Prediction Models for Individuals' Control Skill Development and Retention using XGBoost and SHAP

van Leeuwen, B.A.A.; Toubman, Armon ; van der Pal, Jelke ; Pool, D.M.

DOI
10.2514/6.2023-0542

Publication date
2023

Document Version
Final published version

Published in
AIAA SciTech Forum 2023

AIAA SciTech Forum
23-27 January 2023, National Harbor, MD & Online
AIAA SCITECH 2023 Forum

10.2514/6.2023-0542

Prediction Models for Individuals' Control Skill Development and Retention using XGBoost and SHAP

Barry A.A. van Leeuwen*
Delft University of Technology, Delft, Zuid-Holland, The Netherlands

Armon Toubman† and Jelke van der Pal‡
Royal Netherlands Aerospace Centre (NLR), Amsterdam, Noord-Holland, The Netherlands

Daan M. Pool§
Delft University of Technology, Delft, Zuid-Holland, The Netherlands

Current capabilities for predicting skill retention, i.e., the extent to which human operators retain learned skills over time, at an individual level are limited due to a requirement for large data sets and methods that can extract relevant patterns in highly dimensional data. This paper investigates the application of Extreme Gradient Boosting (XGBoost) decision tree models for predicting a high-resolution individual skill retention curve. For this, a large skill-based tracking experiment dataset is used to extract different feature classes and train an XGBoost predictive model. To identify the robust predictors, the effects of the different features on the model's output are analyzed using SHapley Additive exPlanations (SHAP). Furthermore, the proposed XGBoost model is trained using both the experiment dataset and a matched synthetic dataset, with both approaches evaluated on the experiment data. Overall, the available experiment dataset was found to include too few retention measurements, and too significant between-group differences, to extract a reliable prediction model. On the synthetic dataset, the XGBoost model was found to accurately capture individuals' skill retention curves, where the features that contributed most (21%) to the prediction model's accuracy were found to be the considered learning curve parameters. Overall, this paper shows that experiment data of skill-based tracking tasks can be used to predict skill decay curves using XGBoost, but that more research and data are needed to achieve sufficient accuracy and reliability at an individual level for practical applications.

I. Introduction

Learning and skill acquisition represent an expensive and crucial activity in most large organizations. For example, the total costs of training and development in the United States for 2011 were estimated to be $156.2 billion [1]. For many professions, such as pilots in aviation, skill training may be followed by long periods of inactivity, where learned skills are not used sufficiently to retain competence. To prevent problems, most organizations tend to provide repetitive and frequent training based on standardized intervals [2–8]. It is well-known that while effective, such an approach is inherently inefficient and causes unnecessarily high training costs. For example, [9] claims that only 10% of training costs typically result in "*enduring behavioral change*". Increasing the efficiency of training programs and explicitly optimizing for skill retention requires a better understanding of skill development and skill degradation over time, as well as improved predictive models that can be used to grasp and predict individuals' future training needs.

Over the last century, a significant amount of research has been performed into skill retention, its main influencing factors, and how the retention process may be captured in mathematical models [6, 10–16]. Unfortunately, experimental research into skill retention is often limited by practical considerations, as more expensive and time-consuming experiments with longer periods of inactivity and larger groups of participants than can often be achieved are, in

*M.Sc. student, Control and Simulation section, Faculty of Aerospace Engineering, P.O. Box 5058, 2600GB Delft, The Netherlands; b.a.a.vanleeuwen@student.tudelft.nl.

†Research Engineer, Training & Simulation Department, NLR, Amsterdam, 1059 CM, The Netherlands; Armon.Toubman@nlr.nl.

‡Senior Scientist, Training & Simulation Department, NLR, Amsterdam, 1059 CM, The Netherlands; Jelke.van.der.Pal@nlr.nl.

§Assistant Professor, Control and Simulation section, Faculty of Aerospace Engineering, P.O. Box 5058, 2600GB Delft, The Netherlands; d.m.pool@tudelft.nl. Senior Member AIAA.

fact, needed. Furthermore, generalizing the findings of such studies and any extracted simplified models for skill retention generally is often limited [17–19], due to complex human operator skills and large individual variations in skill decay. A currently under-explored research area is the potential application of Machine Learning (ML) models for the prediction of individuals' skill level and skill retention. In numerous applications, ML approaches have shown to excel in recognizing crucial patterns in large and complex multidimensional datasets.

This paper investigates the potential of using machine learning (ML) models for predicting individuals' skill retention in a skill-based manual tracking task. For this study, the individual task performance, (cybernetic) pilot model fitting, and demographic data collected from 37 participants in the previous training and retention experiment of [20] is used. This data is used to train Extreme Gradient Boosting (XGBoost) decision tree models, while the SHapley Additive exPlanations (SHAP) method is applied to extract important predictors to be used as primary features. Using both the experiment data from [20] and a matched and augmented synthetic dataset, the effectiveness of XGBoost models for individual prediction of skill retention is evaluated. The intended contribution of this paper is to verify the crucial features and effective ML model structure and hyperparameter settings for predicting an individual retention curve for skill-based manual control behavior.

The paper is structured as follows. Section II describes the experiment dataset of [20] and the skill-based tracking task used to collect it. Section III explains the two key ML methods used in this paper: XGBoost and SHAP. In Section IV the methodology followed in this paper is outlined, and its outcomes are presented in Section V. The paper ends with a discussion section (Section VI) and the main conclusions in Section VII.

II. Experiment Data

A. Experiment

1. Skill-Based Tracking Task

In this paper, data from a dual-axis compensatory tracking task experiment is used to develop a model that can predict individual skill decay over a period of inactivity. The experiment is described in [20], where its data is used for the objective evaluation of the retention of manual control skills using a 'cybernetic' pilot modeling method. To acquire operationally relevant results, a dual-axis pitch/roll tracking task was performed by a total of 43 task-naive participants, of whom 37 were able to provide a complete dataset. The experiment was performed in the fixed-base HMILab simulator at TU Delft, see Fig. 1.

