 w/o knn 10%	0.6559 ↓	**0.9114** ↓	215	770
	5 w/o cblof 3%	0.6613 ↓	0.9140 ↓	215	831
	6 w/o cblof 5%	0.6561 ↓	0.9122 ↓	215	815
	7 w/o cblof 10%	**0.6557** ↓	0.9116 ↓	215	770
	8 w/o hbos 3%	0.6811 ↓	0.9221	215	832
	9 w/o hbos 5%	0.6805 ↓	0.9223	215	816
	10 w/o hbos 10%	0.6833	0.9227	215	772
	11 w/o lof 3%	0.6826	0.9275	215	832
	12 w/o lof 5%	0.6697 ↓	0.9207 ↓	215	815
	13 w/o lof 10%	0.6634 ↓	0.9153 ↓	215	772
LsR	1 All data	0.6810	0.9199	215	860
	2 w/o knn 3%	0.6697 ↓	0.9161 ↓	215	833
	3 w/o knn 5%	0.6659 ↓	0.9153 ↓	215	812
	4 w/o knn 10%	0.6561 ↓	**0.9108** ↓	215	770
	5 w/o cblof 3%	0.6622 ↓	0.9136 ↓	215	831
	6 w/o cblof 5%	0.6565 ↓	0.9118 ↓	215	815
	7 w/o cblof 10%	**0.6559** ↓	0.9110 ↓	215	770
	8 w/o hbos 3%	0.6804 ↓	0.9198 ↓	215	832
	9 w/o hbos 5%	0.6797 ↓	0.9195 ↓	215	816
	10 w/o hbos 10%	0.6825	0.9192 ↓	215	772
	11 w/o lof 3%	0.6782 ↓	0.9219	215	832
	12 w/o lof 5%	0.6675 ↓	0.9172 ↓	215	815
	13 w/o lof 10%	0.6637 ↓	0.9146 ↓	215	772
SVR	1 All data	0.6716	0.9204	215	860
	2 w/o knn 3%	0.6669 ↓	0.9195 ↓	215	833
	3 w/o knn 5%	0.6642 ↓	0.9165 ↓	215	812
	4 w/o knn 10%	***0.6527*** ↓	***0.9083*** ↓	215	770
	5 w/o cblof 3%	0.6602 ↓	0.9139 ↓	215	831
	6 w/o cblof 5%	0.6558 ↓	0.9110 ↓	215	815
	7 w/o cblof 10%	0.6551 ↓	0.9105 ↓	215	770
	8 w/o hbos 3%	0.6695 ↓	0.9214	215	832
	9 w/o hbos 5%	0.6704 ↓	0.9222	215	816
	10 w/o hbos 10%	0.6727	0.9214	215	772
	11 w/o lof 3%	0.6762	0.9255	215	832
	12 w/o lof 5%	0.6679 ↓	0.9207	215	815
	13 w/o lof 10%	0.6540 ↓	0.9089 ↓	215	772
RF	1 All data	0.7061	0.9552	215	860
	2 w/o knn 3%	0.6830 ↓	0.9407 ↓	215	833
	3 w/o knn 5%	0.6667 ↓	0.9253 ↓	215	812
	4 w/o knn 10%	0.6724 ↓	0.9279 ↓	215	770
	5 w/o cblof 3%	0.6729 ↓	0.9315 ↓	215	831
	6 w/o cblof 5%	0.6680 ↓	0.9310 ↓	215	815
	7 w/o cblof 10%	0.6724 ↓	0.9403 ↓	215	770
	8 w/o hbos 3%	0.7069	0.9372 ↓	215	832
	9 w/o hbos 5%	0.6955 ↓	0.9454 ↓	215	816
	10 w/o hbos 10%	0.7224	0.9607	215	772
	11 w/o lof 3%	0.6747 ↓	0.9247 ↓	215	832
	12 w/o lof 5%	0.6828 ↓	0.9390 ↓	215	815
	13 w/o lof 10%	**0.6577** ↓	**0.9192** ↓	215	772

5.5 Conclusion

This chapter is devoted to investigation of the impact of outliers on prediction models. In the educational domain the most wildly explored prediction tasks concern to dropout prediction and final score prediction. Through developing of the prediction models domain experts try to take profit and make decisions based on the information they gain, e.g. provide a specific advice for students with a high probability to fail the course or even drop out their studies. The existing studies in performance prediction barely mention how they deal with outliers, while they can affect the models.

In this chapter we consider the impact of outliers on the dropout prediction as well as the impact of outliers on final score prediction. To build dropout prediction models, the Student advice dataset has been used. To build final score prediction models, Tomsk polytechnic dataset has been utilized. To investigate how outliers affect prediction models, we compared their outcomes within two approaches: a. keep all data in train set, b. remove outliers from train set. Thus, for each prediction task we built 52 models:

- for 13 datasets: dataset 1 keeps all data, datasets 2 to 4 are without outliers detected by kNN algorithm with different threshold assumptions (3%, 5%, and 10%); datasets 5 to 7 are without outliers detected by CBLOF algorithm with different threshold assumptions; datasets 8 to 10 are without outliers detected by HBOS algorithm with different threshold assumptions; datasets 11 to 13 are without outliers detected by LOF algorithm with different threshold assumptions.

- for 4 prediction algorithms: logistic regression (LR), decision tree (DT), random forest (RF), and AdaBoost for dropout prediction; linear regression (LnR), lasso regression (LsR), support vector regression (SvR), and random forest regression (RF) for exam prediction.

The removal of outliers from training data improved the models' performance. For dropout prediction, the best metric value is always approached by the model with removing of outliers. For logistic regression models, the metric is improved by removing outliers detected by HBOS algorithm. Yet, the other prediction algorithms improve the metric values for models trained without outliers detected by kNN, CBLOF, and LOF algorithms. For prediction of the exam score, the smallest error is always obtained by removing outliers with 10% assumptions. Other models trained without outliers also improve the error value

compared to baseline models except models trained without hbos 10% and lof 3% outliers.

The results of this chapter show that outliers affect both dropout prediction and final score prediction. Consequently, the removing outliers from training data might be a good practice to improve quality of the prediction models. This mentioned in the papers devoted to the data prepossessing as cleaning step. However, the existing practices of dealing with outliers on the cleaning step unusually concern to the abnormal values, which are detected as abnormally high or abnormally low ($[mean - 3 * std; mean + 3 * std]$) values of one feature. In this chapter we proved that outliers detected as objects with rare characteristics in a set of certain features also affect prediction results. Furthermore, the removing all outliers from the train set is not always the solution. In our paper [DN.3], we show that removing of outliers with certain characteristics, namely intense passing exams, does not impact on dropout prediction. Thus, by removing all outliers from the training set we might lose the part of information.

