ENhanced VIrtual learning Spaces using Applied Gaming in Education

H2020-ICT-24-2016

D3.2 Updated Predictive Analytics and Course Adaptation Methods

Dissemination level:	Public (PU)
Contractual data of delivery:	Month 16, January 31st, 2018
Actual date of delivery:	Month 17, February 28th, 2018
Work package:	WP3
Task:	T3.1 and T3.2
Type:	Demonstrator
Approval status:	final
Version:	1.0
Number of pages:	60
Filename:	D3.2_Updated_Predictive_Analytics_and_Course_Adaptation_Methods_Final.pdf

Abstract

The initial work on deep analytics in ENVISAGE was introduced in D3.1 with the focus on unsupervised methods and approaches used in game analytics. D3.2 now presents revised requirements and updated algorithms tailored towards educational settings. We provide an extended overview of "Educational Data Mining" and "AI in Education", and we explain how existing approaches fit the ENVISAGE project. We proceed by presenting unsupervised and supervised learning algorithms for deep analytics within the educational context. The work on unsupervised learning extends D3.1 and presents the clustering of students in the 2D Wind Energy Lab as an application. As examples for supervised learning, we introduce the prediction of at-risk students and proficiency levels of students. After identifying at-risk or low-performing students, the next step is to intervene and to help more students to succeed. Here, one approach is to adapt course material to better fit the students' needs. Therefore, we present approaches for dynamic content adaptation and explain how virtual labs can be adapted to personalize learning. Before presenting our conclusion, we show examples from the ENVISAGE platform and demonstrate the current capabilities of the deep analytics components.

co-funded by the European Union

Copyright

History

Version	Date	Reason	Revised by
(alpha)	30.11.2017	Initial version of the table of contents and abstract	Line Ebdrup Thomsen, Georgios N. Yannakakis
(beta)	18.02.2018	Beta version with content fixed and ready for review	Line Ebdrup Thomsen, Georgios N. Yannakakis
(final)	28.02.2018	Final version with feedback incorporated	Fabian Hadiji, Marc Müller

Author list

Organization	Name	Contact Information
GIO	Fabian Hadiji	fabian@goedle.io
GIO	Marc Müller	marc@goedle.io
UoM	Georgios N. Yannakakis	georgios.yannakakis@um.edu.mt
GIO	Aaqib Parvez Mohammed	aaqib@goedle.io
UoM	Antonios Liapis	antonios.liapis@um.edu.mt
UoM	Daniel Mercieca	arakanis@gmail.com
UoM	David Melhart	david.melhart@um.edu.mt
UoM	Daniele Gravina	daniele.gravina@um.edu.mt

Executive Summary

The initial work on predictive analytics in the ENVISAGE project was introduced in deliverable D3.1 [16] with the main focus on unsupervised methods and approaches used in games and game analytics. Deliverable D3.2 now presents revised requirements and updated algorithms tailored towards educational settings but also tested on games. We extend the overview on existing approaches in the fields of "Educational Data Mining" and "AI in Education", and we also explain how existing methods from other areas need to be adapted to fit the ENVISAGE setting.

We start by presenting revised unsupervised learning algorithms in Sec. 3 which directly extend the work in D3.1 [16]. We also present a case study in Sec. 3.2 which gives results on using different clustering algorithms on student data obtained from the 2D Wind Energy Lab. We proceed by presenting supervised learning algorithms for deep analytics within the educational context. Here, we explain two different use cases where supervised learning can be used to personalize the user experience in virtual labs. In particular, we present a prediction of at-risk students and a prediction of students' performance within the *Programme for International Student Assessment (PISA)* 2012 framework for proficiency classes. We also explain the necessary data preprocessing and feature engineering in detail. We do not only evaluate our algorithms on behavioral data from a chemistry lab and the 3D Wind Energy lab, but we also apply our algorithms to player data from a well known online game in order to validate the capabilities on a larger scale.

After identifying students at-risk or students who are predicted to have a lower performance, the next step is to intervene and to support those students to succeed. One possible approach is to adapt course material dynamically to better fit their needs. Therefore, we look at dynamic difficulty adjustment in Sec. 5 and explain how course material can be adapted to fit different segments of students. As an example, we use methods from statistics and machine learning, to adapt the content in a chemistry lab. We explain in detail how the chemistry lab can be adapted to allow the educator to define different learning strategies. We also describe different approaches that can be used to test and validate different learning strategies to find the optimal strategy for a particular lab. Since the implementation of the dynamic content adaptation is still on a prototype level for the chemistry lab, we also provide a case study from one of GIO's customers. This case study in Sec. 5.3.2 details how dynamic difficulty adjustment can be used in quiz games to improve the user experience. Quiz games can be related to educational settings easily and we pave the way for additional experiments in learning environments.

Before presenting our conclusion on the current efforts and motivating future work, we show examples from the ENVISAGE platform and demonstrate the current capabilities of the deep analytics components. The description of the demonstrator highlights how different deep analytics algorithms are already integrated into the authoring tool and shows how all pieces from the ENVISAGE project interact with each other.

Abbreviations and Acronyms

AI Artificial Intelligence

AIEd Artificial Intelligence in Education

ANN Artificial Neural Network

ANOVA Analysis of Variance

API Application Programming Interface

BnS Blade & Soul

CIG Computational Intelligence in Games

DDA Dynamic Difficulty Adjustment

EDM Educational Data Mining

GDPR General Data Protection Regulation

GPU Graphics Processing Unit

GTM Google Tag Manager

JSON JavaScript Object Notation

JSONP JavaScript Object Notation with Padding

KPI Key Performance Indicator

LMS Learning Management System

MAB Multi-Armed Bandit

MMORPG Massive Multiplayer Online Roleplay Game

MOOC Massive Open Online Course

PII Personally Identifiable Information

PISA Programme for International Student Assessment

SDK Software Development Kit

UCB Upper Confidence Bound

Contents

1 Introduction **8**

2 Analytics and AI in Education **10**
2.1 Recap of the AIEd market . . . 11
2.2 Restrictions in School Settings . . . 12

3 Unsupervised Learning **13**
3.1 Archetypal Clustering and Analysis . . . 13
3.2 Case Study: 2D Wind Energy Lab . . . 14

4 Supervised Learning for Educational Scenarios **17**
4.1 State-of-the-Art . . . 17
4.1.1 Academic Research . . . 17
4.1.2 Industrial Approaches . . . 18
4.2 Prediction of At-Risk Students . . . 20
4.2.1 Data Import, Feature Extraction, and Preprocessing . . . 21
4.2.2 Classification Algorithms . . . 22
4.2.3 Choosing an Algorithm and Parameters . . . 23
4.2.4 Fitting the Model and Making Predictions . . . 23
4.3 Student Performance Prediction . . . 24
4.4 Quality Measures . . . 25
4.5 Case Studies . . . 26
4.5.1 Chemistry Lab . . . 26
4.5.2 Blade & Soul . . . 28
4.5.3 3D Wind Energy Lab . . . 31

5 Adaptation of Learning Material **35**
5.1 State-of-the-Art . . . 35
5.1.1 Academic Research . . . 35
5.1.2 Industrial Approaches . . . 35
5.2 Dynamic Difficulty Adjustment . . . 36
5.2.1 Assessing Performance and Measuring Difficulty . . . 37
5.2.2 Designing Learning Strategies . . . 39
5.2.3 Automated Strategy Design: Genetic Algorithms . . . 39
5.2.4 A/B and Multivariate Testing for Learning Strategies . . . 40
5.2.5 Multi-Armed Bandits for Optimization . . . 41
5.2.6 Personalization of Strategies . . . 41
5.2.7 Closing the Loop: Reinforcement Learning . . . 42
5.3 Case Studies . . . 43
5.3.1 Chemistry Lab . . . 43
5.3.2 Mobile quiz Game . . . 45

6 Demo **47**
6.1 Prediction of At-Risk Students . . . 47
6.1.1 Data Upload View . . . 47
6.1.2 Intermediate View . . . 48
6.1.3 Results View . . . 48
6.1.4 Future Extensions . . . 51
6.2 Content Adaptation . . . 51
6.2.1 List of Strategies . . . 52
6.2.2 Add a Strategy . . . 52
6.2.3 Test a Strategy . . . 54
6.2.4 Future Views . . . 55

7 Outlook and Conclusion **56**

List of Figures

1 A comparison between cluster locations on the same dataset from a game when using either k-means (indicated by K's) or archetypal analysis (indicated by A's). . . . 13
2 k-means in the 2D Wind Energy Lab. The cluster labels assigned are as follows **III**: Reflective/communicative (class number 4); **II**: Advanced (class number 2); **I**: Beginner (class number 1); **<I**: No problem solver (class number 3). . . . 14
3 Archetypal analysis in the 2D Wind Energy Lab. The cluster labels assigned are as follows **III**: Reflective/communicative (class number 2); **II**: Advanced (class number 1); **I**: Beginner (class number 4); **<I**: No problem solver (class number 3). . . . 15
4 The four PISA clusters depicted as a pie chart in the visual analytics dashboard. For more details about the visual analytics service please refer to D2.4 [12]. . . . 16
5 The process pipeline for at-risk student predictions. . . . 19
6 The figures explains the observation window and churn window used in building the dataset. . . . 20
7 Event distribution in the chemistry lab dataset. . . . 27
8 Electron selection for water (H_2O) in the chemistry lab. . . . 28
9 Feature importance for churner in Blade & Soul. . . . 30
10 Event distribution of the Wind Energy Lab dataset. . . . 31
11 An example histogram of scores at the 3D Wind Energy Lab. . . . 32
12 The ANN approach adopted for predicting the level of the learner's competence (PISA score distribution) at the 3D Wind Energy Lab. The ANN maps in-game features to the score distribution (4 score classes according to the PISA 2012 classification). . . . 33
13 Similarly to the 2D Wind Energy Lab deep analytics solution, the four PISA clusters (four ANN outputs) are depicted as a pie chart in the visual analytics front end of the 3D Wind Energy Lab of the authoring tool. For more details about the visual analytics service please refer to D2.4 [12]. . . . 34
14 Multiple choice question in the 3D Wind Energy Lab as part of the grading. . . . 37
15 Infrastructure for content adaptation and dynamic difficulty adjustment. . . . 38
16 The original chemistry lab containing the default dropdown. . . . 43

17 The new chemistry lab where a molecule is picked based on a strategy obtained from the ENVISAGE API. 44
18 Correlation between time to solve a level and the player feedback regarding the difficulty. 45
19 An example strategy that increases and decreases the difficulty in a smooth manner to diversify the user experience. 46
20 Screenshot of the data upload for the prediction of at-risk students. 48
21 Screenshot after the data upload showing the experiment id which identifies the model being learned in the meantime. 49
22 Result page of the at-risk student prediction. 50
23 Authoring tool showing all available strategies for a virtual lab. 52
24 Adding a new strategy for a virtual lab from within the authoring tool. 53
25 Screen for testing a strategy. 54

List of Tables

1 Confusion Matrix 25
2 Blade & Soul trainings and test data. 29
3 Traffic light system for at-risk students. 51

1 Introduction

The goal of the ENVISAGE project is to improve virtual labs through a structured and data-driven process. First, this requires data from the learners, i.e., the students of virtual labs. This data is then analyzed and prepared to be used by educators. Next, an authoring tool is required that is capable of adapting existing virtual labs based on the insights from the data analysis. The deliverable at hand focuses on the data analysis and insights that can be automatically obtained from the data. While work package 2, e.g., deliverables D2.2 [25] and D2.3 [15], were concerned with shallow analytics, this deliverable focuses on deep analytics, i.e., using algorithms to analyze and understand behavioral data from students automatically. This deliverable describes the continuation of the work on deep analytics presented in deliverable D3.1 [16]. While D3.1 focused on unsupervised learning, the deliverable at hand extends the deep analytics part of the ENVISAGE project to additional types of machine learning. Additionally, this deliverable introduces approaches for content adaptation, allowing teachers to change the configuration of a virtual lab in order to test different learning strategies and to incorporate insights from the data analysis.

When talking about deep analytics, it is important to distinguish between different types of machine learning. Among other characteristics, machine learning differentiates between unsupervised and supervised learning to discover patterns in data. *Unsupervised Learning* does not require any labeled data and can cluster students for example in different groups without knowing these groups in advance. On the other hand, *Supervised Learning* requires annotated datasets in order to learn a model. In classification tasks, these labels categorize students in previously known groups. For example, one can build a dataset for training an algorithm with two labels by classifying students if they passed an exam or failed. Besides unsupervised and supervised learning, another form of machine learning exists which is called *Reinforcement Learning*. Here, the algorithm learns from actions and their rewards, i.e., there is not a gold set of annotated labels available in advance but a reward function instead that scores different actions. Different use cases require different types of machine learning and in this deliverable, we provide examples for each setting. For example, unsupervised learning is used to cluster students into different groups depending on their learning behavior in Sec. 3. Supervised learning is used to detect at-risk students and the learned models provide insights into the root causes of students losing interesting in a virtual lab in Sec. 4. Lastly, when designing new strategies to personalize and improve learning, there is no knowledge in advance how these new strategies perform. Here, and in the automation of the entire process, different forms of reinforcement learning can be used as motivated in Sec. 5.

Deliverable D3.1 already covered unsupervised learning for educational purposes. For example, it was described in Sec. 5 how k-means and archetypal clustering can be used to group students. Here, we present first results on using those algorithms on virtual lab data. To be more precise, the case study in Sec. 3.2 describes how data from the 2D Wind Energy Lab can be used to cluster students. This highlights how the past months have been used to actively transfer approaches from the gaming industry to the education sector. As the case study shows as well, the algorithms are equally applicable in education and result in interesting insights into the learner's behavior.

We will proceed as follows. First, we will summarize the efforts of the communities in *Artificial Intelligence (AI)* and Machine Learning when it comes to applying these algorithms in education and e-learning. Next, we will present the advancements of the unsupervised learning approaches as a sequel to deliverable D3.1. This includes one case study on virtual lab data. In the following section,

we will describe in detail how supervised learning is used to predict at-risk students and students' performance. This includes three case studies giving results on the algorithmic capabilities. Afterwards, our approach to dynamic content adaptation is presented and we also give two examples how virtual labs can benefit from the adaptation. Before giving an outlook on the next steps, we provide the reader with an extensive description of the demonstrators and provide sufficient instructions so that the results can be tested and verified.

2 Analytics and AI in Education

When reviewing ongoing research and available products for deep analytics, different terminologies can be observed which were established over the past years. A few years ago, *Big Data* was hyped and in particular companies were referring to this term. This wave set the expectation that large amounts of data would generate insights previously not available. There are plenty of books describing how Big Data can help to improve learning in school and higher education. From the Big Data hype, two research communities evolved: *Educational Data Mining (EDM)* and Learning Analytics [29, 4]. In D3.1 [16], the differences between EDM and Learning Analytics were discussed in detail. In a nutshell, EDM is a more automated approach to gain information from educational datasets, and Learning Analytics is a tool that helps (educational) analysts to interpret the data. In recent years, AI has become more popular again and people start to rephrase technologies in terms of AI to possibly reach a wider audience and to gain more traction. For example, algorithms from data mining are also often applied in AI scenarios.

As D3.1 also mentioned, the gaming industry is typically a few years ahead of other industries and in particular ahead of the education sector. AI and data-driven thinking is slowly becoming the status quo [37]. A lot of service providers in the market offer different technologies to personalize the gaming experience. Among other companies, *deltaDNA*[1] and *Optimove*[2] offer services to enhance games with help of AI and machine learning.

In contrast, there is a big gap in the usage of such technologies in the field of education. Mostly former researchers are building platforms and software that is capable of closing this gap. It is also important to distinguish between the markets in the United States and Europe. With the EU *General Data Protection Regulation (GDPR)*, e.g., Article 22 ("Automated individual decision-making, including profiling"), it will get more challenging for European education institutions to implement adaptive learning mechanism. The GDPR will establish high standards when it comes to data tracking and using such data for personalization. It will be necessary to obtain the consent of a learner when mechanisms are implemented that are applying automated decision-making based on personal information. Of course, when tracking children and teenagers in schools, this topic is even more sensitive and parents' consent is necessary to comply with privacy protection standards.

Visiting important trade shows in e-learning and digital education also underlines that the education sector is often inspired by technologies used in gaming. For example, *Virtual Reality* is certainly attracting a lot of attention in education, while the gaming industry has been pushing this technology for several years by now[3]. Additionally, learning apps in form of quizzes are also quite prominent. Often *Learning Management Systems (LMSs)* are extended to feature quiz apps to make learning more mobile and provide another engagement opportunity with the course material.

Returning to the discussion about different terminologies, one will certainly notice that there is a big overlap. For example, machine learning can be seen as a subfield of AI. Algorithms used in data mining, such as clustering or classification, are certainly found in machine learning as well. However, data mining also lends itself to Big Data and analytics, as statistical methods are used to detect trends in data and to extract actionable insights. While analytics typically still involves a lot of human labor, AI stands for an automated processing of data, offering insights that were not accessible to humans

[1] http://www.deltadna.com

[2] http://www.optimove.com

[3] http://blog.goedle.io/2018/02/01/trends-in-digital-education-at-learntec-2018/

before, and predicting future behavior.

In 2016, *Pearson* and the *UCL Knowledge Lab* published the open idea report *Intelligence Unleashed* [4] which discussed the opportunities and future development of *Artificial Intelligence in Education (AIEd)*. The report describes three kind of AIEd models. First, the pedagogical model which represents the knowledge and expertise of teaching. Second, the domain model which represents the knowledge of the subject that is being taught. And third, the learner model which represents the knowledge of the learner. Within the ENVISAGE project, there are intersections with all of those three models which is also highlighted by the composition of the consortium. The report also describes two AIEd applications. First, prediction of at-risk students which is already used in schools and universities. Second, a model-based adaptive tutor that has a content adaptation module.

It should be clarified that the prediction of at-risk students is not available as an out-of-the-box solution. However, there are two service providers on the market which are actively advertising the prediction of at-risk students. On the one hand, the open source LMS *Moodle*[4] and on the other hand the commercial company *Blackboard*[5]. The full adaptive tutor as envisioned in [21] as an AIEd application is to the best of our knowledge not implemented in any products yet. The content adaptation process has similarities to the *Dynamic Difficulty Adjustment (DDA)* which is currently being developed for content adaptation in the ENVISAGE project. The main difference is that the content adaption within ENVISAGE is a more generalized approach that adapts content based on behavioral information, not only based on domain knowledge. Sec. 4 will show how prediction of at-risk students is implemented and Sec. 5 explains how the content adaptation works in a virtual lab.