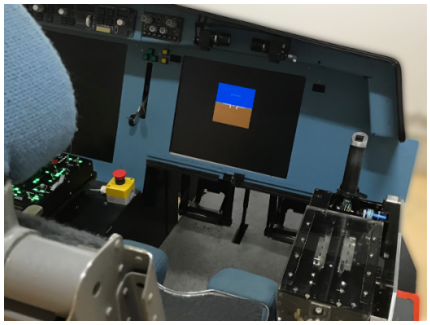


Fig. 1 Experiment setup in the fixed-base HMILab simulator at TU Delft [20].

A schematic representation of the dual-axis tracking task is shown in the block diagram of Fig. 2. As shown in Fig. 2, in this task the Human Operator (HO) is required to control the roll ϕ and pitch θ attitudes to match the forcing functions f_{t_ϕ} and f_{t_θ}. The HO controls the ϕ and θ outputs of the aircraft dynamics H_{c_ϕ} and H_{c_θ}, respectively, using a control stick with roll and pitch gains K_{s_ϕ} and K_{s_θ}. Human manual control behaviour in such a dual-axis compensatory tracking task can be captured with two parallel error responses [20, 21], as indicated in Fig. 2 by $H_{p_e\phi}(s)$ and $H_{p_e\theta}(s)$.

2. Human Operator Model

In [20], measured HO control behavior was quantified using "cybernetic" HO models as proposed in [21, 22]. In compensatory tracking tasks, see also Fig. 2, the HO control dynamics can be modeled using only a compensatory

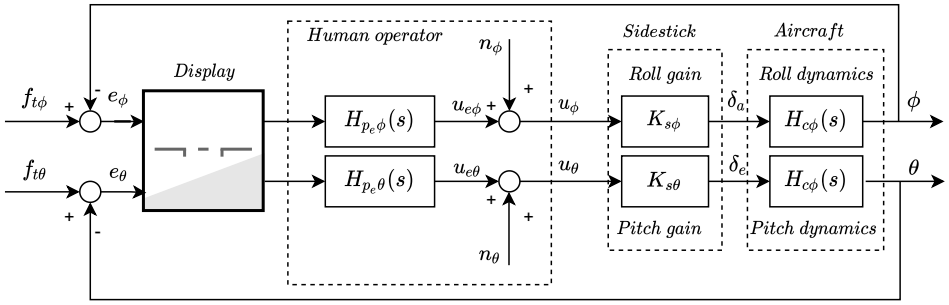


Fig. 2 Schematic representation of the compensatory dual-axis roll and pitch altitude skill-based tracking task performed in [20].

control response with the tracking error e as the input [22, 23]:

$$H_{p_e}(s) = K_p(T_L s + 1)e^{-\tau_e s} H_{nm}(s) \tag{1}$$

$$H_{nm}(s) = \frac{\omega_{nm}^2}{s^2 + 2\zeta_{nm}\omega_{nm}s + \omega_{nm}^2} \tag{2}$$

In Eq. (1) and (2), K_p, T_L, τ_e, ω_{nm} and ζ_{nm} are the HO's control gain, lead time-constant, time delay, neuromuscular frequency, and neuromuscular damping ratio, respectively. In [20], these parameters were estimated separately for both the roll and pitch human operator responses shown in Fig. 2.

3. Experiment Setup and Procedures

The experiment data was collected in a 6-month experiment that included an initial training phase and a retention phase. The training phase consisted of 100 training runs divided over four 1-hour training sessions performed on four successive days. The tracking runs were always 90 seconds in length. After the training phase, the participants were divided over three experiment groups, where the three groups were matched for their average end-of-training performance. As listed in Table 1, all three groups performed a final retention-phase measurement after 180 days. The participants in Groups 2 and 3 returned after 90 days or 60 and 120 days of inactivity, respectively, for additional intermediate retention tests.

Table 1 Experiment schedule used by [20].

Experiment schedule	Group 1	Group 2	Group 3
Training phase	100 runs	100 runs	100 runs
60-day retention test	-	-	5 runs
90-day retention test	-	5 runs	-
120-day retention test	-	-	5 runs
180-day retention test	25 runs	25 runs	25 runs

Finally, for potential correlation with the measured retention performance data [20] accumulated demographic data of all participants through a pre-experiment survey. However, no statistically significant correlation at the group level was identified, see the corresponding Appendix of [24] for details.

B. Data Structure

In this paper, the dataset collected in the experiment of [20] is used to extract a model that predicts individuals' skill decay. Table 2 lists all candidate features – i.e., characteristics/metrics that quantify a participant in the experiment – considered for this prediction. Table 2 lists the (coded) name of each feature and its unit, as well as the 'feature class' it was assigned to in our analysis. To see which types of data best facilitate accurate prediction of individual skill retention, the following feature classes are considered: 1) *Performance Data*, 2) *Learning Curve Data*, 3) *Retention Data*, 4) *Experimental Data*, 5) *Demographic Data* and 6) *Cybernetic Data*. Below each class is described in detail:

Table 2 Candidate features from [20] dataset.