Chapter 6

General conclusion and perspectives

Nowadays, educational institutions seek to improve their learning and teaching through analysing data collected while students are studying. They develop new data-based mechanisms and improve exciting models, which might help to improve academic results, stimulate students' motivation, or avoid drop out. This thesis proposes outlier detection as an effective tool for reinforcement data analysis and prediction in education.

Outlier detection is an important research area, especially in the conditions of the tremendous growth of digitally stored data. It has been widely investigated in many applications, where the definition of outlier/abnormality/anomalies varies from domain to domain. The main challenge of outlier detection is to determine the areas of normality and abnormality and the border between them. This task becomes more difficult when the ground truth is implicit (there are no labels that show which object is an outlier and which is not). Moreover, labelling of data is usually made by a domain expert, who has to decide which objects are outliers. The unlabelled data is the most frequent scenario for real-world datasets, which requires the implementation of unsupervised outlier detection algorithms.

Unsupervised outlier detection is a crafty task that usually appeals to human intervention. First, almost every unsupervised outlier detection algorithm has pre-specified parameters, e.g. number of neighbours for algorithms based on nearest neighbours graph or the number of clusters for algorithms based on partitioning clustering methods. Then threshold which separates inliers from outliers has to be defined. In the thesis, we detected outliers by unsupervised methods, and compared their performance in terms of robustness to the change of the parameters and different algorithms. The threshold in each case was defined as an assumption that a particular percentage of objects are outliers. This is the first limitation of the work, since one can always argue with the choice

of the threshold. Thus, we considered for numerical cases several assumptions (3%, 5%, and 10%), but mainly focus on 10% assumption, due to the thumb rule of outlier detection that partition of outliers does not exceed 10%.

The first part of the results is devoted to outlier detection in educational datasets to understand their characteristics and possible reasons for arising. Three case studies of outlier detection and data prepossessing in educational datasets have been accomplished according to the data nature and data context peculiarities. The first two case studies refer to numerical data, while the third case addresses sequential data. Concerning outlier detection in numerical data, two cases were studied: (i) students/outliers in the data of two semesters performance in the German University (SA dataset); (ii) students-outliers in the data of the high mathematics course performance during their studies at the first semester in the Russian University (TPU dataset). Firstly, statistically abnormal values were investigated. For SA case, abnormal values correspond to the extremely high points for features related to the number of failed, enrolled, passed in advance/delay courses. This means that the majority of students prefer to follow the curriculum and try to avoid fails and enrolments without passing exams. In the TPU case, the features with abnormal values related to the high scores for the first attestation and low scores for the exam. The rare excellent AT1 score might point out the difficulty of students' adaptation to the study in a university during their first semester. Meanwhile the rare low exam score shows the desire of students to avoid the course fail. The outliers detected by unsupervised outlier detection algorithms (kNN, CBLOF, HBOS, and LOF) for both cases are completely different since the data context varies. Yet, some tendencies of outlier behaviour in students' performance data can be generalized:

1. Students-outliers that have an abnormally high or low value of the certain features. For instance, clusters of outliers with a high number of passed exams in advance for SA dataset, or students with problems in attestation tests for TPU dataset.

2. The improvement/degradation of performance. For example, outliers that are characterized by procrastination in S1 for SA dataset, or outliers who passed well enrolments test and after failed attestation tests in TPU dataset.

For the third case study, the outliers were detected with two algorithms (Clustering and kNN). The results of these two algorithms are strongly overlapped.

Within outlier detection, we exposed the main reasons for arising outliers: delivering care before planning and inconsistent planning. Outliers can occur due to a bad/wrong understanding of the game processor's unawareness of good practice guidelines. The game allows players to override the good practice, for example, minimize the scheduling step and jump to the delivering step. The outliers pointed out the students, who were more engaged in the gaming process and not in the learning process. They do clicks and try to progress without using previous courses, knowledge, and skills.

These results allow to expand the knowledge about processes that exist in the educational environments. However, the main limitation of outlier detection from this point of view is the difficulty of estimating the positive impact. Even if some decisions are made based on the results of outlier detection, the new data should be collected and compared with previous one to prove the effectiveness of these decisions. This certainly takes time. Furthermore, in experiments related to students, it is always difficult to say if the changes are caused by decisions made before or by differences in control groups. Despite the possible obstacles, in future work, we plan to find a way to analyse the positive impact of outlier detection. For Tomsk polytechnic university and Berliner Hochschule für Technik, outlier detection can be included in the pipeline in the early warning system of detecting students with a high risk of dropouts. The investigation of behavioral tendencies of students-outliers with excellent characteristics can help to build the recommendation insights for students to improve their performance. For the serious game CLONE, outcomes of outlier detection can evoke changes in the game process which will help students to better educate. For example, pop-up windows with tips will help students to focus on the certain action.

The second part of the results is dedicated to the investigation of the impact of outliers on prediction models. Two types of prediction models in education have been explored: dropout prediction and final score prediction. The models were compared in terms of two approaches: a. to keep all outliers in training data, b. to remove outliers from training data. Unlike studies in other domains [133, 132], which focused on deleting statistically abnormal values, we considered removing outliers detected by unsupervised outlier detection algorithms. The findings of the experiment show that removing outliers from the training set improves the prediction models. Thus, removing outliers from training data might be a good practice to improve the quality of the prediction models. Several

limitations are noteworthy regarding these results, and they point out directions for future work. Firstly, the certain prediction algorithms have been applied for regression and classification tasks. In future work other, more advanced, algorithms as neural networks can be used. Secondly, all detected outliers have been removed from the training set. However, removing all outliers from the train set is not always the solution. In the paper [DN.3], we found out that outliers with a high number of passed exams do not impact dropout prediction models for the SA dataset. Thus, by removing all outliers, the models lose the part of information. In future work, we intend to investigate how other kinds of outliers may impact dropout prediction and how prediction models work for outliers situated in test data. Finally, the investigated datasets are rather small compared to the ones that are commonly used in the data science. The accumulation of training data might help to prevent underfitting, and to build more advanced models. Therefore, as the follow-up, we intend to perform a similar analysis for larger educational datasets in order to confirm our hypothesis.

Appendices

Appendix A

The computational details of unsupervised outlier detection algorithms for numerical data

Table A.1: The computational details for outlier detection in numerical data.