2.1 Recap of the AIEd market

Beside *Moodle* and *Blackboard*, there are other companies focusing on building a bridge between machine learning and education. A few successful examples are presented in the following. *Mindojo*[6] and *CENTURY*[7] are both platforms that provide AIEd in general. There are also more specialized companies, especially for subjects such as math, where a couple of companies are using AI for education. One example is *ScreenTime Learning*[8], an app that was released in December 2017 to prevent an excessive usage of smartphones and tablets by children. A child gets a math task which then locks the screen until it is solved. *ScreenTime Learning* uses DDA to adjust the difficulty of the math tasks for a child. Additional examples are the online courses by *Trueshelf*[9] or *bettermarks*[10]. Both offer adaptive learning in their courses which directly integrates in their learning material. *Adaptemy*[11] offers custom solutions for adapting educational content for learners. We will provide more examples in the sections below for particular use cases and applications.

[4] https://www.moodle.org
[5] https:/www.blackboard.com
[6] https://www.mindojo.com
[7] http://www.century.tech
[8] https://www.screentimelearning.com
[9] https://www.trueshelf.com
[10] https://www.bettermarks.com
[11] https://www.adaptemy.com

2.2 Restrictions in School Settings

One has to be careful when it comes to privacy regarding the analysis of a learner's behavior. Especially information about children are very sensitive. All analysis, methods, and software presented, make use of telemetric data but do not require personal information. By avoiding any kind of *Personally Identifiable Information (PII)* or demographic information, the privacy of children is respected. In many cases, anonymous data is already sufficient to gain valuable insights that help the educators to improve the quality of the course. Additionally, insights on the level of groups of students can be informative without harming the privacy of individuals.

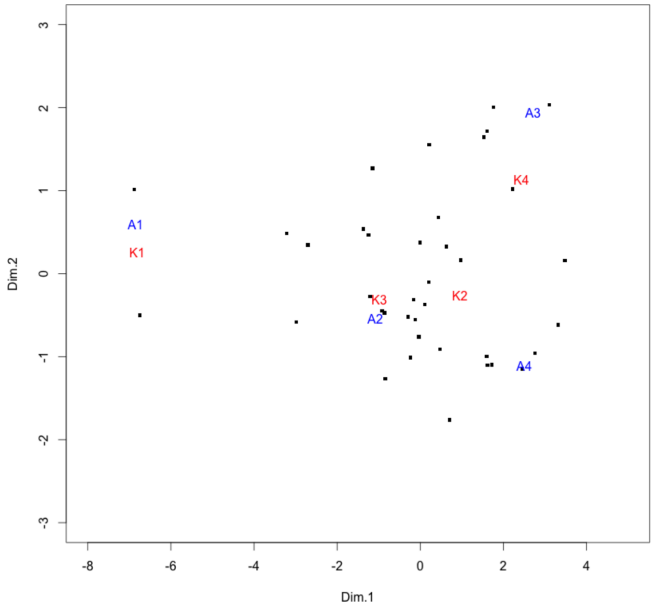


Figure 1: A comparison between cluster locations on the same dataset from a game when using either k-means (indicated by K's) or archetypal analysis (indicated by A's).

3 Unsupervised Learning

Unsupervised learning was the main approach adopted for the 2D Wind Energy Lab as presented and detailed in D3.1 [16]. In this section, we present the algorithms adopted and detail their application to the 2D Wind Energy Lab. A key goal of ENVISAGE is to understand how different students' behaviors are indicative of different groupings within, e.g., the whole student base or particular classes. In the terminology of EDM, this is a *Structure Discovery* problem, which is a well-known class of problems. For both, game analytics and educational data analysis, this is typically addressed by applying clustering methods, that partition observations into groups. Two clustering algorithms have been employed for the datasets collected from the virtual labs: *k-means* and *archetypal analysis*.

In brief, k-means allows for identifying groups based on typical behavior whereas archetypal analysis, allows for identifying groups based on extreme behavior. While both types of groupings may be of interest to teachers adapting virtual labs to suit their needs, archetypal analysis turned out to be a far more useful approach to clustering as it manages to better separate students within meaningful classes as mapped to the PISA 2012 categorization. The next section describes the final algorithm used.

3.1 Archetypal Clustering and Analysis

While k-means and similar algorithms such as *k-medoids* focus on identifying groups around average behavior in the data, archetypal analysis is focused on identifying extreme examples in the data. The algorithm works by drawing the minimally possible convex hull around all the observed data points. Using this hull, the algorithm searches for linear combinations of the observed data points that minimize Eq. 1 to determine coefficients that allow the data to be represented by the archetypes [5]:

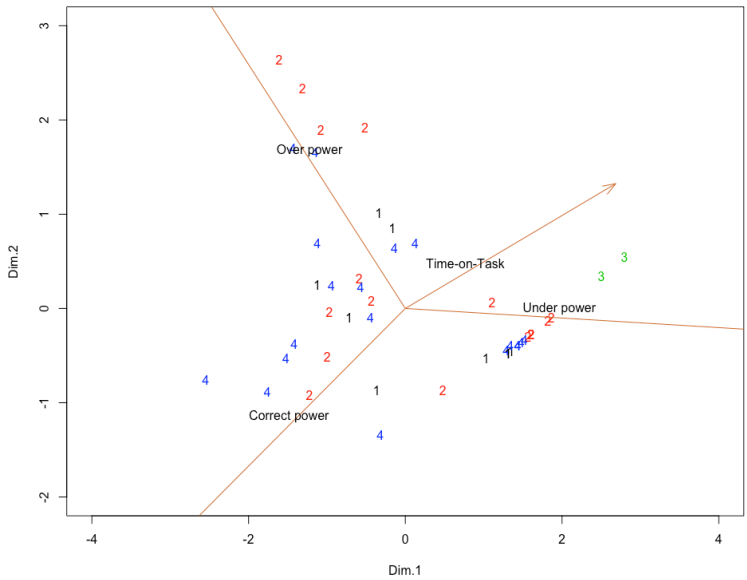


Figure 2: k-means in the 2D Wind Energy Lab. The cluster labels assigned are as follows **III**: Reflective/communicative (class number 4); **II**: Advanced (class number 2); **I**: Beginner (class number 1); **<I**: No problem solver (class number 3).

$$\arg\min_{S,H} \frac{1}{2}||X - XSH||_F^2 \tag{1}$$

Observations are then labeled according to their closeness to these archetypes, using a distance function, much akin to the way observations are labeled in k-means. When used in combination with k-means, archetypal analysis provides a useful alternative perspective that allows the user to see hypothetical extreme examples. This can help the user understand the overall directions of the behavior that the players of a game or the students in a digital learning environment are exhibiting. Fig. 1 shows a comparison of cluster centers found using k-means and archetypal analysis, respectively, when applied to the same dataset of player actions in a game.

3.2 Case Study: 2D Wind Energy Lab

Fig. 2 and Fig. 3 show a comparison of cluster centers found using k-means and archetypal analysis when applied to the same dataset of player actions in the 2D Wind Energy Lab. Both algorithms consider the following shallow analytics and tasks definitions (ad-hoc designed metrics) as described in D2.4 [12].

Time-on-task This metric measures the time it took the students to reach correct power from a state of being either under or over powered.

Correct power The amount of time the student has the wind simulation correctly powered.

Over power The amount of time a student has the wind simulation over-powered.

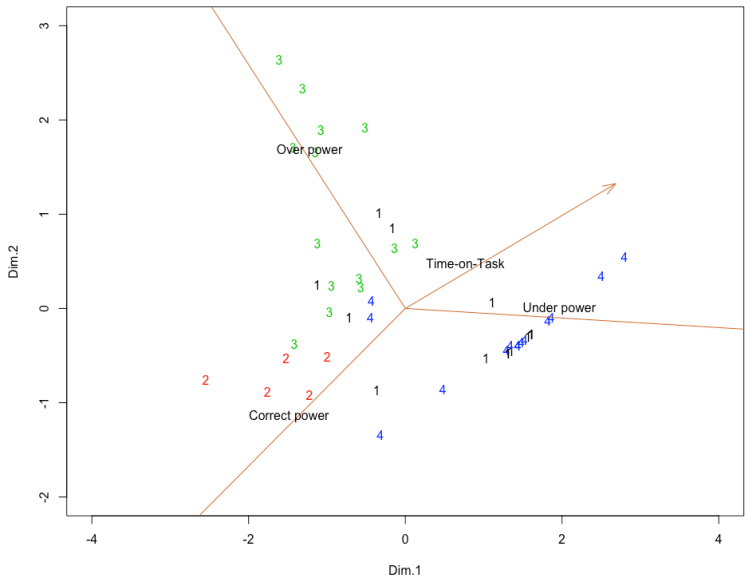


Figure 3: Archetypal analysis in the 2D Wind Energy Lab. The cluster labels assigned are as follows **III**: Reflective/communicative (class number 2); **II**: Advanced (class number 1); **I**: Beginner (class number 4); **<I**: No problem solver (class number 3).

Under power The amount of time a student has the wind simulation under-powered.

Based on the above in-lab on-task behaviors, learners are clustered into four typical groups (PISA 2012 classification; D1.1 [32]) by either method. In particular, the four clusters are as follows:

- **III:** Reflective/communicative
- **II:** Advanced
- **I:** Beginner
- **<I:** No problem solver

The difference in the way the two algorithms operate is rather visible from Fig. 2 and Fig. 3. The figures display the clusters as determined by the two algorithms and the data points within the four feature planes, which are projected onto the two-dimensional figure via principal-component analysis. We use this case study example to demonstrate the advantages of archetypal analysis over k-means in the task of automatically clustering learners according to their performance in the 2D Wind Energy Lab (PISA classification). As it is directly observable from Fig. 2, k-means places only two learners who under-power the Wind Energy Lab in their own category (category 3 in green color or PISA class >I), since they are rather dissimilar from the rest of the group. In general, k-means tends to place most students within the center of the hypersphere as this is the way the algorithm operates. In our particular domain, most students perform alike and that results in crowded data points for k-means to cluster. This shows coherence in the class, but does not show trends.

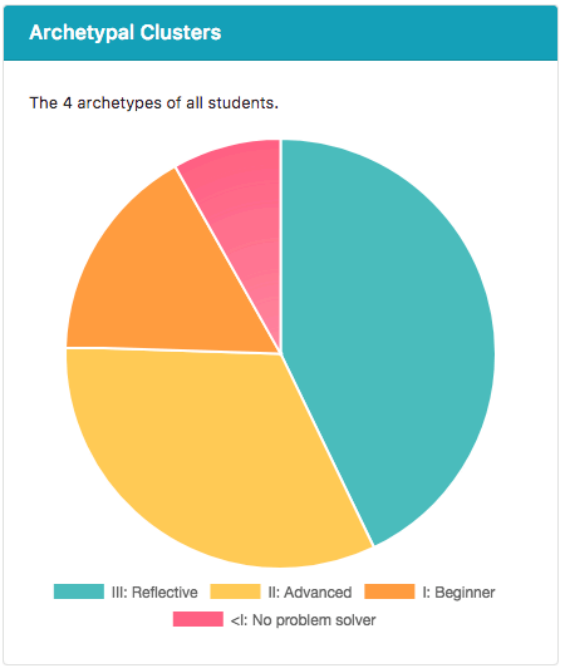


Figure 4: The four PISA clusters depicted as a pie chart in the visual analytics dashboard. For more details about the visual analytics service please refer to D2.4 [12].

In contrast, archetypal analysis, as displayed in Fig. 3, correctly identifies which directions learners are veering in, and assigns a group of students to the "low-performing" category (category 3 in green color or PISA class <I) and correctly identifies students moving toward the "high-performing" category (category 2 in red color or PISA class III). It is important to note that archetypal analysis, in contrast to k-means, is able to identify two groups of learners who (groups 3 and 4) underperformed in different ways: the first is over-powering the wind energy whereas the latter is under-powering the lab.

Also notice in Fig. 3 that time-on-task is inversely related to correct power, whereas under/over-powered is unrelated. In a nutshell, Fig. 3 illustrates that **good students are faster** than average/poor students, but **slow speed does not tell us what kind of errors a student would make**. This example dataset validates that time-on-task is a good indicator of performance and learnability. In particular, **lower time-on-task predicts better performance**.

Given the above qualitative characteristics and benefits of archetypal analysis over k-means in the Wind Energy Lab domain, we opted for the former approach for clustering learner performance in virtual labs. The cluster membership (<I to III) distribution is reported back through the analytics service to the visualization front-end. A depiction of the service is shown below in Fig. 4. The implementations used to experimentally realize the unsupervised models described in this document can be found at the following URL:

`https://github.com/Envisage-H2020/Analytics-Server`

It is important to note that the unsupervised learning approach was adopted only for the 2D Wind Energy Lab and not for its 3D version given the substantial differences between the two labs. In the supervised learning section, we detail the deep analytics approach employed for the 3D Wind Energy Lab (Sec. 4.5.3).

4 Supervised Learning for Educational Scenarios

In deliverable D1.1 [32], it was discussed that statistics from shallow analytics like time-on-task can be combined with deeper analytics to provide insights to a student's learning process. For example, an at-risk student prediction supports the identification of learners who are not going to continue using a virtual lab or having troubles following the course material. This leads to insights about students where one knows in advance that a learner gets stuck or does not finish parts of the solution. With such a forecast of students' behavior, it is possible to pro-actively support the students by improving their achievements and success. Two examples for proactive actions are (human based) support through blended learning or with (machine based) content adaptation. The content adaptation approach is described in Sec. 5. Another use case for supervised learning in education is predicting students' performance. To simplify the problem formulation, one can map the scores of students to the PISA 2012 categories. By doing so, each student gets a label based on the achieved score. Afterwards, one can learn a model that predicts which students fall in which PISA 2012 category based on their behavior. In this section, we describe those two use cases for supervised learning in greater detail. Before doing so, we give an overview on current approaches in this area. We finalize this section by presenting three case studies that show first results on using the algorithms on real-world data.

4.1 State-of-the-Art

At-risk student prediction is quite similar to *churn prediction*. In the gaming industry or telecommunication industry, churn prediction has been applied for years, if not decades. This is originated by the fact that retention is one of the most important *Key Performance Indicators (KPIs)* in these areas. Also in the academic research, churn prediction has been analyzed for years, while prediction of at-risk students in virtual labs is relatively new. The at-risk student prediction has already found its way in the industry, with companies offering it as a service. Performance prediction of students is a research field that has not found its way into products like at-risk student predictions yet. But, there are a lot of academic research projects which cover this topic. These focus mainly on higher education though. The state-of-the-art section gives an overview about churn prediction in other industries and the prediction of at-risk students in educational settings. Additionally, a brief overview about the current academic research on performance prediction is provided.

4.1.1 Academic Research

Because of the strong similarity between learners' behavior in virtual labs and players' or customers' behavior in games or apps, different resources were taken into account to develop predictions of at-risk students. Predicting churn has a long history. For example, in 2000 Mozer et al. [27] already published work on churn prediction for a telecommunication carrier. A more recent publication about churn prediction in a setting more similar to virtual labs can be found in [14]. Here, player churn in free-to-play mobile games was analyzed and predicted. The work in [14] also inspired the basic features which were used in the following sections. Additional inspiration for features, more focused on educational data, can be found in the literature about community inquiry models. This is also used by *Moodle* in their module for prediction of at-risk students. Inspired by the work from Garrison et. al [11], the features are based on three pillars: cognitive presence, social presence, and teacher

presence. Also the work by Marks et al. [23] and Slavin et al. [31] describe how important time on task is. This fact has also been acknowledged in previous deliverables within the ENVISAGE project (cf. D1.1 [32] and D1.4 [24]).

In e-learning, the information about at-risk students is very important. Prior knowledge about students possibly dropping out can be used to increase retention by taking proactive measurements to prevent the dropout from actually happening. In 2009, Lykourentzou et al. [22] applied machine learning on data from *Massive Open Online Courses (MOOCs)* to predict dropouts in online courses. Kai et al. [18] used student interaction data from online courses to build prediction models. These models predicted at-risk students and the future student registration behavior for online courses. The second use case is rather a conversion prediction, i.e., the prediction if a student will enroll for a course in the future. A conversion prediction can also be used to predict if a student passes an exam or not. In more advanced settings, this can be extended to even predict a student's score or grade in an exam. Having such information at hand can further help to improve students' success rate. Imagine having a list of students available a few weeks ahead of an exam that indicates which student could benefit from additional help. Most of the research on this topic is done with higher education institutions or online courses. Along those lines, Al-Seleem et al. [2] build a model that predicts a student's grade based on their academic records. The work by bin Mat et. al [6] covers student performance predictions in distance higher education. The authors also discuss the effectiveness of active learning methodologies in predicting student's behaviors. Shahiria et. al [28] present a systematical review of the literature on predicting student's behavior. This work covers approaches on predicting a student's performance and evaluates different algorithms.

4.1.2 Industrial Approaches

Moodle, one of the most frequently used open source LMS, has integrated an at-risk student prediction in their 2017 released version 3.4[12]. The prediction of at-risk students is integrated in the core of the software. The results of the predictions are binary, i.e., either a student drops out of a course or remains an active member. Besides these results, *Moodle* offers opportunities to reach out to at-risk students to influence their behavior in a positive way.

A more business oriented application is offered by *Blackboard*[13] since 2016. The solution is called *Blackboard Predict* and is currently in a beta phase. It is planned to be released in Q1/Q2 2018. *Blackboard* is a full service provider for digital education. This includes communication services for different stakeholders (e.g., teacher, student, or parent) and an LMS among other solutions. Their website provides a full catalog of products and services[14]. A deeper look at *Blackboard Predict* in particular shows interesting applications. *Blackboard Predict* consists of three parts:

- prediction
- visualization of results
- communication for engagement

[12] `https://docs.moodle.org/dev/Moodle_3.4_release_notes`
[13] `http://blog.blackboard.com/introducing-blackboard-predict/`
[14] `https://www.blackboard.com`

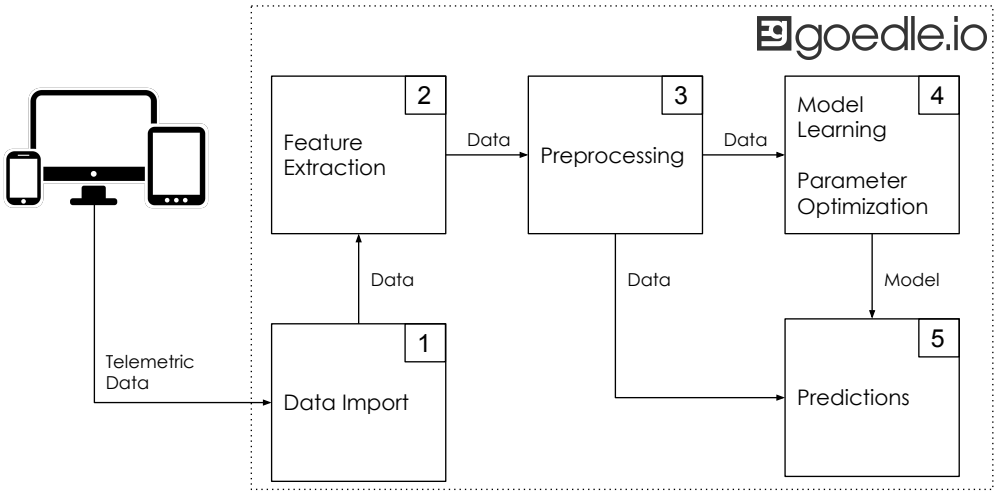


Figure 5: The process pipeline for at-risk student predictions.