Feature class	Feature	Unit	Feature class	Feature	Unit
Performance Data	RMSe@First5Trainingruns	deg	Demographic Data	GamingExperienceIO	-
	RMSu@First5Trainingruns	deg		GamingExperience	years
	RMSe@Last5Trainingruns	deg		StillGaming	-
	RMSu@Last5Trainingruns	deg		RetirdOfGaming	-
	RMSe@trainingrun100	deg		Hobbies	-
	RMSu@trainingrun100	deg		OtherHobbies	-
Learning Curve Data	Learning curve p_0	deg		Sport	-
	Learning curve p_a	deg		Still_executing	-
	Learning curve f	-		NonTrackingTaskGaming	-
Retention Data	RMSe@retenttest	deg		FineMotorSkill	-
	RMSu@retenttest	deg		GrossMotorSkill	-
	kp@retenttest	-		CognitiveDemandLow	-
	TL@retenttest	-		CognitiveDemandAverage	-
	tv@retenttest	-		CognitiveDemandHigh	-
	wnm@retenttest	rad/s		PhysicalDemandLow	-
	dnm@retenttest	-		PhysicalDemandAverage	-
Experiment Data	Retention Interval	days		PhysicalDemandHigh	-
	Retentietest Number	-		RandomInfluencingfactor	-
	Group Number	-	Cybernetic Data	kp@Last5Trainingruns	-
	Subject Number	-		TL@Last5Trainingruns	-
	Roll(0) or Pitch(1)	-		tv@Last5Trainingruns	-
Demographic Data	Age	years		wnm@Last5Trainingruns	rad/s
	(fe)male (1)/0	-		dnm@Last5Trainingruns	-
	AE-student	-		kp@First5Trainingruns	-
	CS-student	-		TL@First5Trainingruns	-
	Study year	years		tv@First5Trainingruns	-
	DriversLicenseobtained	-		wnm@First5Trainingruns	rad/s
	DriversLicenseYears	years		dnm@First5Trainingruns	-
	EstimatedKm/y	Km/years		kp@trainingrun100	-
	GamesNever	-		TL@trainingrun100	-
	GamesTwicePerYear	-		tv@trainingrun100	-
	GamesMonthly	-		wnm@trainingrun100	rad/s
	GamesWeekly	-		dnm@trainingrun100	-
	GamesDaily	-			

1) The class of *Performance Data*, as often considered in HO research [20, 25], quantifies participants' skill performance during the training phase, see Table 2. This class contains features, e.g., *RMSe@First5Trainingruns* and *RMSu@Last5Trainingruns*, that quantify task performance and control effort averaged over the first and last five training runs. This five-run average is used to reduce feature sensitivity to HO noise and randomness present in single samples and thereby accurately represent participants' performance profile during training. In addition, the features *RMSe@trainingrun100* and *RMSu@trainingrun100* are used to also include a single-run end-of-training performance snapshot from the 100^{th} training run. These features are included because the amount of progression during training, as well as the final level of task performance, are known to affect skill retention [24].
2) The class *Learning Curve Data* in Table 2 represents a reduced dataset based on the training phase performance ($RMS(e)$) data at the individual level. The performance variation across all training runs was quantified using an exponential learning curve model as defined in Eq. (3) and as also applied in [20, 25]:

$$y_{lc}(i) = p_a + (1 - F)^i (p_0 - p_a) \tag{3}$$

In Eq. (3), p_a, p_0, and F represent the final asymptotic performance level, the initial performance level, and the learning rate, respectively. These learning curve parameters are all considered as potential features, i.e., *Learning curve p_a*, *Learning curve p_0* and *Learning curve f* in Table 2. This training phase data is illustrated

in Fig. 3, which shows the individual training phase roll tracking performance data for participant 12 in Group 3 (yellow markers), as well as fitted learning curves for a selected participant from each group (solid lines) and group-averaged p_0 and p_a data (boxplots).

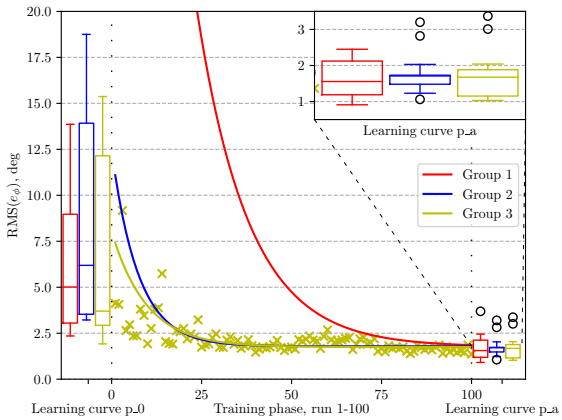


Fig. 3 Example features for *Learning Curve Data*.

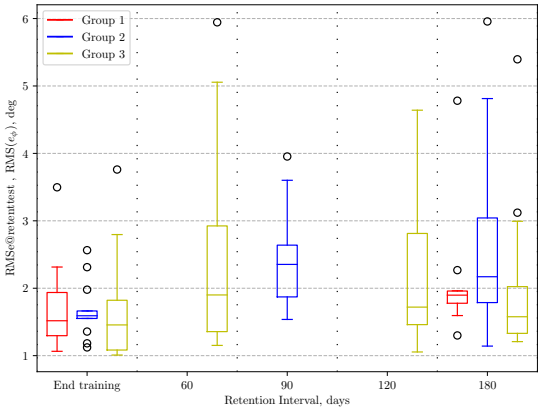


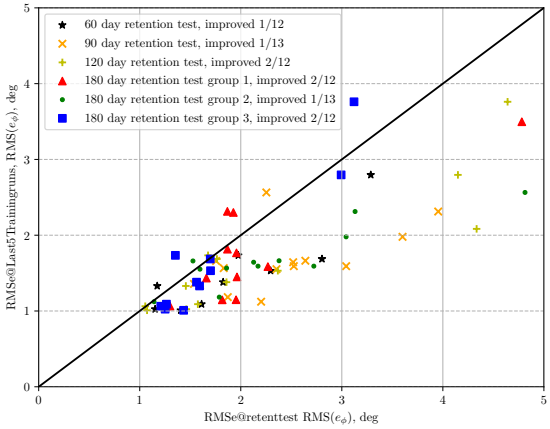
Fig. 4 Example *Retention Data* features.