Step	Computational details	Explanation
The abnormal values	the value O is considered as abnormal if $O \notin [mean - 3 * std; mean + 3 * std]$	Three-sigma rule of thumb
Interquartile range	the value O is considered as abnormal if $O \notin [Q1 - 1.5 * IQR; Q3 + 1.5 * IQR]$	$1.5 * IQR$ rule of thumb
Choice of the algorithms	distance-based kNN, cluster-based CBLOF, histogram-based HBOS, density-based LOF	The state-of-the-art unsupervised outlier detection algorithms, which present different types of techniques
Algorithms' robustness study	kNN algorithm: the number of nearest neighbors $k \in [10; 50]$	Goldstein *et al.* [28]
Algorithms' robustness study	CBLOF algorithm: the clustering algorithm k-means, the parameters $\alpha = 0.95, \beta = 5$, the number of clusters $k \in [6; 11]$	α and β parameters as default proposed by inventors of the algorithm [66]
Algorithms' robustness study	HBOS algorithm: (i) SA case number of bins $b \in [40; 50]$, (ii) TPU case number of bins $b \in [30; 40]$	The rule of thumb determines b by the square root of the number of instances N [125]
Algorithms' robustness study	LOF algorithm: the number of nearest neighbors $k \in [10; 50]$	To achieve more meaningful results the lower boundary of the parameter $k \geq 10$ [69]
Outlier detection	kNN algorithm: $k = 50$	Goldstein *et al.* [28]
Outlier detection	CBLOF algorithm: clustering method k-means, $\alpha = 0.95, \beta = 5$, number of clusters for (i) SA $k = 7$, (ii) for TPU $k = 6$	The number of clusters - elbow curve and silhouette criteria, α and β as proposed in [69]
Outlier detection	HBOS algorithm: the number of bins (i) for SA $b = 43$, (ii) for TPU $b = 33$	$\sqrt{1809} = 43$, $\sqrt{1075} = 33$
Outlier detection	LOF algorithm: $k = 50$	$k \geq 10$ [69]
Clustering	k-means algorithm	The number of clusters k is defined with help of the elbow curve and silhouette criteria

Appendix B

The results of clustering of outliers detected by CBLOF, HBOS, and LOF algorithms

Table B.1: Characteristics of clusters of outliers detected by CBLOF algorithm with 10% assumption for SA dataset.

Cluster	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
N	28	2	38	22	32	29	17	13
Graduate	25	2	18	15	12	2	0	0
Dropout	3	0	20	7	20	27	17	13
N 5%	26	2	34	7	11	5	5	1
N 3%	24	2	24	0	3	0	2	0
Av_grade S1	2.33	4.58	5.79	2.9	2.13	3.95	4.79	4.01
Av_grade S2	2.88	2.5	2.62	2.49	2.73	4.64	5.05	3.85
F_ex S1	0.01	0.00	0.05	0.16	0.03	0.13	0.40	0.32
F_ex S2	0.01	0.00	0.08	0.00	0.04	0.55	0.13	0.04
En_ex S1	0.08	0.00	0.26	0.22	0.26	0.13	0.73	0.26
En_ex S2	0.09	0.00	0.06	0.11	0.16	0.27	0.15	1.11
P_ex_p S1	0.99	0.00	0.00	0.57	0.13	0.40	0.13	0.35
P_ex_p S2	0.07	1.00	0.03	0.51	0.07	0.10	0.01	0.02
P_ex_d S2	0.01	0.83	0.70	0.27	0.16	0.02	0.15	0.07
P_ex_a S1	3.24	0.2	0.02	0.26	0.48	0.06	0.16	0.12
P_ex_a S2	0.52	2.75	0.04	0.31	0.27	0.07	0.00	0.08

Table B.2: Characteristics of clusters of outliers detected by HBOS algorithm with 10% assumption for SA dataset.

Cluster	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
N	13	12	31	13	19	15	35	43
Graduate	10	2	16	0	8	0	7	10
Dropout	3	10	15	13	11	15	28	33
N 5%	7	6	14	5	9	6	22	22
N 3%	2	5	5	4	4	6	13	16
Av_grade S1	2.5	4.89	3.02	3.12	2.55	4.23	3.92	3.19
Av_grade S2	2.16	3.21	2.75	5.26	3.02	4.75	2.97	3.94
F_ex S1	0.05	0.28	0.19	0.18	0.02	0.4	0.42	0.18
F_ex S2	0.02	0.16	0.11	0.13	0.16	0.41	0.08	0.38
En_ex S1	0.18	0.65	0.19	0.37	0.54	0.36	0.14	0.13
En_ex S2	0.18	0.26	0.25	0.88	0.49	0.49	0.47	0.34
P_ex_p S1	0.4	0.03	0.62	0.39	0.35	0.2	0.41	0.68
P_ex_p S2	0.24	0.08	0.45	0.01	0.15	0.06	0.31	0.21
P_ex_d S2	0.2	0.44	0.18	0.00	0.15	0.01	0.16	0.06
P_ex_a S1	0.85	0.07	0.02	0.07	0.06	0.02	0.00	0.02
P_ex_a S2	0.49	0.14	0.06	0.02	0.05	0.00	0.1	0.09

Table B.3: Characteristics of clusters of outliers detected by LOF algorithm with 10% assumption for SA dataset.

Cluster	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
N	120	22	6	25	4	2	1	1
Graduate	101	9	3	24	2	0	0	0
Dropout	19	13	3	1	2	2	1	1
N 5%	62	10	5	9	3	2	0	0
N 3%	45	2	2	2	2	2	0	0
Av_grade S1	2.04	2.13	2.49	2.31	5.04	4.05	4.34	3.06
Av_grade S2	2.01	2.24	6.00	2.10	2.46	6.00	1.30	4.83
F_ex S1	0.02	0.02	0.00	0.01	0.05	0.40	0.80	0.00
F_ex S2	0.00	0.03	0.00	0.01	0.00	0.00	0.00	0.83
En_ex S1	0.05	0.13	0.30	0.09	0.40	1.80	0.00	0.00
En_ex S2	0.08	0.09	0.17	0.11	0.08	0.00	1.20	0.00
P_ex_p S1	0.90	0.26	1.00	0.99	0.00	0.40	0.20	1.00
P_ex_p S2	0.76	0.24	0.00	0.10	0.60	0.00	0.00	0.17
P_ex_d S2	0.02	0.06	0.00	0.01	0.69	0.00	0.00	0.00
P_ex_a S1	0.08	0.47	4.10	2.73	0.10	1.20	0.00	0.00
P_ex_a S2	0.11	0.42	0.00	0.63	1.74	0.00	0.20	0.00

Table B.4: Characteristics of clusters of outliers detected by CBLOF algorithm with 10% assumption for TPU dataset.