Blackboard describes in their blog that they are aiming at a shift in perspective and want to focus more on behavioral information. Their argument is that there are no at-risk students in general, instead students are classified as being at-risk of not finishing a task. This definition does not only integrate better in an educational setting, it also points out the limitations of predictions and frames the at-risk prediction as a tool to improve software-based learning. *Moodle*'s prediction of at-risk students and *Blackboard Predict* have one thing in common, both approaches heavily rely on meaningful features. In turn, the availability of these features strongly depends on a well implemented tracking and clean datasets with behavioral data about students.

Moodle offers a predefined set of features and provides an internal tracking. This allows to create a model which can be applied within the *Moodle* LMS but at the cost of flexibility. One should also note that it is only possible to make at-risk predictions on a course-level at the moment. However, in *Moodle* one can add custom predictions and the entire prediction code is open source. While there is an *Application Programming Interface (API)* for adding data and creating new features, one should not underestimate the necessary expert knowledge in machine learning and software development to make use of these features. In comparison to *Moodle*, *Blackboard*'s offerings are more focused on consulting. For example, they help to identify and build features for custom predictions. One should also highlight that *Blackboard Predict* is not limited to their own platform. They also offer solutions for *Moodle*, for example at-risk predictions are part of *X-Ray*[15] which is *Blackboard*'s learning analytics suite for *Moodle*. Besides that, one can also integrate *Blackboard Predict* into custom solutions. They support a customer from defining a prediction, over tracking and aggregating the data, learning the model, and lastly using the predictions from the model for proactive measures. In the next section, we will give more details on the at-risk student prediction model in ENVISAGE.

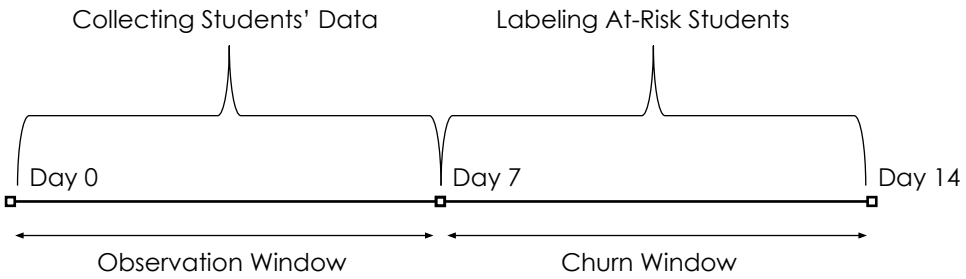


Figure 6: The figures explains the observation window and churn window used in building the dataset.

4.2 Prediction of At-Risk Students

Similar approaches to an at-risk student prediction are also used in the gaming industry, as well as for many other apps where retention is a crucial KPI. In gaming, one typically refers to churn prediction and in human resources departments, people refer to the prediction of *employee turnover*. Games and other industries provide many different approaches to keep players, users, or customers engaged. These methods range from basic interactions, over automated reminders, to the entire personalization of communication and the individualization of content.

While the gaming industry has been utilizing churn prediction for years, there are only a few services which offer a prediction of at-risk students as we have seen in the previous section. While the solution to the prediction problem may be technically similar, the surrounding conditions in education differ significantly. The at-risk prediction tries to classify learners who will stop using a virtual lab, fail an exam, or cancel an entire course or degree. In the following, we present a process pipeline for the prediction of at-risk students developed for the ENVISAGE project. The pipeline is depicted in Fig. 5.

We now describe the different components for the at-risk student prediction. It starts with the process of collecting or importing the data, box 1 in Fig. 5. Afterwards, the feature extraction process is triggered (box 2). Following this, the data is preprocessed (box 3) for being used in machine learning algorithms. The resulting data is then used as input for different classification algorithms (box 4). Lastly, the predictions are inferred based on the learned model (box 5). In Sec. 4.5.1 and 4.5.2, we will present two case studies that make use of the ENVSIAGE pipeline. The first case study is based on a virtual chemistry lab that was described in D1.1 [32] and the second case study is based on gaming data which is in its nature very similar to a virtual lab. Additional gaming data was taken into account, as the amount of data resulting from the virtual lab was limited at the point of writing this deliverable.

As we have previously described, being an at-risk student indicates a high likelihood of not completing a certain course, task, or stopping to learn. The prediction of at-risk students is based on supervised learning algorithms. Supervised learning means that the algorithm requires labeled training data. Therefore, historical data is needed. This means one needs data from the past that provides information about students that canceled their ambitions to learn for a course or exercise.

[15] `https://www.blackboard.com/education-analytics/xray-learning-analytics.html`

A possible labeling process looks like the one given in Fig. 6. Users are observed over a span of two weeks in total. Assuming, one wants to predict the at-risk students after one week, the "churn window" amounts to seven days. Correspondingly, we refer the "observation window" as the first seven days of the total time span of two weeks. To construct labeled training instances, one uses students' data from the first week to construct features and checks if they are active in the second week to label them. Students who are not active in the second week are labeled as "at-risk students", i.e., `true`. Those who are active in the second week are labeled as `false`. This corresponds to the binary classification which is used in *Moodle* as well. But, there is one pitfall regarding the PISA 2012 framework. On average, 10% of the learners have a high proficiency level. Some of these learners are very likely to not return to the lab because they do not require as much learning time compared to the average student. The algorithm now possibly identifies such learners as at-risk students because they are less likely to return. Adding additional learning content is not appropriate for them, as they are already performing well. For that reason, one should take performance into account before automating decisions or making content adaptations.

4.2.1 Data Import, Feature Extraction, and Preprocessing

Before the feature extraction can begin, the data has to be imported and prepared for the extraction process. Typically, data has to be aggregated on a user level and sorted chronologically. Often, additional meta data is added to the user profiles from external sources. For example, resolving IP addresses to locations. The quality of the feature extraction depends on the number of events and attributes, as well on a proper tracking which is the basis for obtaining the data. Often, data comes from different platforms and sources. For example, in the ENVISAGE project virtual lab data can be received from a *Google Tag Manager (GTM)* integration or from the ENVISAGE Unity *Software Development Kit (SDK)*. The data format and tracking scheme, which applies to GTM and the Unity SDK, is described in D2.1 [13]. Once the data has the appropriate format, the data aggregation and augmentation process is started. This process is also described in detail in D2.1 [13]. The case study about at-risk students in Sec. 4.5.1 and the churn prediction case study in Sec. 4.5.2 are both depending on this data aggregation and data augmentation process. The student performance prediction used the raw data directly and applied an additional preprocessing for the feature extraction. This is described in more detail in the case study in Sec. 4.5.3.

Features are the core of a machine learning model. They describe and represent the behavior of a student. The algorithms use features and their weights to build a model. An example feature is the count of a certain interaction. In the chemistry lab, this could be the information on how often a learner has added a bonding. Another feature could be the time between two sessions. This so called "inter-session time" is typically averaged over all sessions. An increasing inter-session time often indicates at-risk behavior. If enough learners have been observed, we can learn a model based on the features. The model can then classify if a learner is at risk of not coming back. The model will internally represent certain rules for different behaviors. For example, if learners, who are coming back frequently, have often added a bonding, the feature indicating the count of this event will have a strong impact, when discriminating those learners from at-risk students. A lot of the features are inspired by the work in [14, 30]. Based on GIO's platform, the ENVISAGE project has a large toolbox of features at its disposal. For the educational setting, and based on the educational relevant parameters proposed in D1.4 [24], we created additional features, which include:

time-on-task How much time does a student need for a certain task?

time between tasks How much time has passed between two tasks?

current absence time How long has the student been absent from the virtual lab?

Other features, that are more general, can be grouped in different kinds of behavioral descriptions:

basic activity Measuring basic activity such as the number of days a student has been active or the total number of sessions.

event counts Counting the number of times an event or an event-identifier combination occurs. E.g., the number of times a user added an electron to a bonding in the chemistry lab.

event values Mathematical operations on event values, e.g., the sum of correctly answered questions or the mean value of points scored.

curve fitting Curve fitting can be applied to time series data. Parameters, such as a positive slope of a inter-session time series, indicate an increasing motivation in the virtual lab. More details on this can be found in [14].

frequency Students' activity can be transformed from a time series to a frequency domain. This allows to estimate the strongest recurring frequency of a student.

social These features can count the number of connections within a social network of students[16]. Other features indicate if a student is connected to other classmates that may be important for the mutual learning progress.

4.2.2 Classification Algorithms

There is a variety of classification algorithms that can be used for the prediction of at-risk students. GIO's platform works agnostic of particular algorithms and chooses the most appropriate one for each problem and dataset. We transferred this agnostic approach to the at-risk student prediction within ENVSIAGE. Therefore, one the following algorithms is typically used within the system, depending on the datasets and additional parameters:

- Logistic Regression
- Naive Bayes
- k-Nearest-Neighbors
- Decision Trees
- Random Forests
- Gradient Tree Boosting

[16] One should note that these kinds of features require virtual labs that allow social interactions.

- Artificial Neural Networks
- Support Vector Machines

When testing different algorithms for the prediction of at-risk students, we relied on the implementation of the algorithms in the *scikit-learn*[17] Python library. For *Artificial Neural Networks (ANNs)*, we also used *TensorFlow*[18]. All ANNs ran on an *Nvidia* general purpose *Graphics Processing Unit (GPU)*.

4.2.3 Choosing an Algorithm and Parameters

Every algorithm has strengths and weaknesses. On top of that, algorithms typically cannot be used out of the box with default parameters. Therefore, we have to optimize the parameters first. This is typically done with help of an automatic parameter optimization.

There are different kinds of parameter optimizations. The simplest one is a brute-force grid search. A grid search finds the best configuration for an algorithm based on a given space of parameters. In a nutshell, an algorithm has different parameters, e.g., a logistic regression can be configured with an automatic normalization of the data, different optimization algorithms, various thresholds, and options. If one wants to validate if a parameter has impact on the result, one adds the parameter to the search space of the grid search. Then, the algorithm is evaluated on every combination of the passed parameters, i.e., on every instance in the search space. Typically, the evaluation is done with a cross validation to get stable and reliable results. The cross validation splits the training dataset into k different folds. Each fold is a random subset of the data. Based on the folds, the data is partitioned into a training and test datasets. Typically, one fold is used for testing and the remaining folds form the training dataset. Depending on the dataset and classification problem, different scores are used for the validation. *Accuracy* and *f1-score*, as described in Sec. 4.4, are typical examples.

A basic grid search is very time consuming because all possible combinations of a given parameter space need to be tested. A smarter and faster way to approximate the optimal parameters is a Bayesian optimization, e.g., as described in [9]. Bayesian optimization does not test every item in the entire search space but instead samples configurations from the space and tries to search the space in an intelligent way. This avoids testing all configurations of the search space and the approach tries to avoid configurations that are not promising. Naturally, this also introduces the risk of trapping into a local optimum, i.e., not finding the best parameters available in the search space.

4.2.4 Fitting the Model and Making Predictions

The whole procedure of parameter optimization is done to obtain a well performing prediction model. After the best parameters have been found, the model is trained with the winning parameters. The whole process is done in the “Model Learning and Parameter Optimization” part of the infrastructure (cf. box 4 in Fig. 5). Afterwards, the model is available in the GIO infrastructure and can classify new students (cf. box 5 in Fig. 5).

If one wants to know the probability of learners being at risk of not returning to a virtual lab, GIO’s API can be queried for these learners. By default, these predictions are done in a batch-wise fashion. Every night, learners active within a given time frame are taken into account. These learners

[17] `http://scikit-learn.org`
[18] `https://www.tensorflow.org`

are then classified based on their behavior. The result is a probability for each learner of returning to a virtual lab. Through the API, one can obtain a list of user identifiers and their at-risk probability. By predicting the current likelihood of a student every day, we create an at-risk profile of students. These profiles could support a teacher to respond if the probability increases over time for certain students. Platforms such as *Moodle* or *Blackboard* offer communication modules, allowing to interact with students directly. For example, if the at-risk probability increases, the teacher is alerted and encouraged to support the student. GIO's platform already offers similar capabilities for marketing purposes and is currently extending it to use cases within education. As discussed before, this needs to respect more sensitive privacy regulations and restrictions. Additionally, we have added an demonstrator for historic data for the ENVISAGE project. One can upload historic raw data in the format described in D2.1 [13] and get a first impression on what the model looks like. This is further described in the case study of the chemistry lab (Sec. 4.5.1), and also part of the demonstrator for the at-risk student use case (Sec. 6.1).

4.3 Student Performance Prediction

After explaining the prediction of at-risk students in greater detail, we will now shift our attention to the prediction of students' performance. Here. the objective is to predict a student's performance which is in most cases represented by a grade or a score. In a simplified setting, we might only want to predict if a student solves a quiz correctly. Similar to the at-risk student prediction, student performance prediction needs historical data which describes past behavior of students and includes a corresponding label for their performance, e.g., the reached score or grade of the students. Although it is mainly used in research on higher education at the moment, as described in Sec. 4.1.1, there are different use cases where student performance prediction is valuable. For example, it can help to identify students that will pass or fail an exam, or drop out of school due to low scores or grades. This effects not only the students' future, but it also leads to financial losses and a negative reputation for schools, colleges, and universities. Needless to say that this holds regardless whether these institutions are private or public. For example, the German education system is estimated to lose every year about €2.2 billions due to university drop outs.[19].

Within the ENVISAGE project, we focus on high school students. Therefore, we decided to predict the students' membership in one of the four PISA 2012 proficiency classes. Typically, virtual labs track scores for solving different problems. We can then map these scores from different tests and quizzes to the PISA categories.

When looking at the simplified case of predicting whether a student passes an exam, we can reduce the problem to a binary classification problem akin to the prediction of students at-risk. As we have described above, prediction of at-risk students is also know as churn prediction in other industries. Similarly, the simplified performance prediction can be seen as a "conversion prediction". Typically, a conversion prediction classifies users in two groups. One group contains all users for which a particular conversion event has been observed. The second group contains only users without this conversion event. Conversion prediction is often used in (mobile) games to predict if a certain stage in the game will be reached by a player or if a player will buy in-game items. Other examples occur in e-commerce settings where the cancellation of subscriptions can be predicted.

[19] https://his-he.de/meta/presse/detail/news/studienabbruch-staat-vergeudet-jaehrlich-22-milliarden-euro.html

In this deliverable, the prediction of students' performance is exemplified in a third case study below (Sec. 4.5.3). The case study is based on data from the 3D Wind Energy Lab and the data was collected during a recent pilot test at EA. Here, the tracked raw data was directly used without the preprocessing infrastructure in Fig. 5. To generate labels for the machine learning algorithm, the score system of the 3D Wind Energy Lab was aligned with the PISA 2012 proficiency classes. The case study shows how these categories can be predicted successfully.

4.4 Quality Measures

		predicted class	
		Yes	No
actual class	Yes	True Positive (TP)	True Negative (TN)
	No	False Positive (FP)	False Negative (FN)

Table 1: Confusion Matrix

To evaluate the predictions and measure the quality of a model, we mainly use two metrics. The presented quality measures in this section describe the quality of the algorithms from a statistical perspective. It should be mentioned that these metrics are not meant to be used by educators without additional explanation. There are different measures that can be used to evaluate a model and we will now explain *accuracy* and the *f1-score*. Depending on the dataset and algorithm, one has to figure out what scoring method is better suited. Additionally, the output of the predictions can be represented with help of a *confusion matrix*. An example of such a confusion matrix is shown in Tbl. 1. Accuracy is the ratio of correctly predicted observations and the total number of observations:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{2}$$

While this metric is quite intuitive, it should be used with caution. If the data is well balanced, i.e., all classes occur with similar frequency, accuracy gives a good idea about the performance. However, in the case of unbalanced datasets, where in the extreme case 99% of the students do not finish the course, a trivial algorithm can achieve an accuracy of 99% by always returning a negative prediction. Therefore, accuracy is not meaningful in this example. In particular, we would be interested in an algorithm that can detect the 1% of students who finish the course and the metric should prefer algorithms performing well on this task.

The f1-score is based on the *precision* and *recall* metrics. Precision, $TP/(TP+FP)$, shows how many students are correctly identified at risk. Recall, $TP/(TP+FN)$, calculates how many students among all at-risk students were correctly identified as such. The f1-score is the harmonic mean of precision and recall:

$$\text{f1-score} = 2 \cdot \frac{\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}} \tag{3}$$

This score takes both, false positives and false negatives, into account. The f1-score should be used in particular if the distribution of labels in the dataset is unbalanced. For the predictions of at-risk students, the f1-score is typically used as the classes are often not well balanced. In the case of the chemistry lab, the data showed only a very small number of users who returned to the chemistry

lab. Although the dataset in the churn prediction case study (Sec. 4.5.2) is more balanced, the data providers and organizers of the associated challenge decided to use the f1-score as well. The metrics discussed so far, were defined for binary classification tasks. However, the prediction of students' performance results in more than two classes. For such predictions, the accuracy and f1-score can be extended easily to the multi-class setting.

4.5 Case Studies

The following three case studies are based on three different datasets. First, the GoLab Organic Molecule Covalent Bonding virtual lab is used. This lab was already described in D1.1 [32]. It was initially developed to prepare learners for chemistry exams. This settings is a good environment for a continuous and repetitive usage of the lab. A recurring usage of the lab motivates the application of the at-risk student prediction. Initially, the data of the Wind Energy Lab (cf. D1.4 [24]) was intended to be used to forecast a student's at-risk behavior as well. However, the Wind Energy Lab is constructed in such a way that it is played only once. From a pedagogical perspective, repeating the Wind Energy Lab does not make as much sense as the chemistry lab. However, this does not imply that deep analytics or predictions cannot be used in the Wind Energy Lab in general. For the second case study, we are using data from a *Massive Multiplayer Online Roleplay Game (MMORPG)*, made available to us in a churn prediction challenge at the *Computational Intelligence in Games (CIG)* conference in 2017.

While the first two case studies address the prediction of at-risk students, respectively churn prediction, the third case study addresses the student performance prediction. Here, the data of the 3D Wind Energy Lab was used which also highlights that the 3D Wind Energy Lab is indeed well suited for deep analytics. The case study shows how to predict a student's affiliation in one of the four PISA 2012 proficiency classes.