3) The class *Retention Data* in Table 2 consists of the dependent measures from the experiment's retention phase, see Section II.A.3. To avoid capturing the effects of 'refresher training' due to the retention test schedule of Table 1, this class of features was calculated from only the first tracking run in each retention test. In this paper, we consider the *RMSe@retenttest* feature from the *Retention data* class as the output of the proposed model for predicting individual retention phase performance, i.e., we want to predict *RMSe@retenttest* (output) after a certain retention interval (input). For training the model, the data from all first retention test runs – i.e., multiple retention test runs for the participants from Groups 2 and 3, see Table 1 – are used. For the *RMSe@retenttest* feature, this is shown in Fig. 4 for all 3 groups over different *Retention Intervals* for the roll tracking task (ϕ) data. As the groups all had different retention schedules, see Table 1, the 60-, 90-, and 120-day *Retention Intervals* only show *RMSe@retenttest* data for a single group. Fig. 4 shows that, as expected, tracking performance worsens for all *Retention Intervals* compared to the end-of-training $RMS(e)$ data.
4) The feature class *Experiment Data* in Table 2 lists all experiment settings pertaining to the data from each participant, such as the *Retention Interval*. While the features from this class mostly represent the factors in the experiment that are not expected to strongly correlate with retention test performance, the *Retention Interval* feature is defined as the key input variable for the developed predictive model.
5) The class *Demographic Data* in Table 2 was derived from the participant survey data from [20]. The *Demographic Data* includes the participants' personal attributes that could influence skill retention of the tracking task. To enable interpretation and use as (numeric) input features, some of the *Demographic Data* features were encoded using a binary representation.
6) Finally, the class *Cybernetic Data* in Table 2 represents the estimated HO model parameters obtained with the HO model from Section II.A.2 during both the training phase and retention test(s). As also done for the *Performance Data* features, for the training phase the averages over the first and last five tracking runs, as well as the 100$^{\text{th}}$ training run values are considered as separate features.

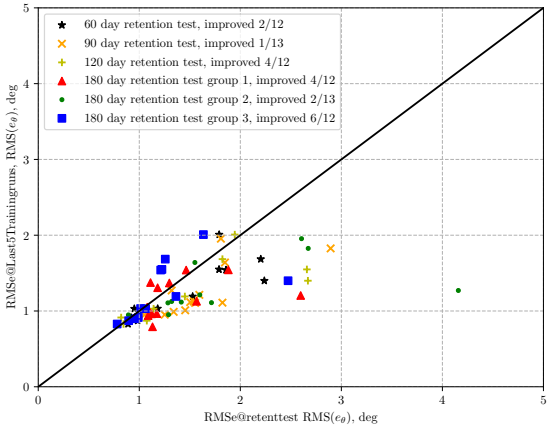
C. Individual Experiment Performance

To be able to develop a model that can accurately predict skill retention, the data used for training the model must include sufficient skill decay for the model to pick up on. Fig. 5 shows correlation plots where the end-of-training performance level (*RMSe@Last5Trainingruns*) is plotted as a function of the performance in the first retention tests (*RMSe@retenttest*). The data for the roll and pitch axes in the two-axis tracking task of [20] are shown in Fig. 5(a) and (b), respectively. For the expected degraded retention test performance compared to end-of-training, markers should be below the solid black 1-to-1 line included in Fig. 5; the legends in Fig. 5 show the numbers of participants with improved retention test performance, i.e., the opposite of what is expected when assumed that all individuals have reached their optimal performance. Overall, Fig. 5 shows that roll performance on average decays more than in pitch. Also, more participants show a performance increase in pitch than in roll. These observations are consistent with [21, 26–29],

where participants performing a similar dual-axis tracking task always prioritize pitch control. Given that skill decay occurs in 87.8% of roll-axis data points – compared to 74.3% for pitch – in this paper only the roll-axis data is used to develop and train a model for predicting individual skill retention.



(a) Overview of skill decay per retention interval per individual in *RMS*(e_ϕ).



(b) Overview of skill decay per retention interval per individual in *RMS*(e_θ).

Fig. 5 Participants' task performance compared between end-of-training and the first retention test, for roll ϕ (a) and pitch θ (b).

III. Machine Learning Methods

A. Extreme Gradient Boosting (XGBoost)

When Machine Learning (ML) methods are used to classify or regress data, in general an approach as illustrated in Fig. 6 is followed. First, a selected model structure is *trained* for predicting the output data *y_train* from the available input data *x_train*. The trained model can then be used for *applications*, where previously unseen input data *x_test* can be used to predict corresponding model output data *y_test*.

In this paper, Extreme Gradient Boosting (XGBoost) decision tree models, as proposed in [30], are applied to predict the skill retention of individuals for the experiment data described in Section II.C. XGBoost is an enhanced random forest estimating technique compared to the 'original' Gradient Boosting Regression Tree (GBRT) method, and is based on a statistical decision tree model. GBRT models can describe complex nonlinear relationships between input and output [31, 32]. This is achieved by dividing the input features over different trees in different layers and determining binary splits in which linear relationships are established. This method allows regression trees to return accurate regression predictions even with small datasets and high dimensionality [31, 32]. For our application, this characteristic is important as our dataset is small and has a relatively high number of features (67) compared to the number of samples (37). Also, decision tree models enable direct visual insight into the internal model structure and how the input-output relation is modeled, i.e., they provide an inherently interpretable model structure.

1. Gradient Boosting Tree Architecture

This section will briefly summarize the gradient boosting regression tree architecture and the XGBoost model structure as described in detail in [31–35]. GBRT is an advanced decision tree model that uses the concept of *boosting*, i.e., combining weak learners with other, iteratively formed, weak learners (decision trees) to form a strong predictor. A decision tree, of which an example is shown in Fig. 7, aims to divide all samples into two different groups according to specific, strategically chosen, features and cut-off criteria. Between these two groups, a certain threshold determines how all samples will be divided towards the next layer of the tree. After one or more layers, the sample will end at a 'leaf' with a certain number (depending on the XGBoost settings) of other samples. The average of these grouped samples represents the prediction for this specific leaf. Next, gradient boosting optimization is applied by adding more trees to the model's architecture to minimize residuals of the loss function, as visually shown in Fig. 8.