Cluster	**1**	**2**	**3**	**4**	**5**
N	25	42	20	11	10
N 5%	6	36	9	1	2
N 3%	2	23	6	1	1
ETmath	0.67	0.67	0.46	0.36	0.79
ETph	0.69	0.44	0.37	0.14	0.43
ETch	0.56	0.4	0.45	0.16	0.59
AT1	0.43	0.47	0.03	0.55	0.81
AT2	0.28	0.28	0.05	0.5	0.73
Exam	0.62	0.08	0.36	0.63	0.91

Table B.5: Characteristics of clusters of outliers detected by HBOS algorithm with 10% assumption for TPU dataset.

Cluster	**1**	**2**	**3**	**4**	**5**
N	41	14	9	16	28
N 5%	26	8	3	5	12
N 3%	17	5	1	2	8
ETmath	0.89	0.81	0.81	0.31	0.13
ETph	0.8	0.48	0.76	0.21	0.14
ETch	0.73	0.88	0.25	0.19	0.22
AT1	0.75	0.8	0.77	0.25	0.17
AT2	0.74	0.77	0.72	0.17	0.14
Exam	0.85	0.95	0.9	0.4	0.15

Table B.6: Characteristics of clusters of outliers detected by LOF algorithm with 10% assumption for TPU dataset.

Cluster	**1**	**2**	**3**	**4**	**5**
N	20	28	12	11	37
N 5%	9	16	6	4	19
N 3%	6	14	4	1	8
ETmath	0.42	0.44	0.77	0.7	0.69
ETph	0.16	0.37	0.61	0.81	0.46
ETch	0.22	0.42	0.86	0.2	0.47
AT1	0.52	0.08	0.55	0.65	0.49
AT2	0.57	0.1	0.39	0.65	0.23
Exam	0.66	0.43	0.72	0.84	0.2

Appendix C

The computational details of hyper parameters tuning for prediction models

Table C.1: The computational details of hyper parameters tuning for dropout prediction models.

Step	Computational details	Explanation
Optimal score	Accuracy	Strategy to evaluate the performance of the cross-validated model on the test set.
Logistic Regression	solver:[lbfgs, liblinear], penalty:[none, l1, l2], C:[0.01; 10; 0.01]	Solver determines the algorithm for solving optimization problem. The penalty specifies the penalty norm. Some penalties work with the certain solvers (for lbfgs: l2 and none; liblinear: l1 and l2). Parameter C inverses the regularization strength.
The best model: {solver: lbfgs, penalty: none}		
Decision Tree	criterion: [gini, entropy], max_features: [1;11;1]	Criterion determines the function which measures the quality of split. Max_features is the number of features considered when looking the best split.
The best model: {criterion: entropy, max_features: 2}		
Random Forest	criterion:[gini,entropy], max_features: [1;11;1], n_estimators:[10;100;1]	Criterion determines the function which measures the quality of split. Max_features is the number of features considered when looking the best split. The n_estimators corresponds to the number of trees in the forest.
The best model: {criterion: gini, max_features: 5, n_estimators: 20}		
AdaBoost	base_estimator:[Decision Tree], n_estimators:[10;100;1], learning_rate:[0.01; 1; 0.05]	The base_estimator refers the base estimator from which the boosted ensemble is built. N_estimators specifies the maximum number of estimators at which boosting is terminated. Learning rate is a weight applied to each classifier at each boosting iteration.
The best model: {learning_rate: 0.46, n_estimators: 40}		

Table C.2: The computational details for final score prediction models.

Step	**Computational details**	**Explanation**
Optimal score	neg_root_mean_squared _error	Mean squared error regression loss.
Lasso regression	alpha:[0.001; 1; 0.001]	Alpha is a constant that multiplies the L1 term, where alpha = 0 is equivalent to an ordinary least square, solved by the linear regression.
The best model: {alpha: 0.011}		
Support vector regression	C:[0.01; 10; 0.1], epsilon:[0.1; 0.4; 0.01]	Regularization parameter C which identifies the regularization (inversely proportional to C). The epsilon parameter specifies the epsilon-tube within which no penalty is associated in the training loss function with points predicted within a distance epsilon from the actual value.
The best model: {C: 0.1, epsilon: 0.4}		
Random Forest	max_features: [1,2,3,4,5], n_estimators:[10;100;1]	Max_features is the number of features considered when looking the best split. The n_estimators corresponds to the number of trees in the forest.
The best model: {max_features: 1, n_estimators: 90}		

Bibliography

[1] C. Romero and S. Ventura, "Data mining in education," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 3, no. 1, pp. 12–27, 2013.

[2] C. Lang, G. Siemens, A. Wise, and D. Gasevic, *Handbook of learning analytics*. SOLAR, Society for Learning Analytics and Research New York, NY, USA, 2017.

[3] R. E. Carneiro, P. Drapal, R. A. Fagundes, A. M. Maciel, and R. L. Rodrigues, "Anomaly detection on student assessment in e-learning environments," in *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*, vol. 2161, pp. 168–169, IEEE, 2019.

[4] M. M. Abu Tair and A. M. El-Halees, "Mining educational data to improve students' performance: a case study," *International Journal of Information*, vol. 2, no. 2, 2012.

[5] M. Ueno, "Online outlier detection system for learning time data in e-learning and it's evaluation," *Proc. of Computers and Advanced Technology in Education (CATE2004)*, 2004.

[6] S. Oeda and G. Hashimoto, "Log-data clustering analysis for dropout prediction in beginner programming classes," *Procedia Computer Science*, vol. 112, pp. 614–621, 2017.

[7] A. Freitas, T. Silva-Costa, F. Lopes, I. Garcia-Lema, A. Teixeira-Pinto, P. Brazdil, and A. Costa-Pereira, "Factors influencing hospital high length of stay outliers," *BMC health services research*, vol. 12, no. 1, pp. 1–10, 2012.

[8] M. Cyganska, "The impact factors on the hospital high length of stay outliers," *Procedia Economics and Finance*, vol. 39, pp. 251–255, 2016.

[9] S. O. Moreaux, C. A. Adongo, I. Mensah, and F. E. Amuquandoh, "There is information in the tails: Outliers in the food safety attitude-behaviour gap," *Food Control*, vol. 87, pp. 161–168, 2018.