As mentioned in Sec. 3.2, there is a 2D and a 3D version of the Wind Energy Lab. Besides the graphic design, there are two major differences between both versions. While the 2D version focuses only on configuring the environmental parameters to generate enough energy, the 3D version has significantly more features. There are different landscapes and the user journey is much more diversified. Additionally, the learner gets a quiz at the end of a simulation. Due to those substantial changes, the data tracking is more sophisticated as well which means a better data basis for machine learning algorithms. For classifying a student in one of the PISA categories, the data of the 3D lab was used.

4.5.1 Chemistry Lab

The dataset for the chemistry lab contains $2{,}079$ events from 107 unique users with 21 different types of events. This is roughly the available data in December 2017. To be more precise, the students were observed from May 31st, 2017, until December 14th, 2017. In total, we had a count of 107 students which is a rather small amount of users for applying machine learning algorithms. Due to this small user count, the case study qualifies as a feasibility study or prototype, which shows the general capabilities of the at-risk student prediction on real education data. Over the entire timespan, the students used the lab in 118 sessions. For a better understanding of the dataset, the statistics in Fig. 7 give a brief overview on the event distribution. A deeper look at the distribution of the

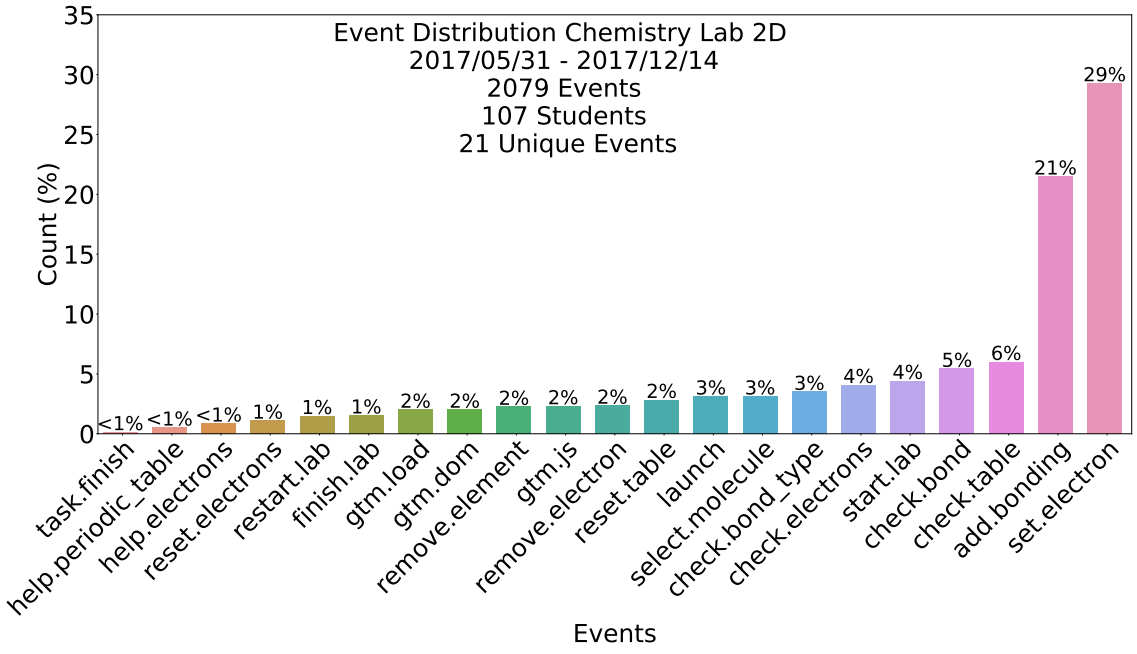
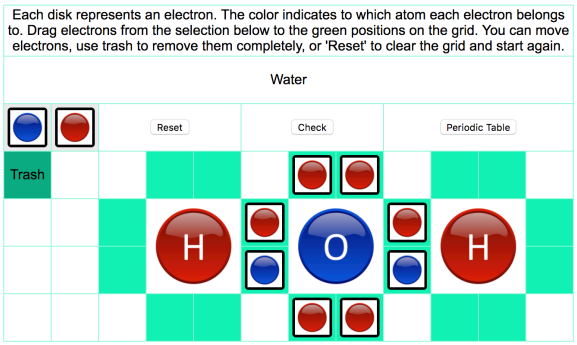


Figure 7: Event distribution in the chemistry lab dataset.

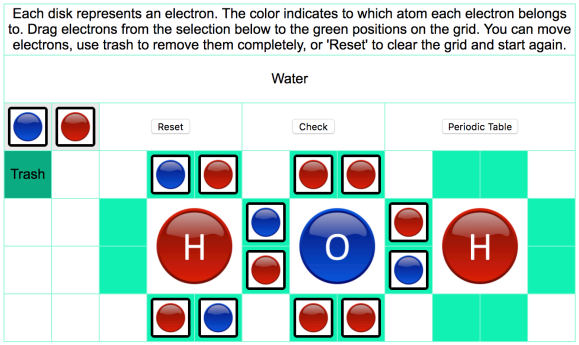
events shows that $set.electron$ and $add.bonding$ represent approximately 50% of all events. This is due to the fact, that the chemistry lab is designed in such a way that both events are triggered frequently by students in each iteration. To be more precise, these events are always triggered when a student selects an electron and drops it to one of the bonding positions. As one can see in the example in Fig. 8a, a correct solution requires 8 electron selections and 8 positioning events. When a student now moves an electron to a wrong position as depicted in Fig. 8b, a rearrangement of the electrons is necessary and even more events are triggered. Compared to the other tracked events, the total number of events from these two types will always be much higher. Additionally to this event information, locale information about the country and language is available for the students as well. We observed that the most popular origin of the students was the US, and the most popular language was English accordingly. This is a surprising insight about the students itself because the lab was not promoted in the US. We currently assume that a large number of bots, for example from search engines, visited the lab frequently. With this data at hand, the learning routine for a model predicting at-risk students was started.

To learn a model, the students were observed for 7 days and the churn window had a timespan of 28 days. With these parameters, we labeled 99 users as at-risk students, respectively churners, because they only used the lab within the first seven days and did not return in the following 28 days. On the other hand, 8 students were labeled as frequent users or students returning to the lab. However, this ratio was not surprising due to the fact that the lab was not in permanent use or part of the curriculum in the last months.

After the model for the chemistry lab was learned, insights and quality estimates were accessible. As a quality measure, the f1-score was used, as the dataset is not well balanced (cf. Sec. 4.4). Looking at the statistics above, only 7.5% of the students return to the lab a second time. The model achieves an f1-score of 0.96 which is a very good result, as we will later see in comparison to the second case



(a) Correct electron positions.



(b) Rearrangement of electrons is necessary.

Figure 8: Electron selection for water (H_2O) in the chemistry lab.

study in Sec. 4.5.2. One of the most helpful insights from the model is a list of the most important features used for the prediction. The machine learning algorithm determined the following five features based on custom events as most important:

- `remove.element`
- `task.finish`
- `remove.electron`
- `check.electrons`
- `start.lab`

This list of features can support teachers for improvements of the lab design. For example, compared to Fig. 7, where $set.electrons$ was the most used event, this event does not appear in the list of the most important events. This underlines the power of machine learning algorithms which are capable of finding important events that do not solely rely on the highest frequency but instead on the discriminative power. Similarly, the event $remove.element$ only represents 2% of the total events but the algorithm identified it as an important event in the case of predicting at-risk students. Such insights are easily accessible and can now be interpreted from a pedagogical point of view for further measures to improve the lab. The results from this feasibility study show that the system is capable of predicting at-risk students. The next case study will additionally show that the ENVISAGE platform can also handle millions of events and scales to Big Data settings.

4.5.2 Blade & Soul

As pointed out in the state-of-the-art section, all machine learning approaches require a dataset that allows to build features. For the ENVISAGE project, we were lacking a dataset from the virtual labs containing thousands of students with millions of events. Due to this problem, we looked for a dataset which has a high similarity to educational apps but is larger in size than the dataset from the chemistry lab. One great solution to the problem was participating in the game data mining competition as part of IEEE's CIG 2017. *NCSOFT*, one the world's largest game studios for MMORPGs,

provided a dataset with telemetric user data from their highly successful *Blade & Soul (BnS)*. The training dataset had about 175 million events from $4{,}000$ players. There were two test datasets containing an additional $3{,}000$ players each. While the players in the training data were observed over 40 days, the players in the test datasets had an observation time of 56 days. While the chemistry lab in Sec. 4.5.1 had only 21 different events, we were able to extract about 80 different event types and 75 event properties from the BnS data. Tbl. 2 summarizes the BnS data.

Dataset	Time Period	Weeks	Number of Gamers
Training	2016/07/27 - 2016/09/21	6	$4{,}000$
Test Set 1	2016/07/27 - 2016/09/21	8	$3{,}000$
Test Set 2	2016/12/14 - 2017/02/08	8	$3{,}000$

Table 2: Blade & Soul trainings and test data.

In contrast to the virtual lab data, the first step was analyzing the data and transferring it to the ENVISAGE format. The data provided by *NCSOFT* differs substantially form the virtual lab data which is directly tracked through the GTM tracking integration. The BnS data also contains more event types and properties which allowed us to build new kinds of features. For example, social interactions within the game were given in the dataset. While recency and frequency matter a lot in predicting at-risk students or churn in general, social interactions were a new kind of information which we did not have in the virtual labs. We found it particularly interesting to engineer social features because this also connects to the social presence as described in [11], which is also used by *Moodle* for their at-risk student prediction. Similar to the pedagogical perspective in virtual labs, certain game characteristics have to be taken into account when designing features for games. For that reason, the feature engineering process was done in an iterative fashion. This includes discussions about features and the game concept, and checking the importance of new features. Therefore, we tried to mainly make use of algorithms that provide information about feature importance. Nevertheless, we did not limit ourselves to such algorithms and also tested ANNs. One should note that many of the newly built features are also well suited for educational settings. Similar to the chemistry lab, the features described in Sec. 4.2.1 were also used for the churn prediction of the BnS players.

The CIG challenge was a great testbed to validate if the churn prediction or at-risk student prediction can be used on large datasets and to show that the algorithms are capable of producing meaningful output in an additional setting. In the end, the GIO platform was ranked among the top five results in the competition with 13 results submitted in total. As an additional outcome, the participants were asked to contribute to a joint paper about the results and used methods within the CIG challenge. This paper is currently under submission and we refer to [19] for more details. GIO, representing the ENVISAGE consortium, participated in this publication and described its work on churn prediction.

The results provided further insights that will help to improve future work on the at-risk student prediction. One observation was that the algorithm did not heavily weight features based on the social network within BnS. Regarding the social presence, one could assume that social features should have a stronger impact and are very important from the pedagogical point of view. The social graph that was created for BnS had about $32{,}000$ players. However, we only had information about connections within the network for $4{,}000$ players. This data leads to an incomplete and very sparse graph which is rather uncommon. For further work on social features, a complete social network

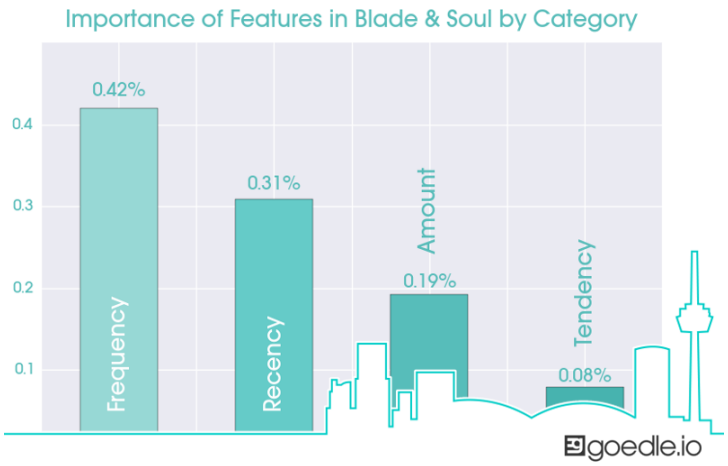


Figure 9: Feature importance for churner in Blade & Soul.

would be necessary. We can transfer this observation to the educational setting as an important insight. If one wants to take collaboration among students into account, the complete social network needs to be tracked within the virtual lab.

After learning a model based on the data described above, one can again analyze the importance of different features. Fig. 9 gives the relative importance of the features used in the BnS case study grouped by different types. Here, "Frequency" represents features that count events. "Recency"-features measure the time since a particular event has occurred or the time between two events. "Amount" groups features which depend on values attached to events. For example, the amount of virtual money spent. Lastly, features in the "Tendency" group indicate an increasing or decreasing level of engagement based on curve fitting. As described above, the BnS dataset provides a rich set of events which allowed us to create a large amount of features. Many of these features can also be used by the prediction of at-risk students. For example, the social features that represent interactions with other players or the frequency-domain feature that represents regular recurring usage.

While the f1-score in the chemistry lab case study was very high (0.96), we were only able to reach an f1-score of 0.58 in the BnS case study. However, one should also not that the winner of the CIG data mining competition reached an f1-score of 0.62. This highlights that the problem is quite difficult and huge improvements in the f1-score cannot be achieved easily but instead require a lot of effort on feature engineering and algorithmic design.

Besides the differences in the two datasets, i.e., chemistry lab and BnS, we were able to use a large intersection of features for both case studies and run the data through the same pipeline as depicted in Fig. 5. By doing so, we were able to validate the performance and capabilities of the infrastructure, resulting in predictions for the chemistry lab and the BnS dataset. On the one hand, we could show that we are able to learn an at-risk student model and on the other hand, we are able to solve a very similar task at a much larger scale. This underlines not only that the pipeline for predicting at-risk students is fully functional but it also shows with respect to the results of the CIG challenge that the work of the ENVISAGE project is highly competitive. The work on research and development in the past months shows to be compatible with other research domains and is applicable in interdisciplinary settings. As described at the beginning of the section on supervised learning, the two case studies also highlight the similarities between educational data from virtual

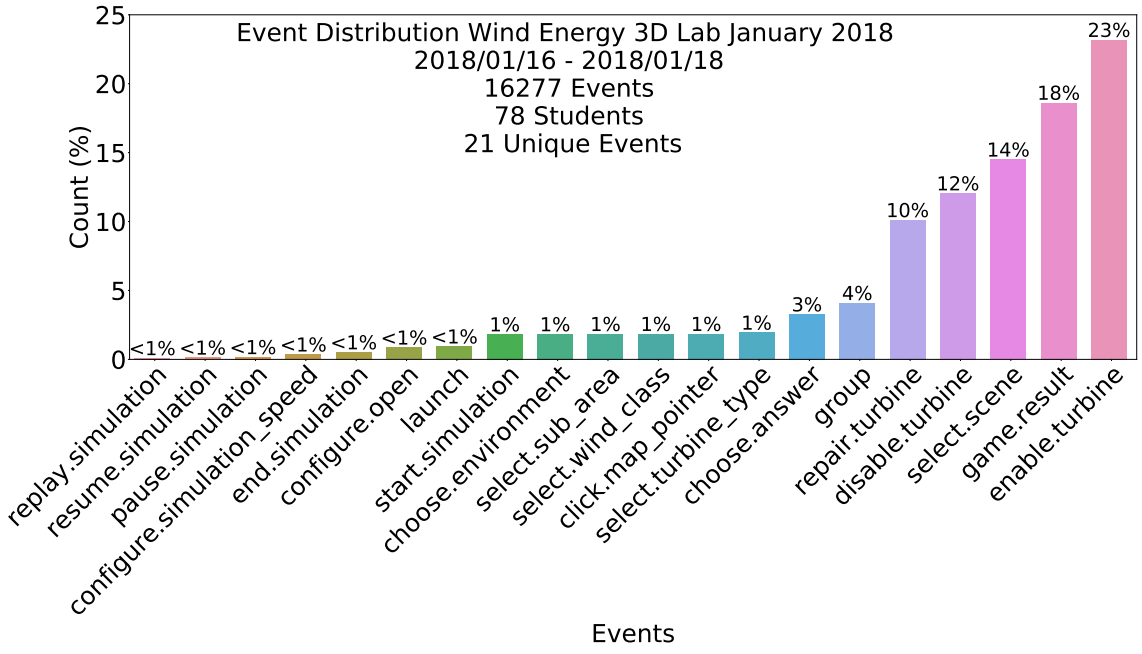


Figure 10: Event distribution of the Wind Energy Lab dataset.

labs and behavioral data from games.

4.5.3 3D Wind Energy Lab

During the pilot execution at EA in January 2018, 78 students used the 3D version of the Wind Energy Lab. The infrastructure was able to track $16{,}277$ events in total over on three days. This included 21 unique event types. Additionally, $918{,}409$ events with `game.state` information were tracked. For a better understanding of the dataset, Fig. 10 gives an overview about the event distribution. Similar to Fig. 7 in Sec. 4.5.1, Fig. 10 shows the relative usage of each event. In the 3D Wind Energy Lab, the event $enable.turbine$ is used most frequently with roughly 23%. Similar to the observations in the chemistry lab, this indicates that the lab concentrates on a particular aspect and triggering the associated event is central to the usage of the entire lab.

The application of deep analytics at the 3D Wind Energy Lab[20] comes in the form of supervised learning and in particular, ANNs. ANNs are chosen because of their wide adoption in modern machine learning applications, their supreme performance in supervised learning tasks and their capacity to approximate any given function with high accuracy (a qualitative feature widely known as universal approximation).

Given the different nature and increased complexity of the 3D version of the Wind Energy Lab and the overall aim of educators (D1.1 [32]) to predict the travel path (or learn ability performance) of learners, we devised the following supervised learning approach. The ANN we employ considers the following *input vector*:

- The *game level* where the learning exercise takes place, split by map/area (e.g., mountains) and map pointer (sub-area within the map). Each of these variables (map/area and subarea)

[20] `http://160.40.51.48/games/energy`

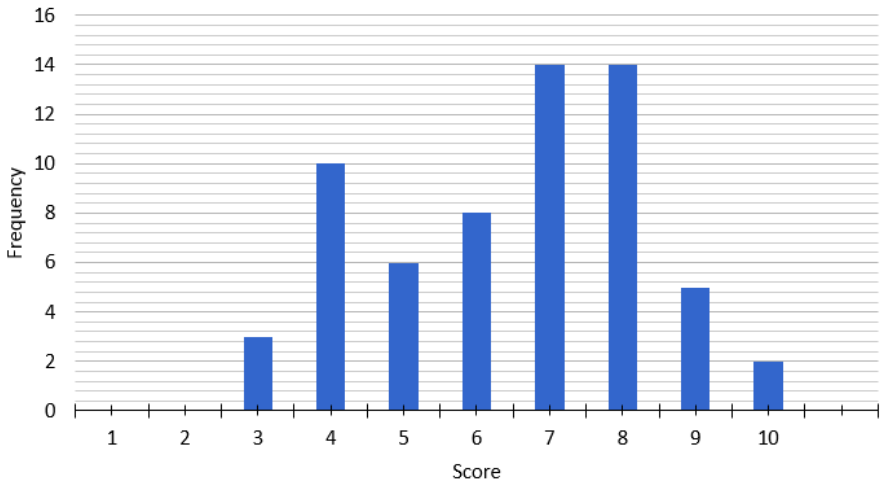


Figure 11: An example histogram of scores at the 3D Wind Energy Lab.

are identified by an integer ID which is transformed for the input vector as one-hot encoding. The subvector of inputs for the game level is thus, for example: 0,0,0,1,0,0,0,0,0,0,0,1,0,0 (the first five digits are for the map, which has an ID of 2, and the last 9 digits are for the map pointer, which has an ID of 3).