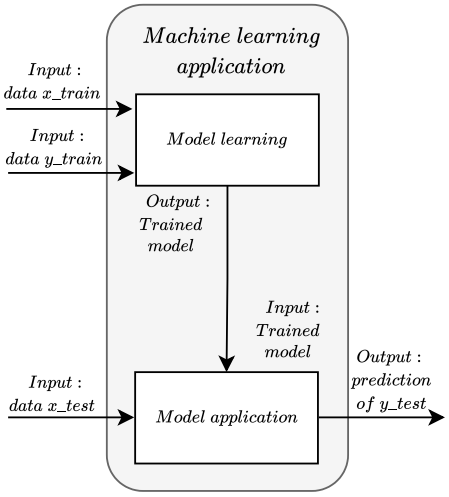


Fig. 6 Schematic representation of the general use of machine learning models.

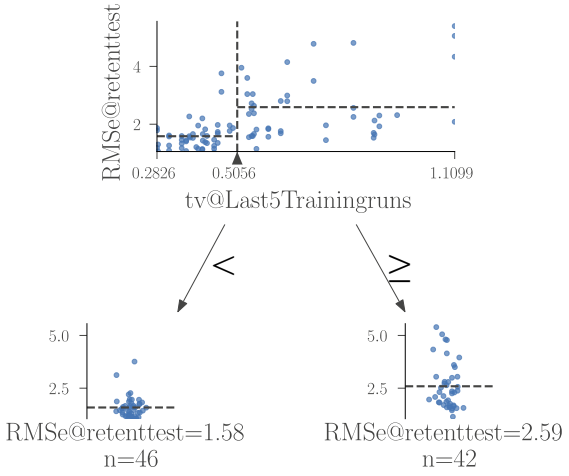


Fig. 7 Example visual representation of a single XGBoost regression tree *RMSe@retenttest* feature.

2. Gradient Boosting Tree Algorithm

In gradient boosting, the loss function $L(y, F(x))$ is defined to minimize the error between the predicted model output $F(x)$ and the true measured output data y. This loss function, defined in Eq. (4), is a direct sum-of-squares error across all N samples of the dataset. A gradient boosting tree model $F(x)$ is then generated starting from an initial model $F_0(x)$ that will be extended to $F_J(x)$ after J iterations, as defined in Eq. (5).

$$L(y, F_j(x)) = \sum_{i=1}^{N} (y_i - F_J(x_i))^2 \tag{4}$$

$$F_j(x) = F_{j-1}(x) + \rho_j h(x; \alpha_j) \tag{5}$$

As shown by Eq. (5), for each iteration $j = 1, .., J$ $F_j(x)$ will be updated by an increment $\rho_j h(x_i; \alpha_j)$. Thereby, another decision tree is added to the total model, as visually shown in Fig. 8. Here $h(x_i; \alpha_j)$ is called the 'base learner', i.e., the newly added decision tree, and is a function of the set of input features x_i. The model's coefficients α_j and ρ_j are both adjusted to achieve the best fit of the output y_i. The coefficients α_j and ρ_j are set through optimization, as explained in [34].

3. XGBoost Characteristics

As an extension of the standard GBRT method, XGBoost was designed to improve accuracy and reduce computational cost [30, 36]. The most important characteristics are: 1) sparsity-aware split finding, and 2) cache-aware access. Sparsity-aware split finding allows the model to compensate for missing values in the dataset by using the most common value as a default. The cache-aware access characteristics allow the model to pre-sort the data in buffers before it is provided to the cache threads, which reduces read/write dependencies [30]. These characteristics are beneficial additions to the XGBoost model with respect to the GBRT package of SciKit-learn [37], since they allow XGBoost models to more accurately and efficiently handle sparse, small, and highly dimensional datasets.

B. SHapley Additive exPlanations (SHAP)

A drawback of machine learning models is that they are generally complex and difficult to verify and interpret, due to convoluted model structures and very high numbers of model parameters. For machine learning models, a well-known tension exists between accuracy and interpretability [38]. As for many applications, including our focus in this paper, the need for model transparency exists, the *SHapley Additive exPlanations (SHAP)* method was developed to gain quantitative insight into the contribution of each feature [39]. SHAP stems from the cooperative game theory domain and is a method for detecting the magnitude of the contribution of individual features to a model's prediction

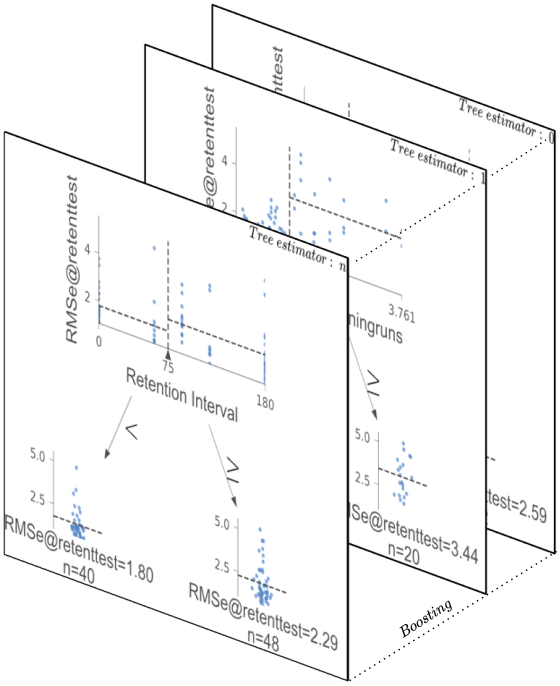


Fig. 8 Schematic representation of a complete boosted regression tree model architecture as used in XGBoost, here shown for the prediction based on the *RMSe@retenttest* feature.