[10] C. Romero and S. Ventura, "Educational data mining and learning analytics: An updated survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 3, p. e1355, 2020.

[11] L. Aulck, D. Nambi, N. Velagapudi, J. Blumenstock, and J. West, "Mining university registrar records to predict first-year undergraduate attrition.," *International Educational Data Mining Society*, 2019.

[12] J. Berens, K. Schneider, S. Görtz, S. Oster, and J. Burghoff, "Early detection of students at risk–predicting student dropouts using administrative student data and machine learning methods," 2018.

[13] G. W. Dekker, M. Pechenizkiy, and J. M. Vleeshouwers, "Predicting students drop out: A case study.," *International Working Group on Educational Data Mining*, 2009.

[14] J. Gardner, C. Brooks, and R. Baker, "Evaluating the fairness of predictive student models through slicing analysis," in *Proceedings of the 9th international conference on learning analytics & knowledge*, pp. 225–234, 2019.

[15] R. Yu, Q. Li, C. Fischer, S. Doroudi, and D. Xu, "Towards accurate and fair prediction of college success: Evaluating different sources of student data.," *International Educational Data Mining Society*, 2020.

[16] A. Ogan, E. Walker, R. Baker, M. M. T. Rodrigo, J. C. Soriano, and M. J. Castro, "Towards understanding how to assess help-seeking behavior across cultures," *International Journal of Artificial Intelligence in Education*, vol. 25, no. 2, pp. 229–248, 2015.

[17] J. R. Magnus, V. M. Polterovich, D. L. Danilov, and A. V. Savvateev, "Tolerance of cheating: An analysis across countries," *The Journal of Economic Education*, vol. 33, no. 2, pp. 125–135, 2002.

[18] A. Pardo and G. Siemens, "Ethical and privacy principles for learning analytics," *British Journal of Educational Technology*, vol. 45, no. 3, pp. 438–450, 2014.

[19] F. E. Grubbs, "Procedures for detecting outlying observations in samples," *Technometrics*, vol. 11, no. 1, pp. 1–21, 1969.

[20] D. M. Hawkins, *Identification of outliers*, vol. 11. Springer, 1980.

[21] V. Barnett and T. Lewis, "Outliers in statistical data," *Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics*, 1984.

[22] E. M. Knorr and R. T. Ng, "A unified notion of outliers: Properties and computation.," in *KDD*, vol. 97, pp. 219–222, 1997.

[23] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, pp. 85–126, Oct 2004.

[24] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, pp. 15:1–15:58, 2009.

[25] C. C. Aggarwal, "Outlier analysis," in *Data mining*, pp. 237–263, Springer, 2015.

[26] R. Kaur and S. Singh, "A survey of data mining and social network analysis based anomaly detection techniques," *Egyptian Informatics Journal*, vol. 17, no. 2, pp. 199 – 216, 2016.

[27] B. Gras, A. Brun, and A. Boyer, "Identifying grey sheep users in collaborative filtering: A distribution-based technique," in *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, UMAP '16, (New York, NY, USA), pp. 17–26, ACM, 2016.

[28] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PloS one*, vol. 11, no. 4, p. e0152173, 2016.

[29] U. D. of Education, "Federal student aid." https://studentaid.gov/data-center/student/portfolio. Accessed: 2021-05-20.

[30] G. F. I. price S&P 500. https://www.google.com/finance/quote/.INX:INDEXSP?window=5Y. Accessed: 2021-05-20.

[31] R. A. Kemmerer and G. Vigna, "Intrusion detection: a brief history and overview," *Computer*, vol. 35, no. 4, pp. supl27–supl30, 2002.

[32] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *computers & security*, vol. 28, no. 1-2, pp. 18–28, 2009.

[33] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: A survey," *Journal of Network and Computer Applications*, vol. 68, pp. 90–113, 2016.

[34] H. Weng, Z. Li, S. Ji, C. Chu, H. Lu, T. Du, and Q. He, "Online e-commerce fraud: A large-scale detection and analysis," in *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pp. 1435–1440, IEEE, 2018.

[35] V. Jain, "Perspective analysis of telecommunication fraud detection using data stream analytics and neural network classification based data mining," *International Journal of Information Technology*, vol. 9, no. 3, pp. 303–310, 2017.

[36] Q. Zhao, K. Chen, T. Li, Y. Yang, and X. Wang, "Detecting telecommunication fraud by understanding the contents of a call," *Cybersecurity*, vol. 1, no. 1, pp. 1–12, 2018.

[37] O. Salem, Y. Liu, and A. Mehaoua, "Anomaly detection in medical wireless sensor networks," *Journal of Computing Science and Engineering*, vol. 7, no. 4, pp. 272–284, 2013.

[38] A. Taboada-Crispi, H. Sahli, D. Hernandez-Pacheco, and A. Falcon-Ruiz, "Anomaly detection in medical image analysis," in *Handbook of research on advanced techniques in diagnostic imaging and biomedical applications*, pp. 426–446, IGI Global, 2009.

[39] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava, "Sensor network data fault types," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 3, pp. 1–29, 2009.

[40] D. Ramotsoela, A. Abu-Mahfouz, and G. Hancke, "A survey of anomaly detection in industrial wireless sensor networks with critical water system infrastructure as a case study," *Sensors*, vol. 18, no. 8, p. 2491, 2018.

[41] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD explorations newsletter*, vol. 19, no. 1, pp. 22–36, 2017.

[42] S. Ghanbari, A. B. Hashemi, and C. Amza, "Stage-aware anomaly detection through tracking log points," in *Proceedings of the 15th International Middleware Conference*, pp. 253–264, 2014.

[43] S. W. Yahaya, A. Lotfi, and M. Mahmud, "A consensus novelty detection ensemble approach for anomaly detection in activities of daily living," *Applied Soft Computing*, vol. 83, p. 105613, 2019.

[44] D. Dasgupta and N. S. Majumdar, "Anomaly detection in multidimensional data using negative selection algorithm," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, vol. 2, pp. 1039–1044, IEEE, 2002.

[45] J. Zhang, "Advancements of outlier detection: A survey," *ICST Transactions on Scalable Information Systems*, vol. 13, no. 1, pp. 1–26, 2013.

[46] H. Wang, M. J. Bah, and M. Hammad, "Progress in outlier detection techniques: A survey," *IEEE Access*, vol. 7, pp. 107964–108000, 2019.

[47] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data mining and knowledge discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[48] C. C. Aggarwal and S. Sathe, *Outlier ensembles: An introduction*. Springer, 2017.

[49] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[50] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, vol. 22, no. 1, pp. 949–961, 2019.