- The power, cost, and area coverage of the *chosen turbine* to be used in this game level and for the purposes of this exercise. These 3 values are normalized between 0 and 1, via min-max normalization considering all currently authored turbine values in the 3D Wind Energy Lab. The subvector of inputs for the chosen turbine is thus, for example: $0.417, 0.974, 0.75$.

Based on the above input the ANN *outputs* (attempts to predict) the 4 PISA categories of student performance based on the *score* metric as described in D2.4 [12]. As a reminder, the score represents a mastery index metric, which is ad-hoc designed by expert educators and designers of the Wind Energy Lab. The score metric is based on a combination of features on the simulation itself and a multiple-choice answer post-simulation. The student's score may vary between 1 (lowest possible performance) and 10 (highest possible performance). Fig. 11 illustrates an example of a score distribution (illustrated as a histogram of scores) at the 3D Wind Energy Lab. The 4 PISA categories are derived as follows and define the 4 outputs the ANN predicts:

- **III:** Reflective/communicative — Score: 8, 9 and 10
- **II:** Advanced — Score: 5, 6, 7
- **I**: Beginner — Score: 2, 3, 4
- **<I**: No problem solver — Score: 0, 1

The ANN may use a varying number of architectures depending on the data size available. The most promising results have been achieved with architectures of one (or none) hidden layer consisting of few neurons (see Fig. 12). All neurons of the ANNs in the final demonstrator of ENVISAGE employ a logistic function. The ANNs are trained on the dataset available (as described earlier) through standard backpropagation.

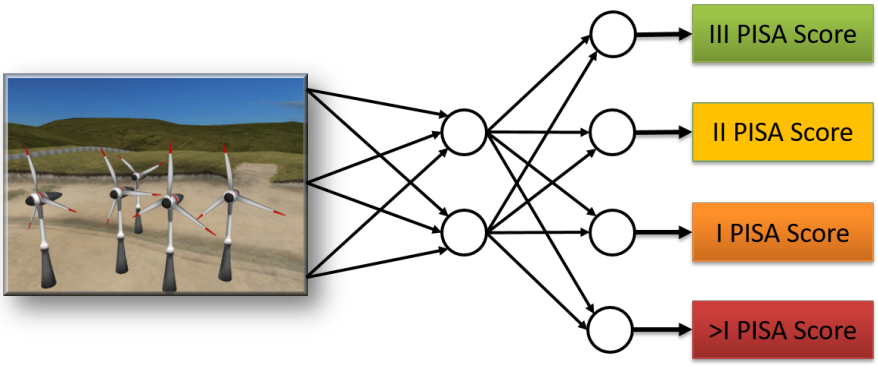


Figure 12: The ANN approach adopted for predicting the level of the learner's competence (PISA score distribution) at the 3D Wind Energy Lab. The ANN maps in-game features to the score distribution (4 score classes according to the PISA 2012 classification).

The cluster membership distribution as it is obtained from the ANN (**<I** to **III**) distribution is reported back through the analytics service to the visualization front-end. An educator that completes a new virtual lab using the 3D authoring tool is presented with this visual analytics information at the end of her design. The pie chart shown below in Fig 13 displays the predicted PISA classification distribution (ANN output) given the choices the teacher made during the authoring process (ANN input).

The implementations used to experimentally realize the supervised models described in this document can be found at the following URL:

`https://github.com/Envisage-H2020/Analytics-Server`

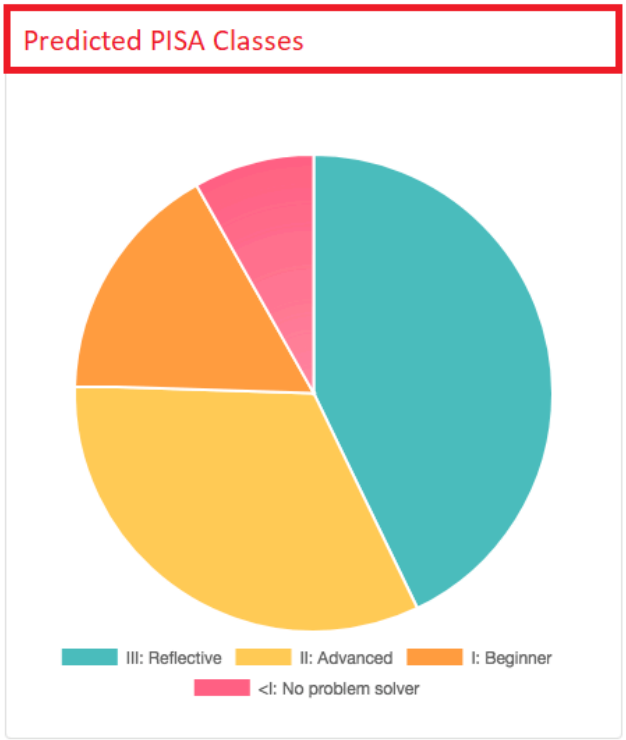


Figure 13: Similarly to the 2D Wind Energy Lab deep analytics solution, the four PISA clusters (four ANN outputs) are depicted as a pie chart in the visual analytics front end of the 3D Wind Energy Lab of the authoring tool. For more details about the visual analytics service please refer to D2.4 [12].

5 Adaptation of Learning Material

The previous section described how supervised learning can be used to identify at-risk students. However, identifying such students is one thing. Recovering these students and keeping them engaged is far more difficult. Students leaving a virtual lab may have various reasons. One reason might be that either too little or too much is demanded from the students. In this case, we can try to adapt the content in such a way that it better fits the needs of the students.

5.1 State-of-the-Art

When looking at state-of-the-art approaches, we differentiate here again between academic approaches and companies applying similar ideas in the industry. Research has been investigating the adaptation of learning material or personalized content in general a lot earlier before companies have started to integrate such approaches into their products. As in the previous examples and within the entire ENVISAGE project, it makes sense to first have a look at the development in games and then compare it to the state-of-the-art in education.

5.1.1 Academic Research

Although there exist different angles for content adaptation in games, one of the common approaches is to adapt the difficulty in games. The seminal work by Hunicke and Chapman from 2004 [17] describes the *Hamlet* system for adjusting the difficulty dynamically in *Valve's Half Life*. *Hamlet* analyzes player behavior and adjusts the games accordingly to control the game difficulty. There are also more recent papers such as the work by Xue et al. from 2017 [36]. Xue et al. try to optimize a player's engagement throughout the entire game by using probabilistic graphs in level-based games by *Electronic Arts*. While the work summarized so far, focuses on adjusting games in general, there is also work on adjusting opponents in games. For example, Olana Missura's dissertation [26] presents an universal framework for games where players have interactions with opponents. Here, the skill level of an opponent can be adapted to match a player's skill.

When it comes to EDM, different techniques have been employed to personalize learning. For example, collaborative filtering [8] has been used to to suggest learning material. Other approaches go even one step further and try to design entire courses or study plans in a data-driven way [1].

5.1.2 Industrial Approaches

In gaming, companies such as *deltaDNA* offer consulting on game balancing[21]. In many cases, this is more oriented towards monetization than players' performance or even skill improvement. For example, a game developer may not be interested in causing a player to solve all levels as quickly as possible because this player will then quickly move on to a new game — possibly from a different developer or game studio.

When looking at the educational sector, this topic is often referred to as *Adaptive Learning* and different aspects are covered by this term. While changing the content is also considered to be adaptive learning, the implementation of different learning theories are also included. For example,

[21] https://deltadna.com/consultancy/

changing the repetitive behavior of a flash card system. Companies such as *WiseLab* offer systems that allow to create content based on questions and answers. This learning material is then rolled out to the learners in different forms (e.g., multiple choice questions) and on different platforms (smartphone, tablet, etc.). In order to optimize the learning progress, the order of the questions are adapted. *D2L*[22], formerly *Desire2Learn*, is an example of a company where the content for students can be adapted. They provide an LMS which offers rule-based content adaptation. One of their introductory videos[23] describes well how triggers can be set to personalize the learning experience. For example, when the system detects that students struggle to complete a test, supporting content can be provided only to those students. Other LMSs' like *SABA*[24], and *Know-How!*[25] also offer support to define learning pathways based on thresholds. Another example is *Teach to One* by *New Classrooms*[26] which promises personalized learning for math. *Teach to One* partners with schools directly and does not only focus on digital learning material but also replaces the core curriculum of a class by creating individual content for each student.

Other companies go beyond rule-based systems and employ machine learning for educational scenarios. For example, *TrueShelf*[27] offers an adaptive learning platform that lets students learn mathematical concepts by helping them to solve math problems and real-world puzzles that get progressively harder as their skills develop. Their AI-powered platform identifies students' strengths and weaknesses, and personalizes content accordingly. *Adaptemy*[28] is another example of a company using an algorithmic approach to personalize the learning experience. *Adaptemy*'s platform does not only provide a recommendation engine that takes the type of content into account but also tries to support learners by estimating their proficiency level and personalizing content accordingly.

5.2 Dynamic Difficulty Adjustment

As we have discussed already, students have different behaviors and show varying performance on learnings tasks. Therefore, we should also adapt the learning material to their needs. We should avoid to demand too much from a student but we should also pay attention to the learner not being bored. In general, we should begin with finding the best approach to teach content to an entire group or class. Afterwards, we can try to find a good pace for smaller subgroups of students, e.g. the high performing students. Ultimately, we are looking for a system that adapts the content for each student individually. Before we can adapt the content, we need to assess the performance of a student on the learning task, exercise, or challenge. At the same time, we also need to know the difficulty of a given task, in order to adapt the course material accordingly. We describe different approaches in the next section.

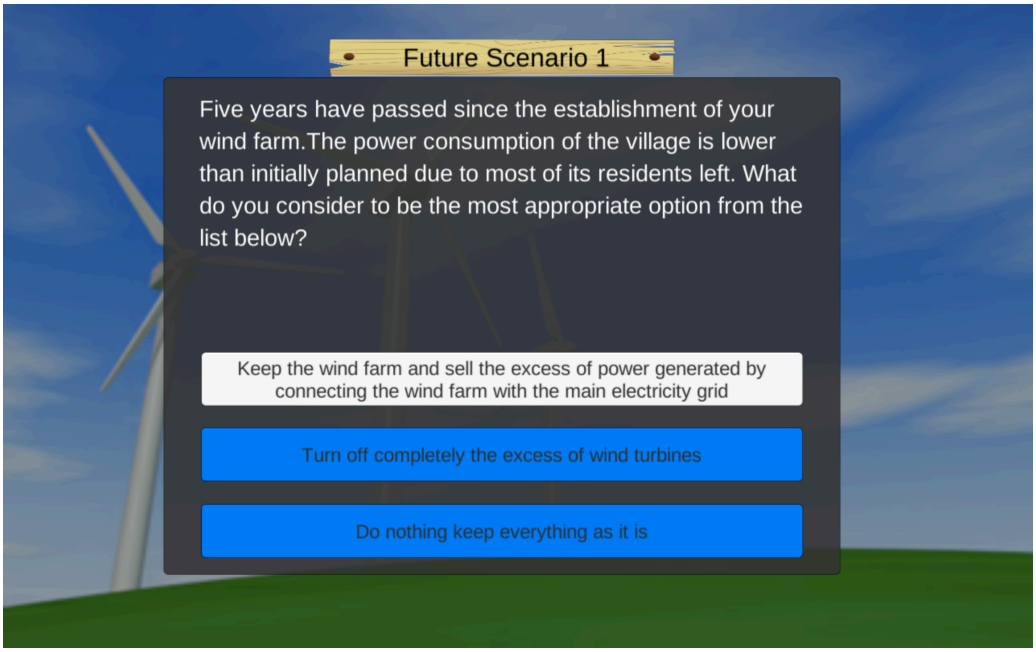


Figure 14: Multiple choice question in the 3D Wind Energy Lab as part of the grading.

5.2.1 Assessing Performance and Measuring Difficulty

Assessing the performance of a student is not always obvious and can be done in different ways. Deliverable D1.1 [32] already described how time-on-task is an important indicator in learning analytics. It was also discussed that time-on-task can be related to a student's learning performance or achievements. We have seen this connection in Sec. 3 too when we clustered students using archetypal analysis. In some cases, time-on-task can measure the difficulty of an exercise as well. For example, if students typically do not require much time for a task, one can consider it to be easier. However, this indicator does not always measure the difficulty as students may also give the wrong answer after only little time because they did not give the exercise enough thought. If we have exercises where we ask students for an answer, we would rather judge the difficulty of an exercise by the total number of correct answers for each exercise, or the average grade of that exercise. For example in the 3D Wind Energy Lab, students have to answers questions which are part of a scoring (see Fig. 14 for an example). The results are used to estimate the student's PISA proficiency level.

If we cannot easily judge the quality of an answer or the learning task, we can also consider to explicitly ask the students to rate the previous task. This could be a setting where we would need a teacher to rate every answer afterwards. For example would be, when a solution requires a free text. Here, it is not possible to immediately assess the quality of a solution. In larger settings like

[22] https://www.d2l.com
[23] https://www.d2l.com/resources/videos/personalize-learning-experience-release-conditions-intelligent-agents/
[24] https://www.saba.com
[25] https://en.knowhow.de/
[26] https://www.newclassrooms.org
[27] https://trueshelf.com/
[28] https://www.adaptemy.com

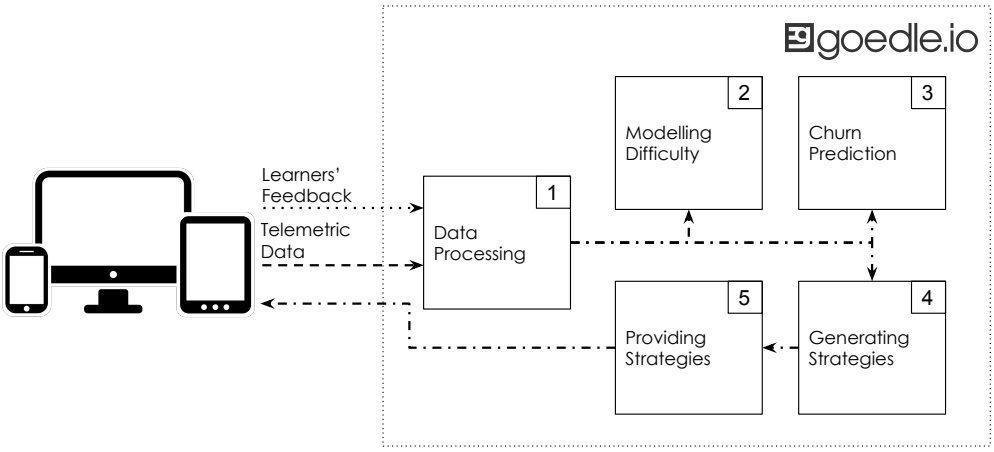


Figure 15: Infrastructure for content adaptation and dynamic difficulty adjustment.

MOOCs, it may even be completely impossible to rate each answer in an acceptable time frame. After the completion of an exercise, we can ask the students to rate the previous exercise as “easy”, “medium”, or “difficult”. Of course, other rating schemes are also possible. Fig. 15 shows the entire process how the ENVISAGE project realizes content adaptation within the GIO infrastructure. The figure also highlights how the learner’s feedback is acquired and processed (Fig. 15 (1)).

Having either a grade or explicit feedback from the learner, we can correlate this feedback with metrics such as time-on-task to estimate the perceived level of difficulty for students based on telemetric behavior. In some cases, the difficulty does not correlate with a single behavior but instead several features have to be taken into account. Having feedback and behavioral data at hand, we can use this data to build a machine learning model that takes as input the tracked data and the grading or perceived difficulty of a student as labels (Fig. 15 (2)). Based on the behavioral data, the learned model predicts how difficult a new task is for a student or predicts the estimated performance of a student on a new task in advance. The case study on the 3D Wind Energy Lab in Sec. 4.5.3 also gave an example how a machine learning model can be trained to predict students’ performance. This approach has several advantages:

- By doing so, we can learn general behavior that correlates with more or less difficult tasks and exercises.
- We can add new tasks and exercises in the future, and use our model to get an idea of the difficulty of each new one.
- We get rid of the requirement to ask the learner for explicit feedback. These request for feedback may annoy the learner and cost time.

Depending on the amount of users, we do not even have to ask each learner to rate each tasks. Instead, we can generalize from a smaller number of students and we do not have to bother each learner. Again, this is in particular interesting when looking at MOOCs with thousands of learners.

5.2.2 Designing Learning Strategies

After measuring the difficulty of an exercise and building a model to judge different exercises automatically, the third step is to design different learning strategies (Fig. 15 (4)). Here, a learning strategy can have a variety of different forms. For example, in the case of the Wind Energy Lab, a strategy may be an initial setting of the environment. Certain parameter configurations make the problem easier for the students because they have to do fewer changes in order to generate the proper amount of energy or income. In the chemistry labs, the strategies may look different. For example, the Molecule Construction Lab[29] asks students to build molecules. In its original form, the student picks a molecule from a given list, solves the current task, and proceeds with the next molecule. One can also think of a version of this lab where the students cannot pick the molecules themselves but instead the order is given by the lab, i.e., by the teacher. Here, different strategies can order the molecules differently. For example, from easy to hard, or vice versa. We have implemented this version as a use case in Sec. 5.3.1.

5.2.3 Automated Strategy Design: Genetic Algorithms

The examples of the previous section require expert knowledge from the teacher to define different strategies. In some cases, the space of all possible strategies is way too large, to manually define and test all strategies. In such settings, machine learning algorithms can be used to define new strategies automatically. In particular, we have started to look into *Genetic Algorithms* to create new strategies.

Genetic algorithms allow to automatically construct new strategies based on existing ones. Inspired by the process of natural selection, genetic algorithms find solutions to search problems in an iterative fashion. Typically, genetic algorithms start by generating a few random solutions. In the current setting, we prefer to have an educator generating initial seed solutions because the educator typically has a good intuition how a "good" strategy may look like. In each iteration, the genetic algorithms pick a few existing strategies from the pool of all available solutions to construct a new generation. Strategies that already perform well, are more likely to be selected to construct the next generation. The performance of a strategy is evaluated based on a *fitness-function*. In our setting, the fitness-function can be the average performance achieved by the students who learned according to a strategy. The construction of a new generation is based on simple permutations and modifications of the current generation. Afterwards, the new generation is then evaluated again based on the fitness-function. This process continues until a quality threshold or a maximum number of iterations has been reached.