[40]. These feature contributions are called SHAP values and give insight into the feature's importance for prediction [39]. In this manner, the model prediction $g\,(x')$ can be broken down into the SHAP values for all considered input features, according to Eq. (6):

$$g\,(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x_i' \tag{6}$$

In Eq. (6), $x' \in \{0,1\}^N$ is a binary array for N input features and ϕ_i is the SHAP value of feature i for which holds: $\phi_i \in \mathbb{R}$. Consequently, SHAP is also capable of analyzing XGBoost models based on a feature dataset M with m features using Eq. (7). The SHAP method is applied after the XGBoost model has been constructed and trained. Hence, SHAP will only be used to analyze the final XGBoost model $F_J(x)$. Since SHAP is an additive model, it only uses linear methods to analyze nonlinear models. This implies that SHAP does assume that all features are independent, which, however, is not always true. SHAP determines the SHAP value for all features by constructing subsets of features $S \in N$, as shown in Eq. (7).

$$\phi_i = \sum_{S \in M} \frac{|S|!(m - |S| - 1)!}{m!} [F_J(S \cup \{i\}) - F_J(S)] \tag{7}$$

The combination of XGBoost and SHAP is widely used for regression problems [36, 40]. This is because this combination enables users to regress complex nonlinear data sets, visualize the modeled relations using decision trees, and quantify each feature's importance accurately. In contrast, other ML models, such as recurrent neural networks [41] or support vector machines, do not facilitate the same level of model interpretation as obtained with the combination of XGBoost and SHAP.

IV. Methods

A. Model Performance Metrics

To evaluate the XGBoost model's performance and accuracy, the regression prediction $\hat{y}$ is evaluated using the Mean Absolute Error (MAE) as defined in Eq. (8). Here, matching our application of the XGBoost model, the true

output data to be matched by the model is indicated by the m samples of *RMSe@Retenttest*.

$$\mathrm{MAE}(\hat{y}, RMSe@Retenttest) = \frac{1}{m}\sum_{i=1}^{m}|\hat{y} - RMSe@Retenttest| \tag{8}$$

For evaluating XGBoost model performance, a baseline for its prediction performance is defined. In this paper, we use the MAE of a 'constant' prediction, i.e., for which no skill loss occurs and *RMSe@Retenttest* is thus equal to *RMSe@Last5Trainingruns*, as the baseline. This baseline is separately calculated for each participant in the dataset. However, to quantify overall prediction performance across the whole data set, the model performance metrics are averaged across all participants.

B. Hyperparameter Tuning

Next to the features and ML model, the chosen hyperparameter settings are essential for an accurate regression prediction. A hyperparameter is a predefined setting for a ML method [42]. The important hyperparameters and their operating value ranges that are tuned in this study are listed in Table 3. In Table 3, the general operating ranges of the hyperparameters are included, except for the hyperparameter *Objective*, as this hyperparameter defines an evaluation method. The hyperparameter settings in this research, including the different types of objectives, will be presented in Section V.

Table 3 Hyperparameters and range of settings for the XGBoost model.

Hyperparameter	Lower range value	Upper range value
Objective	-	-
Learning rate	0	1
Number of trees	1	∞
Max depth	0	∞
Subsample	0	1
Min child weight	0	∞
Colsample bytree	0	1

The performed hyperparameter tuning was split into two steps to pursue a semi-greedy approach, which was implemented as follows: 1) determining the optimal and smaller hyperparameter range in the hyperspace of hyperparameters, and 2) applying a grid search (*Gridsearch CV* [42]) on the different data subsets combined with the optimal hyperparameter range. During the first step, an iterative greedy approach is used to determine the optimal hyperparameter range for only two hyperparameters per iteration. The second step is executed by using *Gridsearch CV*, an exhaustive search method for the optimal hyperparameter settings given a grid of allowed parameter values [42].

As explained in Section II.B, the different feature classes considered in this paper vary significantly in their number of features. For this reason, it was found that a 'universal' set of hyperparameters could not accommodate optimal prediction performance for all feature classes, which complicates comparing between the different feature classes. However, by applying the semi-greedy approach to hyperparameter tuning to each feature class separately, we compare between the optimal effectiveness of the different feature classes when they minimize the computational costs. As a consequence, however, any performance differences observed between feature classes can result from two different factors: the features' informativeness for explaining the output feature data, or the hyperparameter settings.

C. Synthetic Data

The dataset introduced in Section II possesses a relatively low number of samples (N = 74) and a high number of features (m = 67). This makes the data highly dimensional, which complicates recognizing consistent patterns in the dataset. Hence, for the analysis in this paper, we also generate additional synthetic data, based directly on the statistical properties of the experiment data, to help analyse the performance of the XGBoost model approach. To achieve this, a multivariate Gaussian process is applied to generate synthetic data samples. Based on the mean and covariance present in the experiment data, this approach produces additional feature samples while retaining the spread and covariance between the features, assuming normal distributions. However, some of our feature values should not go below specific thresholds, which for a Gaussian process will always be possible for features with high variances. Hence, to avoid unrealistic data patterns, minimum thresholds were applied to the generated synthetic data to exclude unrealistic feature values.

An example result of applying the multivariate Gaussian process on experiment data is shown in Fig. 9, where a comparison between 2849 synthetic generated samples (blue) and the 74 samples of experiment data (orange) is shown for the *RMSe@retenttest*, *RMSe@Last5Trainingruns* and *Retention Interval* features. The 2849 synthetic samples in Fig. 9 were obtained from 80,000 generated samples after application of the exclusion thresholds. As intended, the values of the synthetic data overlap with the experiment data range. As shown in Fig. 9, the synthetic data is generated such that also more samples with varying *Retention Intervals* are available to train the XGBoost model.