[51] J. Laurikkala, M. Juhola, E. Kentala, N. Lavrac, S. Miksch, and B. Kavsek, "Informal identification of outliers in medical data," in *Fifth international*

workshop on intelligent data analysis in medicine and pharmacology, vol. 1, pp. 20–24, Citeseer, 2000.

[52] B. Rosner, "Percentage points for a generalized esd many-outlier procedure," *Technometrics*, vol. 25, no. 2, pp. 165–172, 1983.

[53] R. D. Gibbons, D. Bhaumik, S. Aryal, *et al.*, *Statistical methods for groundwater monitoring*, vol. 2. Wiley Online Library, 2009.

[54] E. Eskin, "Anomaly detection over noisy data using learned probability distributions," 2000.

[55] X. Yang, L. J. Latecki, and D. Pokrajac, "Outlier detection with globally optimal exemplar-based gmm," in *Proceedings of the 2009 SIAM International Conference on Data Mining*, pp. 145–154, SIAM, 2009.

[56] X.-m. Tang, R.-x. Yuan, and J. Chen, "Outlier detection in energy disaggregation using subspace learning and gaussian mixture model," *Int. J. Control Autom*, vol. 8, no. 8, pp. 161–170, 2015.

[57] C. M. Park and J. Jeon, "Regression-based outlier detection of sensor measurements using independent variable synthesis," in *International Conference on Data Science*, pp. 78–86, Springer, 2015.

[58] M. Goldstein and A. Dengel, "Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm," 09 2012.

[59] L. J. Latecki, A. Lazarevic, and D. Pokrajac, "Outlier detection with kernel density functions," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp. 61–75, Springer, 2007.

[60] M. Pavlidou and G. Zioutas, "Kernel density outlier detector," in *Topics in Nonparametric Statistics*, pp. 241–250, Springer, 2014.

[61] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: Algorithms and applications," *The VLDB Journal*, vol. 8, pp. 237–253, Feb. 2000.

[62] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 427–438, 2000.

[63] V. Hautamäki, I. Kärkkäinen, and P. Fränti, "Outlier detection using k-nearest neighbour graph," in *ICPR*, 2004.

[64] F. Angiulli, S. Basta, and C. Pizzuti, "Distance-based detection and prediction of outliers," *IEEE transactions on knowledge and data engineering*, vol. 18, no. 2, pp. 145–160, 2005.

[65] A. Ghoting, S. Parthasarathy, and M. E. Otey, "Fast mining of distance-based outliers in high-dimensional datasets," *Data Mining and Knowledge Discovery*, vol. 16, no. 3, pp. 349–364, 2008.

[66] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recogn. Lett.*, vol. 24, pp. 1641–1650, June 2003.

[67] M. Amer and M. Goldstein, "Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer," 2012.

[68] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *KDD*, 1996.

[69] M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *ACM Sigmod Record*, vol. 29, pp. 93–104, 06 2000.

[70] W. Jin, A. K. Tung, J. Han, and W. Wang, "Ranking outliers using symmetric neighborhood relationship," in *Pacific-Asia conference on knowledge discovery and data mining*, pp. 577–593, Springer, 2006.

[71] M. Bai, X. Wang, J. Xin, and G. Wang, "An efficient algorithm for distributed density-based outlier detection on big data," *Neurocomputing*, vol. 181, pp. 19–28, 2016.

[72] B. Tang and H. He, "A local density-based approach for outlier detection," *Neurocomputing*, vol. 241, pp. 171–180, 2017.

[73] F. Angiulli and C. Pizzuti, "Fast outlier detection in high dimensional spaces," in *European conference on principles of data mining and knowledge discovery*, pp. 15–27, Springer, 2002.

[74] A. Boyer and G. Bonnin, "Higher education and the revolution of learning analytics," *Report of the International Council for Open and Distance Education (ICDE)*, 2016.

[75] L. P. Prieto, K. Sharma, P. Dillenbourg, and M. Jesús, "Teaching analytics: towards automatic extraction of orchestration graphs using wearable sensors," in *Proceedings of the sixth international conference on learning analytics & knowledge*, pp. 148–157, 2016.

[76] C. Alonso-Fernández, A. Calvo-Morata, M. Freire, I. Martínez-Ortiz, and B. Fernández-Manjón, "Applications of data science to game learning analytics data: A systematic literature review," *Computers & Education*, vol. 141, p. 103612, 2019.

[77] S. Custer, E. M. King, T. M. Atinc, L. Read, and T. Sethi, "Toward data-driven education systems: Insights into using information to measure results and manage change.," *Center for Universal Education at The Brookings Institution*, 2018.

[78] N. Bousbia and I. Belamri, "Which contribution does edm provide to computer-based learning environments?," in *Educational data mining*, pp. 3–28, Springer, 2014.

[79] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[80] S. García, J. Luengo, and F. Herrera, *Data preprocessing in data mining*, vol. 72. Springer, 2015.

[81] D. Pyle, "Data collection, preparation, quality, and visualization," *The handbook of data mining*, pp. 366–391, 2003.

[82] C. Romero, J. R. Romero, and S. Ventura, "A survey on pre-processing educational data," in *Educational data mining*, pp. 29–64, Springer, 2014.

[83] E. Rowe, M. V. Almeda, J. Asbell-Clarke, R. Scruggs, R. Baker, E. Bardar, and S. Gasca, "Assessing implicit computational thinking in zoombinis puzzle gameplay," *Computers in Human Behavior*, p. 106707, 2021.

[84] M. Denden, A. Tlili, F. Essalmi, and M. Jemni, "Implicit modeling of learners' personalities in a game-based learning environment using their gaming behaviors," *Smart Learning Environments*, vol. 5, pp. 1–19, 2018.

[85] J. Spacco, T. Winters, and T. Payne, "Inferring use cases from unit testing," in *AAAI workshop on educational data mining*, pp. 1–7, 2006.

[86] K. Wagner, A. Merceron, and P. Sauer, "Accuracy of a cross-program model for dropout prediction in higher education," in *Workshop Addressing Drop-Out Rates in Higher Education ADORE'2020*, pp. 744–749, 2020.

[87] H. Aldowah, H. Al-Samarraie, and W. Fauzy, "Educational data mining and learning analytics for 21st century higher education: A review and synthesis," *Telematics and Informatics*, vol. 37, pp. 13–49, 4 2019.

[88] J. Broadbent, "Comparing online and blended learner's self-regulated learning strategies and academic performance," *The Internet and Higher Education*, vol. 33, pp. 24–32, 2017.