However, in the case of educational settings, this approach comes with additional constraints and challenges. Here, we have to be very careful with random modifications. The ethical requirements do not allow us to test strategies completely at random because students may suffer from bad random strategies. For example, think of a strategy that chooses all parameters to be a the most difficult setting. To a genetic algorithm this may look like a total valid strategy but a teacher would never pick it manually. Although the results would quickly indicate that this strategy is not desirable, our ethical obligations do not allow us to test such a strategy. Furthermore, the fitness-function cannot be evaluated easily in this setting, as we first have to find students to evaluate the new generation

[29] `http://www.envisage-h2020.eu/games/chemistry/lab_molecule_ionic_covelant_bonding/Molecule_IonicCovelantBonding.html`

on. Additionally, we have to make sure that the difference in quality is significant and not just a probabilistic artifact. For the latter issue, we describe appropriate tests in the next section.

While genetic algorithms can easily generate hundreds or thousands of strategies, we also need a sufficient amount of students to validate the quality of each strategy. For that reason, school settings may be less adequate for this approach but MOOCs show a lot potential for this automated process. The next section will describe in detail how strategies can be compared and tested.

5.2.4 A/B and Multivariate Testing for Learning Strategies

If we have multiple strategies at hand, we want to compare those strategies. As we have described above, the performance of a single strategy is typically the average score over a group of students. For that reason, we have to assert that the difference in performance of two strategies is statistically significant and not only due to some extreme outliers. E.g., a few students achieving extremely good results by cheating. If we have two strategies at hand, we can compare them via *A/B-Testing* [20]. Here, it does not matter how a strategy was constructed by genetic algorithms or manually by a human. A/B-testing allows us to pick the more promising alternative of two strategies.

Running an A/B testing experiment on two strategies amounts to a statistical significance test. Typically, we assume a significance level of 95%. This means that one can be 95% confident that the winning strategy is really superior. Nevertheless, there is still a 5% chance that the result is only due to a random chance.

Depending on the nature of our experiment, different tests needs to be used. For example, when the performance is measured by the number of students that pass a test, we have a binomial distribution and should be using a *Chi-square* test. In other cases, where the performance indicator is normally distributed, a *t-test* is what we are looking for. The performance data may be normally distributed in the case of timings for a particular task. However, in a proper setting, we first need to validate the distribution of the data. In other cases, for example when we count the number of correct answers, the data is strictly speaking not normally distributed but may be *Poisson* distributed. Nevertheless, it is also known that the normal distribution is a limit of the Poisson distribution for large mean values. Still, other tests such as the *Wilcoxon-Mann-Whitney* test are more suitable in such cases.

We often have more than two strategies that we want to compare. The most obvious thing to do, is performing pairwise tests. However, this approach will increase the likelihood of false positives. As described above, there is always a 5% chance of the winning strategy being inferior when assuming a significance level of 95%. Now, doing several pairwise tests increases this chance. For that reason, there exist other approaches to compare multiple outcomes such as *Analysis of Variance (ANOVA) F-tests*.

In general, A/B testing comes with some additional disadvantages. For example, we need to specify the number of students in advance who will have to learn following the different strategies. This may have the undesirable effect that the inferior strategy is used on many students who suffer from lower quality teaching. For that reason, we will explain *Multi-Armed Bandits* in the next section, which avoid this disadvantage.

5.2.5 Multi-Armed Bandits for Optimization

Instead of using pairwise tests or other multivariate testing frameworks, *Multi-Armed Bandits (MABs)* also allow us to compare several strategies at the same time and also provide a mechanism to iteratively pick the best performing strategy among all available ones. MABs are inspired by gamble machines in casinos, i.e., the arms of the bandits. This setting assumes that there are multiple slot machines in a row with random rewards. The player has to decide, which machine to play in order to maximize the reward.

We can now view each learning strategy as a "one-armed bandit" and the reward is the performance of a student. We want to find the strategy that maximizes the performance for as many students as possible. If the reward of each strategy was known, the task was trivial. Without this knowledge, we have to try different strategies and track the rewards. A very simple approach would be to choose the strategy with the current best expected reward. However, this yields in the "exploitation vs exploration" dilemma. Some strategies that we have not tested yet, may yield even better results, or some strategies may just look bad after just a few initial tries due to random effects.

The goal of a bandit algorithm is now to find an approach that plays the optimal strategy exponentially more often than any other strategy. One instance of an algorithm that solves the multi-armed bandit problem, is the *Upper Confidence Bound (UCB)* algorithm [3]. We will not give full technical details here, however, the algorithm calculates a score for each strategy that trades off exploitation and exploration in each iteration. Depending on this score, the next strategy is picked. Another popular alternative to UCB is *Thompson sampling* [7]. Thompson sampling achieves state-of-the-art results while being vary easy to implement.

In our setting, we do not calculate the score for each student, i.e., in every single iteration, but change the distribution over all strategies frequently. I.e., we begin with a distribution where all strategies are distributed uniformly and then adapt this distribution as we gain more insights on which strategies perform well. By constantly changing the distribution, we avoid the problem from A/B testing where we have to determine a fixed number of trials per strategy in advance. Therefore, there will be a lot fewer students that receive a suboptimal strategy in many cases.

5.2.6 Personalization of Strategies

The MAB approach to find an optimal strategy has one disadvantage when talking about personalization of learning material and virtual labs: it tries to find an optimal strategy across all students, i.e., it does not find strategies for different groups of students. However, it is very likely that not all strategies are equally well suited for all students. For example, some students may require a slower pace at the beginning than others.

For that reason, one future extension of this approach is to segment students into different groups and to find optimal strategies for the different groups. One grouping of the students could follow the PISA levels of proficiency. This approach has also been propose in D1.4 [24], Sec. 2.1.3.

Another approach could use the unsupervised methods presented in D3.1 [16], Sec 5.1, where students were automatically clustered into groups based on their behavior. Other approaches could first use a prediction of at-risk behavior and group the students depending on their at-risk likelihood. However, here one has to be careful. The classifier could also detect well performing students as potential churners, as they have already learned successfully and are in danger of leaving as the

demand is too little for them. For that reason, one has to carefully craft the adaptation of the content.

Eventually the vision is to have segments of size $n = 1$, i.e., every student gets an individual learning strategy. Taking this even one step further, we can use *Reinforcement Learning* to learn a model of an agent representing a teacher that dynamically adapts the content for each student on a finer level.

5.2.7 Closing the Loop: Reinforcement Learning

Before describing how *Reinforcement Learning* can be used to learn a model of a teacher, let us denote that the multi-armed bandit problem can be seen as one of the most simple reinforcement learning problems or a precursor of reinforcement learning. After each pull of an arm, the reinforcement learning algorithm tries to find the optimal next action that maximizes the rewards. I.e., pulling the same arm again or a different one. In the context of reinforcement learning, one refers to a *policy*. By using a method called *policy gradients*, a policy for picking actions is learned. Currently, it is very popular to use ANNs and *Deep Learning* within the policy gradients approach.

An interesting possibility of reinforcement learning is to learn a model, or an agent, that simulates a teacher. Previously, we defined entire strategies for a virtual lab in advance. For example, in the case of the chemistry lab, we defined all actions in advance. E.g., in one strategy molecule A would always follow molecule B and in another strategy this would possibly happen vice versa. However, personalizing the entire learning experience would not define the next molecule in advance. Instead, the student would solve one challenge and the teacher would then pick the next one matching the current level of proficiency of the student. I.e., an action is the change in course or leaning material, or adjustments to the environment of a virtual lab. For the chemistry lab, this would amount to learning a policy that picks the next molecule based on the previous molecules and the behavior of the student. Here, the reward is the behavior or performance of the student. In deliverable D1.3 [33], the ideas and advantages of active learning are further motivated, in particular from a pedagogical point of view. Reinforcement learning can be used as a technology to improve and advance the approaches to active learning. Lastly, one should note that this setting distinguishes from supervised learning as described above, as we do not know in advance how the next challenge affects the learning behavior of the student. Instead, the student has to solve upcoming challenges and based on the performance, we learn if this was a good design of learning and course material.

This approach has not been implemented yet and is left for future work. The approach also comes with several challenges. We need a large set of students and evaluations so that we can learn a reliable model. The feedback from the students does not necessarily come immediately. For example, we may design a virtual lab with several sub-tasks. However, the students' performance is only evaluated once at the end of the lab. Such a setting, for example, is present in the 3D Wind Energy Lab. Additionally, this approach demands much higher computational power from the infrastructure. Whenever the student is evaluated, the model needs to make the next training iteration and has to update its model. Similar to the MAB approach, one can also collect evaluations in batches, which however then only approximates the optimal training procedure.

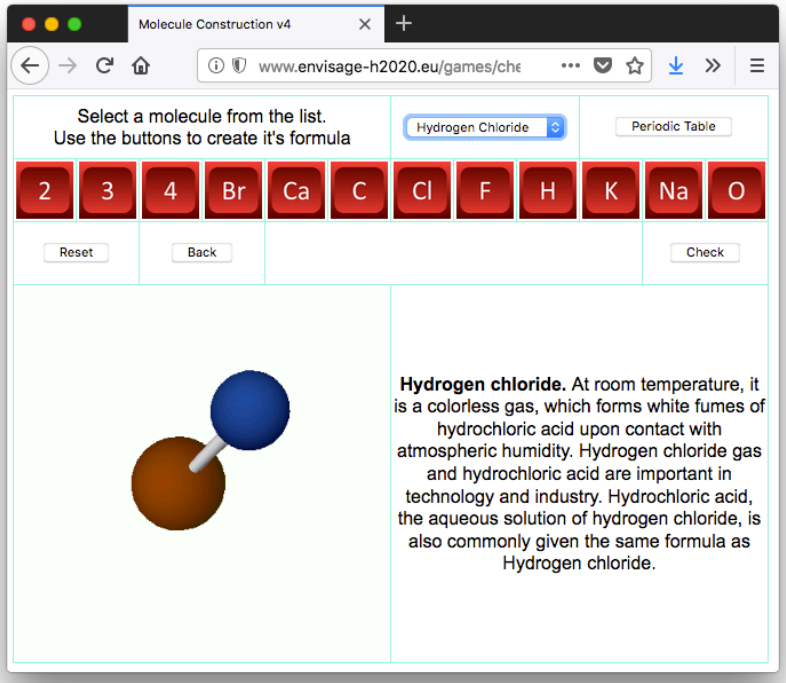


Figure 16: The original chemistry lab containing the default dropdown.

5.3 Case Studies

We will now present two different case studies that exemplify the usage of dynamic content adaptation. We will begin by presenting the integration of content adaptation in a chemistry lab. This is followed by a case study produced in cooperation with one of GIO's customers who operates a highly successful mobile quiz game.

5.3.1 Chemistry Lab

One use case for difficulty adjustment or content adaptation is the *Organic Molecule Covalent Bonding* virtual lab. As shown in Fig. 16, in its current form, the student can pick a molecule from a dropdown. After this selection, the student has to answer different questions with respect to this molecule and solve associated tasks. After all tasks have been solved, the student can pick the next molecule. More information about the 2D Chemistry Labs can be found in deliverable D1.1 [32], Sec. 6 and the lab is still available online[30].

We have now modified the lab in such a way that the order in which to solve molecules is determined by the teacher[31]. The source code also be found in ENVISAGE's GitHub-repository[32]. Each of

[30] `http://www.envisage-h2020.eu/games/chemistry/lab_molecule_ionic_covelant_bonding/Molecule_IonicCovelantBonding.html`

[31] `https://envisage.goedle.io/dda/examples/chemlab/Molecule_IonicCovelantBonding.html`

[32] `https://github.com/Envisage-H2020/lab_molecule_ionic_covelant_bonding/tree/gio/content_`

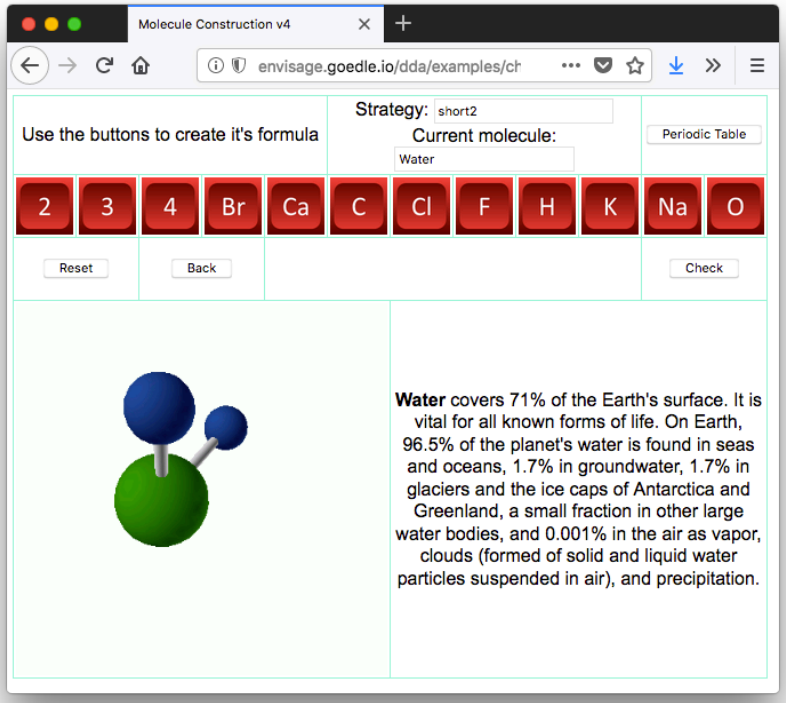


Figure 17: The new chemistry lab where a molecule is picked based on a strategy obtained from the ENVISAGE API.

such orderings is what we considered a learning strategy in the description of the approach in the text above. The teacher now defines different strategies in the authoring tool and once a student starts the lab, a random strategy is assigned by querying the ENVISAGE API. By doing so, teachers can test if students stay longer engaged if easy molecules are followed by difficult ones. Or, if a good ordering should contain more or fewer difficult molecules because an ordering may contain the same molecules more than once. Accordingly, the adapted chemistry lab looks as depicted in Fig. 17.

This implementation has only been made available shortly before the submission of the deliverable. For that reason, we do not have enough data available to measure the impact of a possible adaptation and we cannot say yet which strategy works best. However, we are aiming at integrating this content adaptation into the next pilot study, in order to obtain more behavioral data from students and feedback from teachers. While the dynamic content adaptation is currently only integrated in the 2D version of the lab, we are also working on integrating the same mechanism into the 3D version of the chemistry lab, as well as the Wind Energy Lab.

Right now, the student will always receive a new strategy when the virtual lab is loaded. However, in the future one could also consider storing the current strategy as long as not all molecules of a strategy have been solved. In that case, we would also require a skip button, so that too difficult molecules can be skipped and students do not leave due to insurmountable obstacles. Tracking the usage of the skip button would also be an interesting behavioral datapoint. Its analysis could

adaptation/

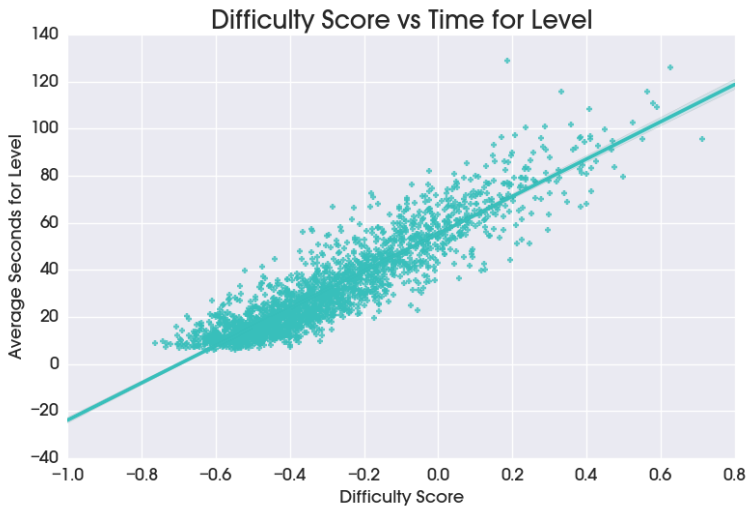


Figure 18: Correlation between time to solve a level and the player feedback regarding the difficulty.

additionally support the design of new strategies. If we proceed with this approach, we can also implement more sophisticated approaches to assign a follow-up strategy. E.g., if students perform well, they will receive a more difficult strategy afterwards.

After a sufficient number of students have used the different strategies, we can start to evaluate the different strategies based on various performance indicators. For example, which strategy lead to more correctly solved molecules? Which strategy had students engaged for the longest amount of time? Similar to the case study for at-risk student prediction, the implementation for the chemistry lab is at a prototype stage and there was only limited data available as of February 2018. Therefore, we now provide another case study in the gaming sector where the same system was used to dynamically adapt the difficulty of a mobile quiz app.

5.3.2 Mobile quiz Game

One of GIO's customers runs a successful mobile quiz game. This game is divided in hundreds of levels and with the approach described in Sec. 5.2, we helped the quiz game to find an improved ordering of their content. Opposed to educational apps, their KPIs may be different but the technical approach remains very similar. The results presented here, were first published on GIO's blog in November 2017[33].

From the surface it was not obvious which levels in the quiz game were more or less difficult, as players could skip levels by using jokers. Therefore, we needed to measure the perceived level of difficulty by the players, before we could analyze the relationship between the customer's KPIs and level difficulty. For that purpose, GIO's infrastructure supports the tracking of player feedback. After the completion of a level, the app simply asked the player to score the previous level. In a simple setting, one can just ask the player to rate the level as "easy", "medium", or "difficult". This data is then used to calculate a score for each level.