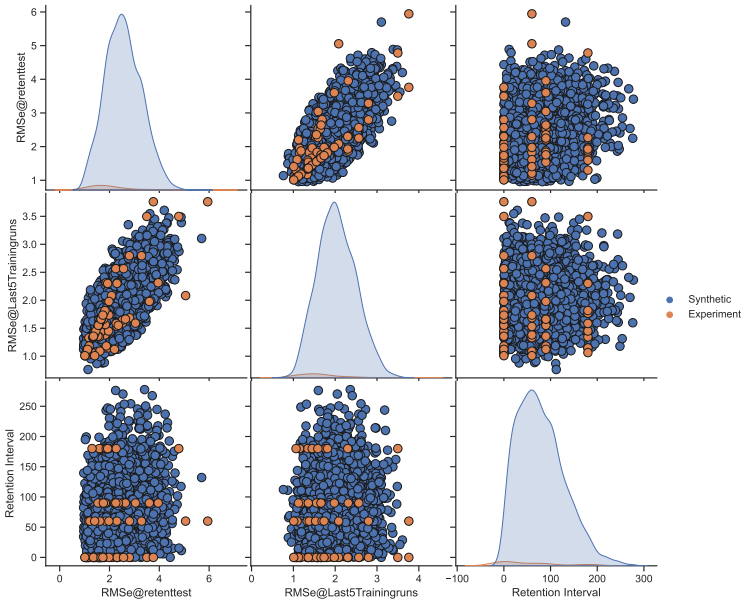


Fig. 9 Synthetic data generated using a multivariate Gaussion process for the *RMSe@retenttest*, *RMSe@Last5Trainingruns* and *Retention Interval* features.

D. Model Construction Workflow

In this paper, we follow a five-step workflow for deriving individual skill retention prediction models from our experiment and synthetic datasets, see Fig. 10.

Step 1: SHAP analysis In Step 1, the data is pre-processed and split into training and test datasets as described in Section II. Then, the XGBoost model is fitted to the data to determine the SHAP value for each candidate feature. For the feature rank analysis, we generate 100 regression tree models and sum the (positive and negative) SHAP values across the repeated model fits. For each repetition, the train and test data is randomly split to ensure a valid analysis. The final outcome of Step 1 is the SHAP feature rank, from which the features with the highest absolute SHAP values can be considered the most influential (and important) features for the model's prediction.

Step 2: Hyperparameter tuning In Step 2, the experiment data from [20] is used to determine the optimal XGBoost model hyperparameter ranges. Since we compare XGBoost models with multiple different input feature classes, see Section II.B, we optimize settings for each feature class. Therefore, in this Step, the most influential hyperparameters are determined heuristically, so that emphasis can be placed on tuning these hyperparameters. The resulting hyperparameter ranges are chosen based on the resulting model performance, as well as, their potential for interaction with other hyperparameters.

Step 3: Optimize number of features & performance analysis With Steps 1 and 2 completed, in Step 3 we use the experiment data to extract models for individual performance prediction. For this, first the number of features used in the XGBoost model is optimized. Based on a performance analysis of the different feature classes described in Section II, the minimum number of features is determined for which the models still show acceptable (asymptotic) performance in

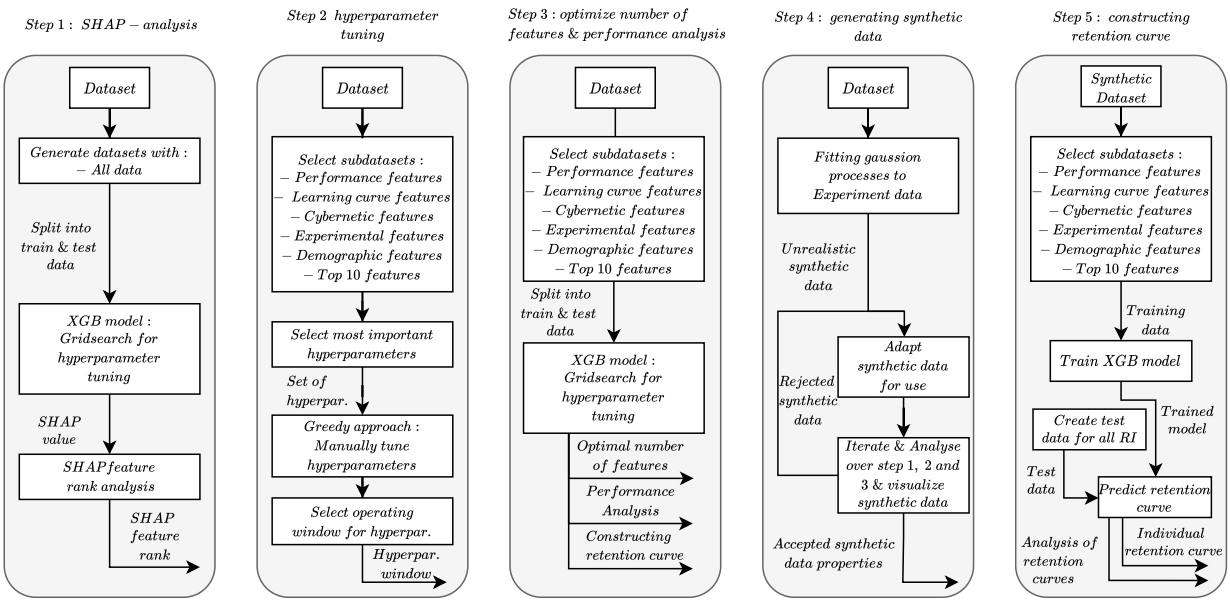


Fig. 10 Model construction workflow followed in this paper.

terms of the cross-validated MAE. For comparing models with varying numbers of features included, an approach similar to stepwise regression [43] is used. In this paper, the stepwise regression is implemented as follows: from a baseline model where only the best-ranked feature from Step 1 is included, at every iteration the next lower-ranked feature is added to the model's input feature set. This process results in the selected XGBoost model structure, which is then trained on the experiment data for a final performance comparison. Finally, the experiment-data trained model is used to predict individual retention curves, using the methodology explained in detail below under Step 5.

Step 4: Generating synthetic data As the experiment data size was found to limit the results of Steps 1 to 3, which may impact our assessment of the XGBoost method for predicting skill retention, in Step 4 we show the results of our proposed synthetic data generation as explained in Section IV.C. To analyse and validate that the synthetic data is suitable for the retention prediction in comparison to the experiment data, both experiment and synthetic data are compared based on the outcomes of Steps 1-3 with a matching number of samples.