[89] D. Tempelaar, B. Rienties, J. Mittelmeier, and Q. Nguyen, "Student profiling in a dispositional learning analytics application using formative assessment," *Computers in Human Behavior*, vol. 78, pp. 408–420, 2018.

[90] R. Asif, A. Merceron, S. A. Ali, and N. G. Haider, "Analyzing undergraduate students' performance using educational data mining," *Computers & Education*, vol. 113, pp. 177–194, 2017.

[91] X. Ochoa and A. Merceron, "Quantitative and qualitative analysis of the learning analytics and knowledge conference 2018.," *Journal of Learning Analytics*, vol. 5, no. 3, pp. 154–166, 2018.

[92] S. Halawa, D. Greene, and J. Mitchell, "Dropout prediction in moocs using learner activity features," *Proceedings of the second European MOOC stakeholder summit*, vol. 37, no. 1, pp. 58–65, 2014.

[93] W. Wang, H. Yu, and C. Miao, "Deep model for dropout prediction in moocs," in *Proceedings of the 2nd International Conference on Crowd Science and Engineering*, pp. 26–32, 2017.

[94] R. Flin, R. Patey, R. Glavin, and N. Maran, "Anaesthetists' non-technical skills," *British journal of anaesthesia*, vol. 105, no. 1, pp. 38–44, 2010.

[95] D. R. Michael and S. L. Chen, *Serious games: Games that educate, train, and inform.* Muska & Lipman/Premier-Trade, 2005.

[96] M. Zyda, "From visual simulation to virtual reality to games," *Computer*, vol. 38, pp. 25 – 32, 10 2005.

[97] G. Petri and C. Gresse von Wangenheim, "How games for computing education are evaluated? a systematic literature review," *Computers & Education*, vol. 107, pp. 68 – 90, 2017.

[98] F. Spyridonis and D. Daylamani-Zad, "A serious game to improve engagement with web accessibility guidelines," *Behaviour & Information Technology*, 01 2020.

[99] A. Abdul and P. Felicia, "Gameplay engagement and learning in game-based learning: A systematic review," *Review of Educational Research*, vol. 85, 03 2015.

[100] D. Rodriguez-Cerezo, A. Cabezuelo, M. Gomez-Albarran, and J.-L. Sierra, "Serious games in tertiary education: A case study concerning the comprehension of basic concepts in computer language implementation courses," *Computers in Human Behavior*, vol. 31, pp. 558–570, 02 2014.

[101] G. Gris, H. Alves, G. Assis, and S. R. L. de Souza, "The use of adapted games for assessment of mathematics and monetary skills," 2017.

[102] H. Pope and C. Mangram, "Wuzzit trouble: The influence of a digital math game on student number sense," *International Journal of Serious Games*, vol. 2, 12 2015.

[103] N. Shin, L. Sutherland, C. Norris, and E. Soloway, "Effects of game technology on elementary student learning in mathematics," *British Journal of Educational Technology*, vol. 43, 07 2012.

[104] J. Asbell-Clarke, E. Rowe, V. Almeda, T. Edwards, E. Bardar, S. Gasca, R. Baker, and R. Scruggs, "The development of students' computational

thinking practices in elementary-and middle-school classes using the learning game, zoombinis," *Computers in Human Behavior*, vol. 115, p. 106587, 2021.

[105] M.-T. Cheng, Y.-W. Lin, and H.-C. She, "Learning through playing virtual age: Exploring the interactions among student concept learning, gaming performance, in-game behaviors, and the use of in-game characters," *Computers & Education*, vol. 86, pp. 18 – 29, 2015.

[106] D. Kerr and G. K. W. K. Chung, "Identifying key features of student performance in educational video games and simulations through cluster analysis," in *EDM 2012*, 2012.

[107] J. Kang, D. An, L. Yan, and M. Liu, "Collaborative problem-solving process in a science serious game: Exploring group action similarity trajectory.," *International Educational Data Mining Society*, 2019.

[108] A. Slimani, E. Fatiha, E. Lotfi, O. Bakkali Yedri, and M. Sbert, "Learning analytics through serious games: Data mining algorithms for performance measurement and improvement purposes," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 31, p. 46, 01 2018.

[109] M. Martinez-Garza and D. Clark, "Investigating epistemic stances in game play with data mining," *International Journal of Gaming and Computer-Mediated Simulations*, vol. 9, pp. 1–40, 07 2017.

[110] G. Chung, "Guidelines for the design and implementation of game telemetry for serious games analytics," pp. 59–79, 01 2015.

[111] A. Tlili, F. Essalmi, M. Jemni, and D. Kinshuk, "An educational game for teaching computer architecture: Evaluation using learning analytics," pp. 1–6, 12 2015.

[112] A. R. Cano, B. Fernández-Manjón, and Á. J. García-Tejedor, "Using game learning analytics for validating the design of a learning game for adults with intellectual disabilities," *British Journal of Educational Technology*, vol. 49, no. 4, pp. 659–672, 2018.

[113] R. Bansal, N. Gaur, and S. N. Singh, "Outlier detection: Applications and techniques in data mining," *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*, pp. 373–377, 2016.

[114] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognition*, vol. 74, pp. 406–421, 2018.

[115] Z. z. Xing, J. Pei, and E. Keogh, "A brief survey on sequence classification," *SIGKDD Explorations*, vol. 12, pp. 40–48, 11 2010.

[116] Q. Zhao and S. S. Bhowmick, "Sequential pattern mining: A survey," *ITechnical Report CAIS Nayang Technological University Singapore*, vol. 1, no. 26, p. 135, 2003.

[117] A. Borah and B. Nath, "Rare pattern mining: challenges and future perspectives," *Complex & Intelligent Systems*, vol. 5, no. 1, pp. 1–23, 2019.

[118] S. A. A. D.-D. Approach. https://projekt.beuth-hochschule.de/students-advice/,Accessed: 2021-09-08.

[119] D. Baneres, M. E. Rodríguez-Gonzalez, and M. Serra, "An early feedback prediction system for learners at-risk within a first-year higher education course," *IEEE Transactions on Learning Technologies*, vol. 12, no. 2, pp. 249–263, 2019.

[120] I. Shcheglova, E. Gorbunova, and I. Chirikov, "The role of the first-year experience in student attrition," *Quality in Higher Education*, vol. 26, no. 3, pp. 307–322, 2020.

[121] C. Pons Lelardeux, H. Pingaud, M. Galaup, A. Ramolet, and P. Lagarrigue, "The challenge of designing interactive scenarios to train nurses on rostering problems in a virtual clinical unit," *Advances in Intelligent Systems and Computing*, vol. 1, 09 2018.