However, one does not want to ask every single player to rate every single challenge as this will

[33] http://blog.goedle.io/2017/11/29/increase-ad-revenue-by-74-with-difficulty-adjustment/

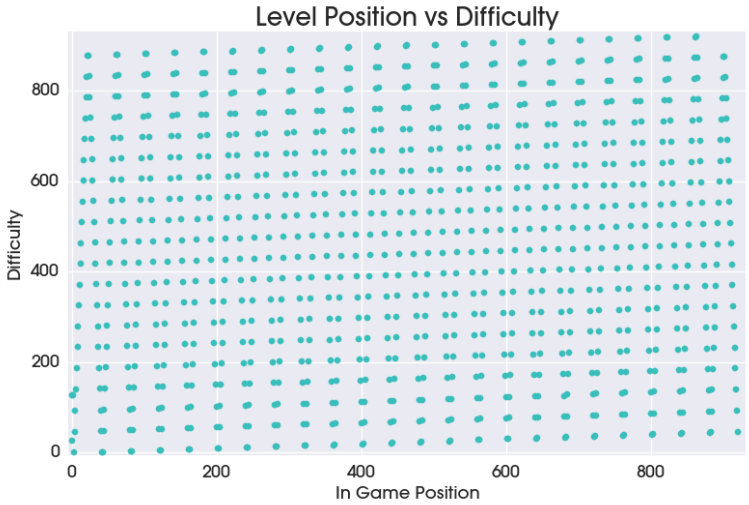


Figure 19: An example strategy that increases and decreases the difficulty in a smooth manner to diversify the user experience.

annoy the user and possibly lead to churn. For that reason, we analyzed the data in detail and found out that the player feedback correlated very well with the time it took to solve a level. Such an observation is not rare and can be found in educational scenarios as well. For example, we have seen in Sec. 3.2 as well that time-on-task is a good indicator of performance and learnability. Fig. 18 shows data from more than $2{,}000$ different levels. In total, $750{,}000$ level completions were taken into account from roughly $60{,}000$ players.

Once we have a function to estimate the difficulty for all levels which only depends on behavioral user data, such as time-on-task, we can get a better understanding of how the user journey looks like in terms of the level difficulty. We can now test different strategies and measure their impact on the KPIs or use Multi-Armed Bandits to find the best strategy directly. The result of each test also gives new ideas on designing additional strategies. One example of a strategy could be the one depicted in Fig. 19. The strategy in Fig. 19 was designed in such a way that the level difficulty increases with every level for a certain number of levels before it then decreases again for the same number of levels. Users who want to be challenged right away might find such a strategy more appealing than the initial one. We can test dozens, hundreds, or even thousands of such strategies depending on the number of players available.

In (mobile) games, revenue is typically the most important KPI. By testing various different strategies, we were able to improve ad revenue for the mobile game mentioned above by 50% after 7 days and 74% after 14 days compared to the initial baseline. One should mention that this was all possible by only operating on the macro level and we have not started to group users into segments yet. As we have seen, measuring and analyzing the difficulty level has several benefits and applications within education sector and beyond. It helps to optimize the retention or monetization in mobile games but can also be used to optimize other KPIs depending on the nature of the app or students' performance in virtual labs.

6 Demo

We will now describe two demonstrators that include the prediction of at-risk students and the content adaptation from the sections above. We will begin with the prediction of at-risk students and then describe how the content adaptation is managed from within the authoring tool. Most of the functionalities shown below are accessible through the authoring tool at:

`http://160.40.50.238/envisage/wpunity-main/`

To login, a test account has been created with the username “author” and the password “reviewerenvisag”.

6.1 Prediction of At-Risk Students

To demonstrate the training of a model for the prediction of at-risk students, we have prepared a web-service where raw tracking data can be uploaded. This data is then analyzed and preprocessed to be used for the model learning. If the target app already uses the ENVISAGE infrastructure to track behavioral data, the ENVISAGE API can be used to download raw data for specific days. We also provide a helper script to directly download raw data for an entire time span and to merge multiple days into a single file. This script can be obtained from ENVISAGE’s GitHub-repository[34]. The final dataset has to be a *JavaScript Object Notation with Padding (JSONP)* file. The JSONP file contains one *JavaScript Object Notation (JSON)* dictionary per line. The dictionaries require the following mandatory fields to be used in the demonstrator:

app_key Identifier of the virtual lab

user_id Unique identifier of a learner

event Event which was triggered by the learner

ts Unix timestamp that indicates when the event was triggered

A more detailed description of the fields can be found in D2.1 [13], Sec. 4.1. D2.1 also contains a description about the data types and which additional fields can be used. The following code snippet represents a single dictionary of the JSONP file, i.e., a single line:

```
{
    "user_id": "learner_1", "ts": 1516095542,
    "app_key": "1", "event": "answer.question"
}
```

6.1.1 Data Upload View

Once the data is in the correct format and has sufficient size, it can be uploaded to the ENVISAGE backend to invoke the demonstrator. Currently, the demonstrator supports JSONP files which can be optionally compressed via *GNU zip*. If the data is compressed, the file name should end with `.gz`. The URL for the data upload is as follows:

[34] `https://github.com/Envisage-H2020/Tools/blob/master/utility_scripts/merge_api_files.py`

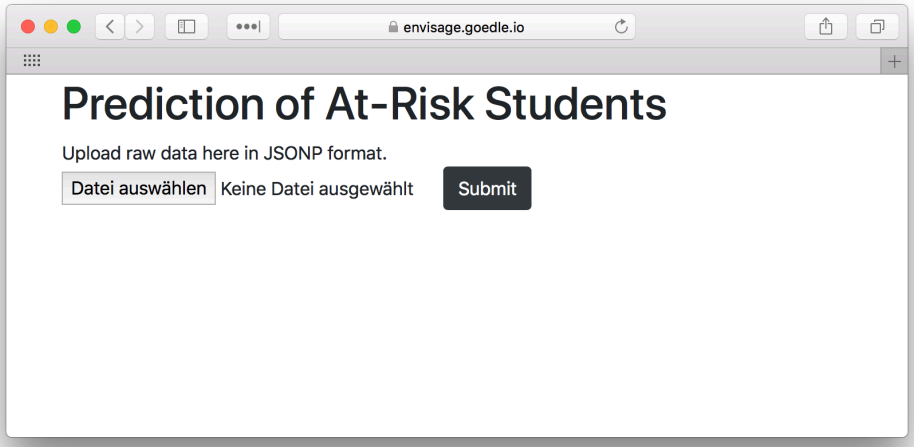


Figure 20: Screenshot of the data upload for the prediction of at-risk students.

`https://envisage.goedle.io/at-risk/upload.htm`

Fig. 20 shows the upload screen. On pressing the submit button, the data is first uploaded to GIO's servers. The data is then checked for the correct format and afterwards a new process for learning a model for the prediction of at-risk students is started.

6.1.2 Intermediate View

After the dataset has been uploaded, the user gets an experiment id and a link pointing to a result page. This step is depicted in Fig. 21. In the background, the data has to pass the entire process pipeline which was shown in Fig. 5. If users want to check results now, they can click the link. Otherwise, they should save the experiment id for checking the results later. Depending on the size of the dataset, results will be available sooner or later.

6.1.3 Results View

With the link from the intermediate page, one can access the result page. Due to the complex process pipeline, the page might not be ready yet. If this happens, one will only receive limited information and has to update the result page a few minutes later. This depends on the number of events and students in the dataset. If one did not click the link after the upload immediately but saved the returned experiment id (`exp_id`), one can also obtain the results via the following URL:

`https://envisage.goedle.io/at-risk/index.htm?exp_id=<exp_id>`

Once the results are ready, one can obtain different descriptive statistics about the dataset, e.g., the number of events and students, and information about the model for the at-risk student prediction, e.g., quality of the learned model and important features. An example screenshot of the result page is shown in Fig. 22. The results can also be accessed directly from the authoring tool. After the login,

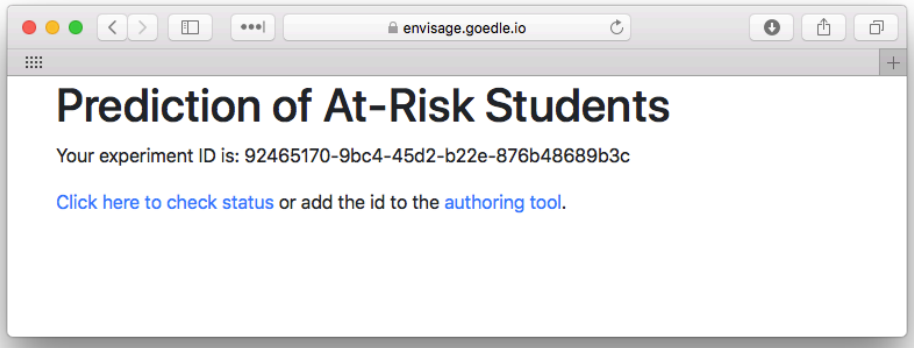


Figure 21: Screenshot after the data upload showing the experiment id which identifies the model being learned in the meantime.

one has to select an existing project. Within the project, one has to select a scene and in the next view, the at-risk student prediction appears in the menu. In summary, the results page contains the following information:

Number of Unique Events The count of unique events in the dataset. This number corresponds to the different types of events, e.g., `add.bonding`.

Number of Events This is the number of events which was uploaded from all users in the dataset.

Number of Students This is the count of students in the dataset.

Number of Churned Students The total number of students that were labeled as at-risk students in the dataset.

Timespan The time interval from the first tracked data point to the last tracked data point in the dataset.

Number of Observation Days The number of days a user is observed before making the at-risk prediction (cf. Sec. 6).

Churn Window The churn window used in the experiment (cf. Fig. 6).

Number of Sessions The total number of sessions in the dataset. Deliverable D2.1 [13] explained how sessions are calculated.

F1-Score The f1-score obtained by the model in a 5-fold cross validation (cf. Fig. 4.4 for details on evaluating machine learning algorithms).

Top Countries A list of up to five countries that were observed most often in the dataset.

Top Languages A list of up to five languages that were observed most often in the dataset.

Top Features Up to five features which have the greatest impact from a statistical point of view that lead to an at-risk behavior.

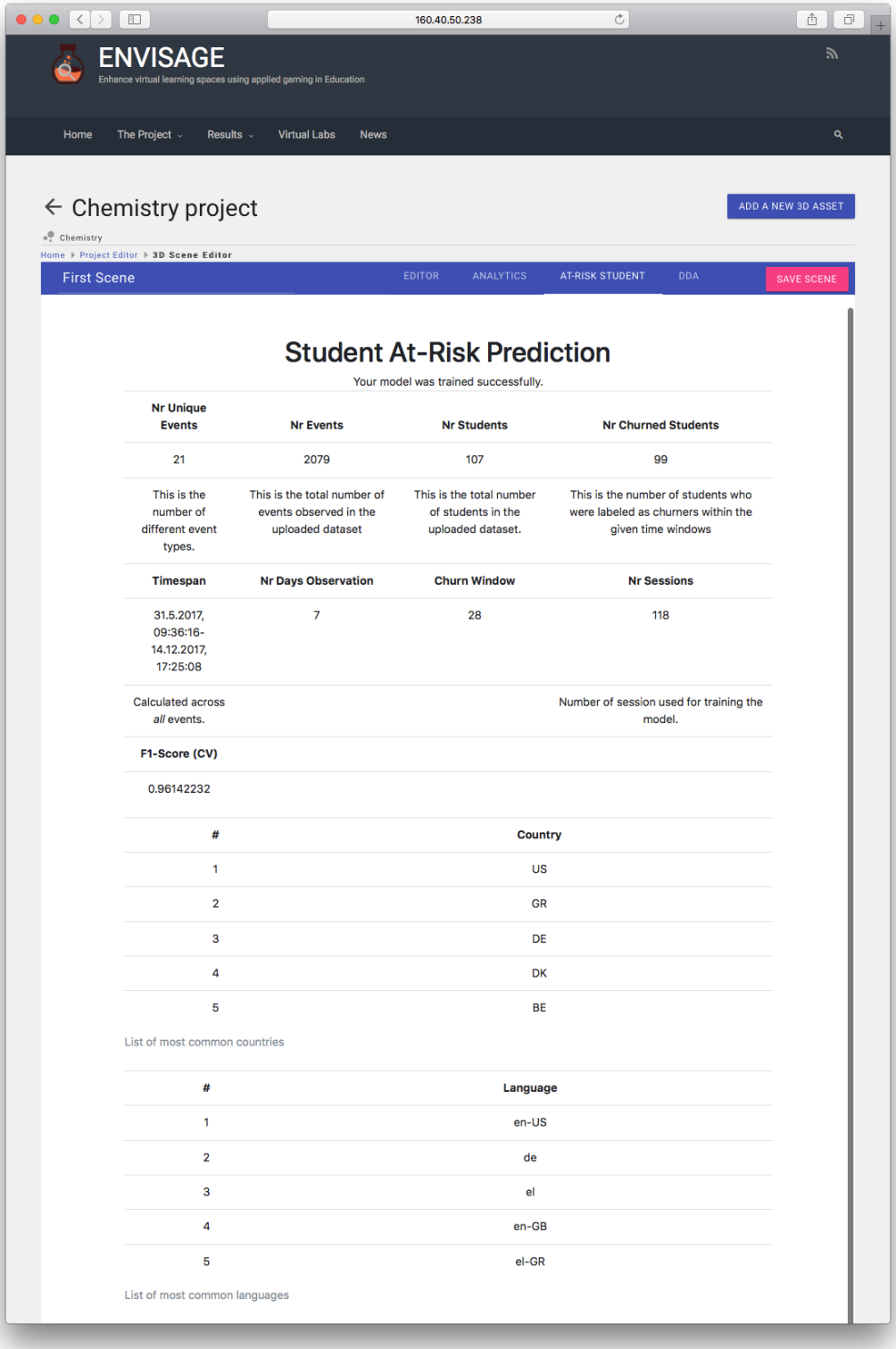


Figure 22: Result page of the at-risk student prediction.

6.1.4 Future Extensions

Right now, the insights into the at-risk behavior of students are somewhat limited and the model cannot be used easily to predict behavior of new students. In the future, we plan to improve on both of these issues.

Regarding the insights, we envision an algorithm that is capable of extracting additional and easy to digest insights from the model. I.e., rules or examples why students are, or become, at-risk. This should go beyond the single dimension that we present right now. For example, instead of just providing the information that a particular event correlates with at-risk behavior, we want to present combinations of multiple events and their characteristics. E.g., students with a **high** number of **event A** but a **low** average of **event value B** tend to have an increased at-risk behavior. When using the learned models for predictions, we also want to make sure that the quality of the model is sufficient. To get a better understanding of the model quality, tools such as ROC-curves [10] or confusion matrices [34] can support teachers as well.

There are different options to make the predictions of the model available. A straightforward approach would be to allow the user, i.e., game developer or educator, to also upload an additional dataset with the most recent students for which predictions are supposed to be made. These students would not be used for training but instead those students would be evaluated by the algorithm. The prediction for each student could be written to a result file. E.g., if one wants to use data from the last two years for training but only needs predictions for the current class which is using the lab. Another approach would be to automatically detect which students in the dataset are new and do not qualify as training instances yet. These students could be removed from the training dataset and instead be used to make a predictions. Again, a result file could be provided with predictions on those users.

Students at-risk in the next **2 weeks**		
Continue	Uncertain	Stop
45%	25%	30%

Table 3: Traffic light system for at-risk students.

One of the main problems, when it comes to interpreting future behavior, is an easy to read representation of the predictions. Without further knowledge and a concrete use case, understanding future behavior is often hard to grasp. Therefore, a traffic light system could help the teachers to directly see how the behavior is distributed among the students. We have already had great success in the past with traffic light based visualizations in marketing settings. An exemplary visualization is depicted in Tbl. 3. The next step for a teacher is to adjust a virtual lab based on insights gained from the learned models. For example, if a feature like reading the manual or answering questions leads to a reduced at-risk propensity. This closes the loop and nicely connects to the content adaptation in the next section.

6.2 Content Adaptation

We will now describe the demonstrator of the content adaptation module. Similar to the demonstrator of the prediction of at-risk students, we will explain different screens that are used within the

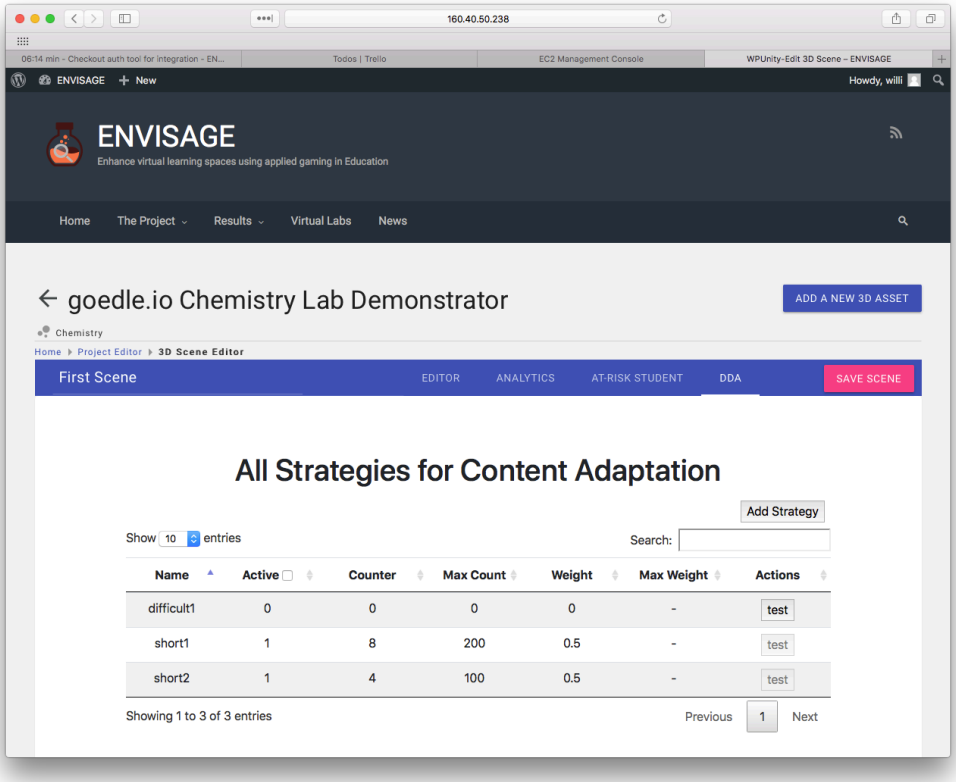


Figure 23: Authoring tool showing all available strategies for a virtual lab.

entire process. All screens can also be accessed directly from the authoring tool. After the login, one has to select an existing project. Within the project, one has to select a scene and in the next view, the dynamic content adaptation appears in the menu under “DDA”.

6.2.1 List of Strategies

The first screenshot from the authoring tool in Fig. 23 shows a list of all available strategies. This view shows all strategies with basic information, such as the current counter, i.e., the number of times this strategy has been allocated to students, a maximum value which defines the limit of tries for each strategy, and a weight that defines the probability of this strategy being returned. In many cases, a lab has a large variety of strategies and showing only the active ones is helpful. For that reason, one can remove the inactive ones from the view by clicking the checkbox next to “Active”. Clicking this checkbox leaves the user with only the currently active strategies.

6.2.2 Add a Strategy

The view shown in Fig. 24 allows to add a new strategy to the set of available strategies. It only requires a new name for the strategy and its description in valid JSON. Here, one has to be careful to enter JSON that is compatible with the particular virtual lab. In the case of the chemistry lab, an example strategy could look as follows:

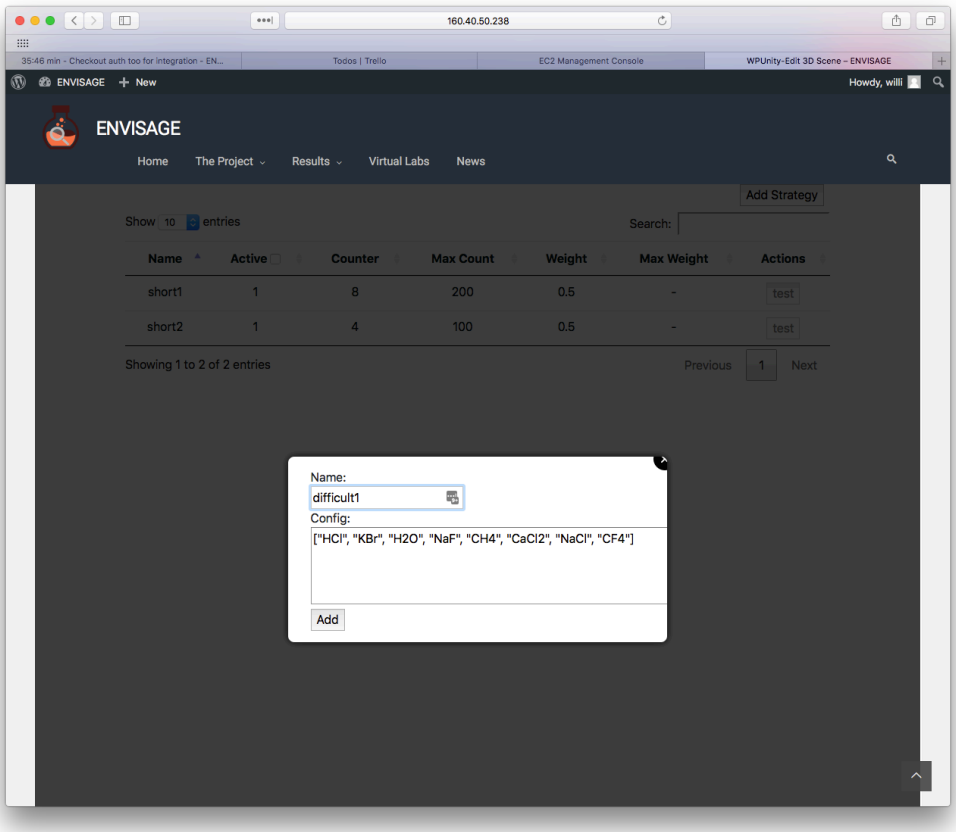


Figure 24: Adding a new strategy for a virtual lab from within the authoring tool.

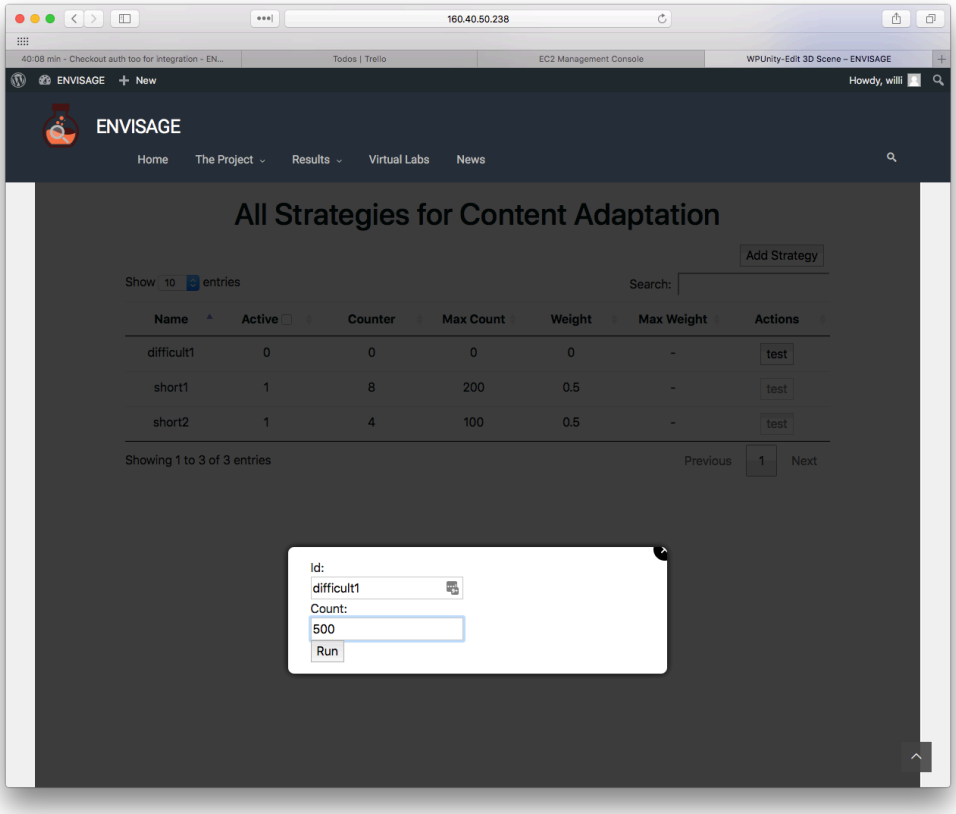


Figure 25: Screen for testing a strategy.

```
[
    "H2O", "HCl", "H2O", "KBr"
]
```

This strategy begins with water (H_2O), continues with hydrogen chloride (HCl), repeats water again, and finishes with potassium bromide (KBr). While entering JSON is quite technical and not every teacher may be used to that notation, it has the advantage that it gives a lot flexibility to the dynamic content adaptation. In the future, a developer of a virtual lab may provide a small tool that helps the teachers to generate proper JSON. These tools could be integrated in the authoring tool as well for each different type of virtual lab.

6.2.3 Test a Strategy

After a new strategy has been added to a virtual lab, it needs to be activated and tested. By setting up a test, the strategy is activated and an upper limit of tries is set. The screen in Fig. 25 shows how a test for a strategy is started. For new strategies, an initial number is set that determines how many students will see the strategy. For already tested strategies, the number can be increased if the maximum has been reached.

6.2.4 Future Views

There is a number of views that are currently work in progress and will be released in the near future. This includes but is not limited to:

Edit View This view allows to edit basic properties of a strategy such as the maximum number of trials or the current weight. It is important to note that a change of the weight triggers an update of the other strategies as well so that the probabilities add up to 100%.

Auto Redistribution The "Edit View" will allow to manually change the weights of the strategies. However, it is often more desirable to have the MABs re-adjust the weights automatically according to the performance.

Deactivate a Strategy Stops a current test and deactivates a strategy.

Performance View This view compares the performance of different strategies for a specific timeframe, e.g., the last month.

We have now seen how the demonstrator currently supports different types of machine learning algorithms. The outlook in the next section describes in greater detail how these different models and predictions can be combined in the future.

7 Outlook and Conclusion

As we have described in the previous sections, multiple deep analytics algorithms are operational by now. We have already applied them to different virtual labs and the evaluation of the results is still going on. Not every algorithm makes sense to be integrated into every virtual lab. For example, the design of the Wind Energy Lab does not support the usage of the prediction of at-risk students. Students may use the lab once to understand the physics of wind energy but are not necessarily encouraged to use the lab multiple times. On the other hand, the at-risk student predictions is technically ready to be used in the 3D versions of the chemistry labs but there is not always sufficient data available yet to learn a model for each lab. For that reason, we are planning to finish the integration of the predictions into all labs, once enough data was gathered. We currently assume that the next pilot phase will generated a batch of data at a reasonable scale to observe at-risk behavior of students in the chemistry labs. Nevertheless, we have also shown with the help of game data that the algorithms and the entire platform is capable of generating positive impact on a large scale and in business relevant use cases.

We have not started yet to implement the reinforcement learning for content adaptation in small steps as presented in Sec. 5. Here, one action in the algorithm amounts to adapting the content. The reward of this action is measured by students solving a task or exercise. The implementation of this approach in an educational setting requires a lot data and we should validate the simpler approaches based on A/B testing or Multi-armed bandits first. Once we have satisfying results from those approaches, we are at a good starting point to implement the more sophisticated reinforcement learning approaches. Realizing the multi-armed bandit problem with help of reinforcement learning might be a good approach to transit to the more advanced setting.

One avenue for future work that we consider to be equally interesting and potentially easier to realize in the remaining amount of time is the combination of the prediction of at-risk students and the content adaptation. As it has already been motivated, the two approaches can be connected by first continuously making an at-risk prediction for students and adapting the course material accordingly. For example, when the at-risk prediction indicates a high likelihood of a student failing or dropping out of a course because it is too challenging, the platform should intervene. There are different options to support those students. For example, the content for these students should be extended in such a way that it offers more supporting material that guides the students in solving the problems. In contrast to this, if the system identifies students that only spend little time in the virtual lab but easily solve all challenges, the content should be expanded in such a way that these students are challenged as well.

There are some technical extension that we consider to be meaningful and important for the infrastructure and platform. For example, a proper management of machine learning models for different teachers and virtual labs is very important. With new data arriving, models need to be updated and the algorithms need to decide which data to take into account. For example, we have observed that the virtual labs change substantially over time. With educators revising labs, possibly even based on the insights generated by the shallow and deep analytics, the structure of the data tracking changes. By doing so, previous datasets may be become obsolete and the algorithms should primarily learn form the most recent data. Nevertheless, the old datasets can be used to bootstrap the algorithms to learn more quickly. This entire process should be organized and implemented in such a way that educators can be informed about the current quality of the data and the models.

Teachers can even be guided to run certain experiments with the students to generate the next iteration of data without harming the quality of teaching. Such approaches are often referred to as active learning within machine learning. I.e., the algorithms request specific training examples to improve the quality of the model. Similar to A/B testing and genetic algorithm in Sec. 5, we also have to be careful when the algorithms suggest changes. All changes need to be compliant with the teacher's point of view and human intuition.

Let us briefly summarize the contributions and findings in the deliverable at hand. Besides providing an up-to-date overview on the Artificial Intelligence in Education community, our contributions focused on the implementation of deep analytics and the evaluation of the algorithms in six different case studies. We have shown how to use unsupervised clustering to group students based on their behavior in Sec. 3. We compared two different clustering algorithms, namely k-means and archetypal analysis. After examining the results, we concluded that archetypal analysis is better suited for clustering of students in the 2D Wind Energy Lab. We continued by using supervised learning algorithms to predict at-risk students and the performance of students in Sec. 4. We added three case studies to validate these approaches by not only using data from virtual labs but also a large scale dataset from an MMORPG. In Sec. 5 it was described how content can be adapted dynamically in virtual labs. We showed how to extend a chemistry lab to integrate a simple content adaptation and further motivated this approach by demonstrating that this approach was previously used in a mobile quiz game with great success. As this deliverable is of type "Demonstrator"', we showed in Sec. 6 in detail how the deep analytics is integrated into the ENVISAGE authoring tool and gave references to the corresponding source code if applicable.

While we have presented several algorithms in this deliverable and its predecessors, AI in education and deep analytics for virtual labs is still at a basic level. Although we can borrow many technologies from the gaming industry and rely on algorithms that have been analyzed for decades, the educational setting comes with its own challenges. For example, ethical obligations of teaching restrict the possible tests of learning strategies and require a more careful approach. Additionally, privacy regulations are justifiably more restrictive when it comes to school education. Nevertheless, the possible impact of AI in education is huge and the possibilities of personalized and active learning outweigh the challenges.

References

[1] Rakesh Agrawal, Behzad Golshan, and Evangelos E. Papalexakis. Toward data-driven design of educational courses: A feasibility study. In *Journal of Educational Data Mining*, pages 1–21, 2016.

[2] Mona Al-Saleem, Norah Al-Kathiry, Sara Al-Osimi, and Ghada Badr. Mining educational data to predict students' academic performance. In Petra Perner, editor, *Machine Learning and Data Mining in Pattern Recognition*, pages 403–414. Springer International Publishing, 2015.

[3] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, pages 235–256, 2002.

[4] Ryan Shaun Baker and Paul Salvador Inventado. *Educational Data Mining and Learning Analytics*, pages 61–75. Springer New York, 2014.

[5] Christian Bauckhage and Christian Thurau. Making archetypal analysis practical. In Joachim Denzler, Gunther Notni, and Herbert Süße, editors, *Pattern Recognition*, pages 272–281. Springer Berlin Heidelberg, 2009.

[6] Usamah bin Mat, Norlida Buniyamin, Pauziah Mohd Arsad, and RosniAbu Kassim. An overview of using academic analytics to predict and improve students' achievement: A proposed proactive intelligent intervention. In *IEEE 5th Conference on Engineering Education*, pages 126–130, 2013.

[7] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando Pereira, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2249–2257. Curran Associates, Inc., 2011.

[8] Michael D. Ekstrand, John T. Riedl, and Joseph A. Konstan. Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction*, 4:81–173, 2011.

[9] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2962–2970. Curran Associates, Inc., 2015.

[10] Peter A. Flach. *ROC Analysis*, pages 869–875. Springer US, 2010.

[11] Donn Randy Garrison. *Communities of Inquiry in Online Learning*, pages 352–355. Information Science Reference, 2009.

[12] Benedikte Mikkelsen Line Ebdrup Thomsen Georgios N. Yannakakis, Christoffer Holmgård. D2.4: Updated shallow analytics and visualization strategies. In *ENVISAGE*. 2018.

[13] Fabian Hadiji and Marc Müller. D2.1: Analytics infrastructure installation and data aggregation. In *ENVISAGE*. 2017.

[14] Fabian Hadiji, Rafet Sifa, Anders Drachen, Christian Thurau, Kristian Kersting, and Christian Bauckhage. Predicting player churn in the wild. In *IEEE Conference on Computational Intelligence and Games*, pages 1–8, 2014.

[15] Christoffer Holmgård and Fabian Hadiji. D2.3: Visualization strategies for course progress reports. In *ENVISAGE*. 2017.

[16] Christoffer Holmgård, Georgios Yannakakis, Daniel Mercieca, and Spiros Nikolopoulos. D3.1: Preliminary predicive analytics and course adaptaion methods. In *ENVISAGE*. 2017.

[17] Robin Hunicke. The case for dynamic difficulty adjustment in games. In *Proceedings of the ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, pages 429–433. ACM, 2005.

[18] Shimin Kai, Juan Miguel L. Andres, Luc Paquette, Ryan S. Baker, Kati Molnar, Harriet Watkins, and Michael Moore. Predicting student retention from behavior in an online orientation course. In *Proceedings of the 10th International Conference on Educational Data Mining*, 2017.

[19] Kyung-Joong Kim, DuMim Yoon, JiHoon Jeon, Seong-il Yang, Sang-Kwang Lee, EunJo Lee, Yoonjae Jang, Dae-Wook Kim, Pei Pei Chen, Anna Guitart, Paul Bertens, África Periáñez, Fabian Hadiji, Marc Müller, Youngjun Joo, Jiyeon Lee, and Inchon and Hwang. Game Data Mining Competition on Churn Prediction and Survival Analysis using Commercial Game Log Data. *ArXiv e-prints*, 2018.

[20] Ron Kohavi and Roger Longbotham. *Online Controlled Experiments and A/B Testing*, pages 922–929. Springer US, 2017.

[21] Rose Luckin, Wayne Holmes, UCL Knowledge Lab, and University College London. Intelligence unleashed: An argument for AI in education, 2016.

[22] Ioanna Lykourentzou, Ioannis Giannoukos, Vassilis Nikolopoulos, George Mpardis, and Vassili Loumos. Dropout prediction in e-learning courses through the combination of machine learning techniques. pages 950–965, 2009.

[23] Helen M. Marks. Student engagement in instructional activity: Patterns in the elementary, middle, and high school years. *American Educational Research Journal*, pages 153–184, 2000.

[24] Georgios Mavromanolakis, Pavlos Koulouris, Nikos Katsifos, Ioannis Kompatsiaris, Spiros Nikolopoulos, Giannis Chantas, Fabian Hadiji, and Marc Müller. D1.4: Data structure and functional requirements (update). In *ENVISAGE*. 2017.

[25] Benedikte Mikkelsen, Line Ebdrup Thomsen, Christoffer Holmgård, and Yannakakis Georgios N. D2.2: User profiling and behavioral modeling based on shallow analytics. In *ENVISAGE*. 2017.

[26] Olana Missura. *Dynamic Difficulty Adjustment*. PhD thesis, Rheinischen Friedrich-Wilhelms-Universität Bonn, 2015.

[27] Michael C. Mozer, Richard H. Wolniewicz, David B. Grimes, Eric Johnson, and Howard Kaushansky. Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry. *IEEE Transactions on Neural Networks*, pages 690–696, 2000.

[28] Amirah Mohamed Shahiri, Wahidah Husain, and Nur'aini Abdul Rashid. A review on predicting student's performance using data mining techniques. *Procedia Computer Science*, pages 414 – 422, 2015.

[29] George Siemens and Ryan S. J. d. Baker. Learning analytics and educational data mining: Towards communication and collaboration. In *Proceedings of the 2Nd International Conference on Learning Analytics and Knowledge*, pages 252–254. ACM, 2012.

[30] Rafet Sifa, Fabian Hadiji, Julian Runge, Anders Drachen, Kristian Kersting, and Christian Bauckhage. Predicting purchase decisions in mobile free-to-play games. In *Artificial Intelligence and Interactive Digital Entertainment*, 2015.

[31] Robert E. Slavin, Eric A. Hurley, and Anne Chamberlain. *Cooperative Learning and Achievement: Theory and Research*. John Wiley and Sons, Inc., 2003.

[32] Sofoklis Sotiriou, Pavlos Koulouris, and Georgios Mavromanolakis. D1.1: Educational scenarios and stakeholder analysis. In *ENVISAGE*. 2016.

[33] Sofoklis Sotiriou, Georgios Mavromanolakis, Pavlos Koulouris, Nikos Katsifos, Christos Tselembis, Ioannis Kompatsiaris, Spiros Nikolopoulos, Giannis Chantas, Fabian Hadiji, and Marc Müller. D1.3: Educational scenarios and stakeholder analysis (update). In *ENVISAGE*. 2017.

[34] Kai Ming Ting. *Confusion Matrix*, pages 209–209. Springer US, 2010.

[35] Dimitrios Ververidis, Stathis Nikolaidis, Anastasios Papazoglou, Christoffer Holmgard, Fabian Hadiji, and Marc Müller. D4.2: First version of the "Virtual Labs" authoring tool. In *ENVISAGE*. 2017.

[36] Su Xue, Meng Wu, John Kolen, Navid Aghdaie, and Kazi A. Zaman. Dynamic difficulty adjustment for maximized engagement in digital games. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 465–471. International World Wide Web Conferences Steering Committee, 2017.

[37] Georgios N. Yannakakis and Julian Togelius. *Artificial Intelligence and Games*. Springer, 2018. `http://gameaibook.org`.