[122] X. Li, T. Wang, and H. Wang, "Exploring n-gram features in clickstream data for mooc learning achievement prediction," pp. 328–339, 03 2017.

[123] C. Loh and S. Yanyan, "Maximum similarity index (msi): A metric to differentiate the performance of novices vs. multiple-experts in serious games," *Computers in Human Behavior*, vol. 39, pp. 322–330, 10 2014.

[124] T. Martin, C. Smith, N. Forsgren, A. Aghababyan, P. Janisiewicz, and S. Baker, "Learning fractions by splitting: Using learning analytics to illuminate the development of mathematical understanding," *Journal of the Learning Sciences*, vol. 24, 08 2015.

[125] M. Goldstein, "Anomaly detection in large datasets," p. 248, 2 2014.

[126] L. Aiken, S. Clarke, D. Sloane, J. Sochalski, and J. Silber, "Hospital nurse staffing and patient mortality, nurse burnout, and job dissatisfaction," *JAMA : the journal of the American Medical Association*, vol. 288, pp. 1987–93, 11 2001.

[127] M. D. McHugh, A. Kutney-Lee, J. P. Cimiotti, D. M. Sloane, and L. H. Aiken, "Nurses' widespread job dissatisfaction, burnout, and frustration with health benefits signal problems for patient care," *Health Affairs*, vol. 30, no. 2, pp. 202–210, 2011.

[128] A. Khan and S. K. Ghosh, "Student performance analysis and prediction in classroom learning: A review of educational data mining studies," *Education and information technologies*, vol. 26, no. 1, pp. 205–240, 2021.

[129] H. Guruler and A. Istanbullu, "Modeling student performance in higher education using data mining," in *Educational Data Mining*, pp. 105–124, Springer, 2014.

[130] A. Sandoval, C. Gonzalez, R. Alarcon, K. Pichara, and M. Montenegro, "Centralized student performance prediction in large courses based on low-cost variables in an institutional context," *The Internet and Higher Education*, vol. 37, pp. 76–89, 2018.

[131] O. H. Lu, A. Y. Huang, J. C. Huang, A. J. Lin, H. Ogata, and S. J. Yang, "Applying learning analytics for the early prediction of students' academic performance in blended learning," *Journal of Educational Technology & Society*, vol. 21, no. 2, pp. 220–232, 2018.

[132] H. Do and K. S. Cetin, "Evaluation of the causes and impact of outliers on residential building energy use prediction using inverse modeling," *Building and Environment*, vol. 138, pp. 194–206, 2018.

[133] T. Nyitrai and M. Virág, "The effects of handling outliers on the performance of bankruptcy prediction models," *Socio-Economic Planning Sciences*, vol. 67, pp. 34–42, 2019.

[134] H. Yang, K. Huang, L. Chan, I. King, and M. R. Lyu, "Outliers treatment in support vector regression for financial time series prediction," in *Neural Information Processing* (N. R. Pal, N. Kasabov, R. K. Mudi, S. Pal, and S. K. Parui, eds.), (Berlin, Heidelberg), pp. 1260–1265, Springer Berlin Heidelberg, 2004.

[135] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* O'Reilly Media, 2019.

[136] X. Yan and X. Su, *Linear regression analysis: theory and computing.* World Scientific, 2009.

[137] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *the Journal of machine Learning research*, vol. 9, pp. 1871–1874, 2008.

[138] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learnin," *Cited on*, p. 33, 2009.

[139] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[140] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[141] C. Krauss, A. Merceron, and S. Arbanowski, "The timeliness deviation: A novel approach to evaluate educational recommender systems for closed-courses," in *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pp. 195–204, 2019.

List of Publications

2019

[DN.1] Daria Novoseltseva, Nadine Baptiste Jessel, Florence Sedes. Unsupervised Outlier Detection: students profiling in an effort to indicate learning problems in Higher Educational Institutions. 3rd Annual Learning & Student Analytics Conference (LSAC 2019), Oct 2019, Vandoeuvre-lès-Nancy, France. https://hal.archives-ouvertes.fr/hal-03268010

2020

[DN.2] Daria Novoseltseva, Catherine Pons Lelardeux, Nadine Jessel. Factors Affecting Success in a Digital Simulation Game for Nurse Training. 9th International Conference Games and Learning Alliance (GALA 2020), Dec 2020, Laval, France. pp.263-272, https://doi.org/10.1007/978-3-030-63464-3_25

2021

[DN.3] Daria Novoseltseva, Kerstin Wagner, Agathe Merceron, Petra Sauer, Nadine Jessel, Florence Sedes. Investigating the Impact of Outliers on Dropout Prediction in Higher Education. Proceedings of DELFI Workshops 2021. Dortmund (Online), Germany, September 13, 2021, pp. 120-130. https://nbn-resolving.org/urn:nbn:de:hbz:1393-opus4-7338

In progress

[DN.4] Daria Novoseltseva, Catherine Pons Lelardeux, Nadine Jessel. Examining Students' Behavior in a Digital Simulation Game for Nurse Training. Transactions on Learning Technologies. *Submitted.*

List of Figures

List of Tables

Acknowledgments

This work was supported by many people which I was lucky to meet and for which I would like to express my gratefulness.

First of all, I would like to thank my supervisors Nadine Jessel and Florence Sedes for providing the opportunity to perform research on this interesting topic and for supporting my initiatives.

I acknowledge my collaborators, which brought a solid contribution to my research. Thanks to Catherine Pons Lelardeux, who involved me in the interesting world of serious games. Thanks to Agathe Merceron, Petra Sauer, and Kerstin Wagner from the Student Advice project, who warmly welcomed me in Berlin. I'm very grateful for the insightful discussions, inspiring ideas, and for long-distance walkings with you.

I also would like to thank the people who inspired me on this journey. Thanks to Dmitriy Skvortsov, my first teacher of mathematics, who encouraged me to chose this professional way. Thanks to my previous supervisor from Tomsk Polytechnic University, Aleksandr Mikhalchuk, who spent a lot of hours explaining the statistical analysis by phone during my bachelor's and master's degrees. Thanks to Vladislav Spitsin, who drove me to continue as Ph.D.

Thanks to Evgeny and my family for loving me, supporting me, and always believing in me. Especially, thanks to my mum, who invested a lot of time and other resources in me and always being for me a model of commitment and persistence. Thanks to Manya, other friends and colleagues, who shared a lot of joyful moments with me.