Step 5: Constructing retention curves from synthetic data In Step 5, the synthetic datasets generated in Step 4 are used to select optimal features and hyperparameter settings for synthetic data, train an XGBoost model, and extract individual retention curve predictions from the trained model. As opposed to the experiment, which contain a very sparse set of *Retention Intervals* for model training, the synthetic data used here include a much more informative variety in training data, as explained in Section IV.C. The trained XGBoost model's output prediction of *RMSe@retenttest* as a function of its *Retention Interval* input feature value, while keeping all other test data input features constant, see Table 4, provides the predicted retention performance per participant. Furthermore, the XGBoost model predictions are fitted with a second-order polynomial to smooth the XGBoost model's inherently discontinuous predictions. To conclude Step 5, the performance of the approach is analyzed over all participants in the dataset and the extent to which an improved prediction is achieved compared to the (no skill loss) baseline prediction, as defined in Section IV.A, is verified.

V. Results

A. Step 1: SHAP Analysis

As the main result of Step 1 of our analysis, as introduced in Section IV.D, Fig. 11 shows the top ten largest (positive and negative) SHAP values identified for the roll-axis (ϕ) and pitch-axis (θ) data in blue and orange, respectively.

Table 4 Input feature dataset for extracting an individual retention curve.

Sample number	Roll(0) or Pitch(1)	...	Retention Interval	...	dnm@trainingrun 100
1	0		1		0.52
2	0		2		0.52
...			...		...
198	0		198		0.52
199	0		199		0.52

Fig. 11 shows the ranked features such that the most influential features are on top, with independent rankings for ϕ and θ. We only show the top 10 ranking features in Fig. 11, accounting for a total of 69.6% of the model's predictions, for brevity. As SHAP values scale directly with the magnitude of the predicted values, and in our data $RMS(e)$ was considerably lower in pitch than in roll (see Fig. 5), the SHAP values in Fig. 11 are also feature-wise higher for roll than for pitch. It should be noted that for the SHAP analysis, all features in the dataset are used, i.e., including features that are more or less equivalent. For example, Fig. 11 shows that for both the roll and pitch datasets both the features *RMSe@Last5Trainingruns* and *Learning curve p_a* are ranked in the top 10. As both features quantify the end-of-training level of task performance, the presence of both features affects the SHAP values and, consequently, increases or decreases their relative rank compared to other features. This implies that if one of these features would be excluded, the remaining feature would have an even higher SHAP rank in our analysis.

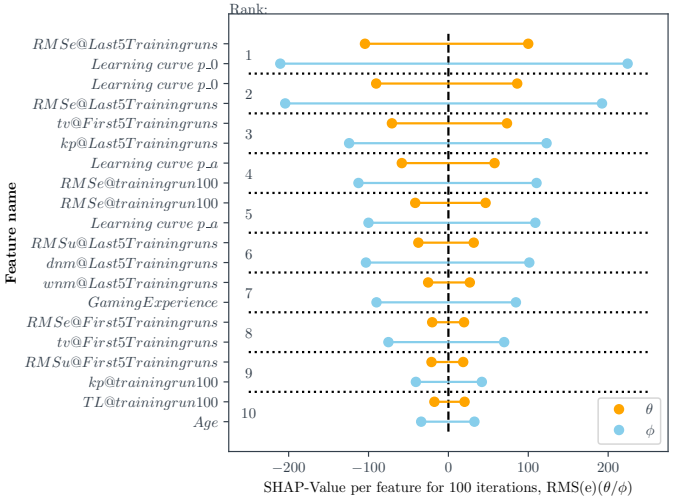


Fig. 11 SHAP values and rank for the top 10 contributing features for the roll (ϕ) and pitch (θ) experiment data.

Fig. 11 thus directly shows the importance of *Performance Data* features that quantify the level of task performance in the training phase. For example, the features *RMSe@First5Trainingruns* and *RMSe@Last5Trainingruns* occupy two of the first three ranks for both roll (ϕ) and pitch (θ). Furthermore, also comparable features to those from the *Performance data* class, such as *Learning curve p_a* and *Learning curve p_0* also appear high on the feature ranking. Overall, Fig. 11 thus shows that the XGBoost model's prediction is strongly dependent on the feature classes *Performance Data* and *Learning Curve Data*. As the extent to which control skills are retained is known to be affected by the level of performance, as well as the performance improvement, during initial skill acquisition, the fact that these features significantly contribute to prediction of the retention test performance is not surprising.

In addition, the class of *Cybernetic Data* features, obtained from human operator model fits to experiment data, has at least five features ranked in the top 10 for both ϕ and θ. This indicates the potential benefit of this class of features, that directly quantify specific aspects of the participants control skills, for retention performance prediction. Lastly, while Fig. 11 does not show many features from the *Demographic Data* class, the *GamingExperience* and *Age* features are found to significantly contribute to the retention prediction for the roll task data. Overall, the *Demographic Data* and also the *Experiment Data* feature classes seem to contribute to less to the retention prediction than anticipated.

These results will be revisited in Step 3 (Section V.C), where the XGBoost model with optimized input features and hyperparameter settings is analyzed.

B. Step 2: Hyperparameter Tuning

In Step 2 of our model construction workflow (see Section IV.D), a hyperparameter tuning was performed for the XGBoost model. Fig. 12 shows the cross-validated MAE for the retention test $RMS(e)$ prediction by the XGBoost model for varying hyperparameter settings for the *Number of trees* (x-axis) and the *Learning rate* (different lines). These hyperparameters together most strongly affect the training of the XGBoost model, see Table 3. Fig. 12 shows a clear pattern, where higher *Learning rates* are seen to require a lower *Number of trees* for reaching a minimum prediction error. Still increasing the size of the model (*Number of trees*) beyond this point will result in worse prediction performance value (higher MAE). In addition, the higher the learning rate, the quicker the model performance drops with an increasing number of trees, see inset in Fig. 12. Hence, using a lower learning rate tends to yield more stable and accurate XGBoost model performance. Table 5 lists the hyperparameters that were tuned for each feature class for the experiment data, as well as their upper/lower limits and increment as considered in the search grid.

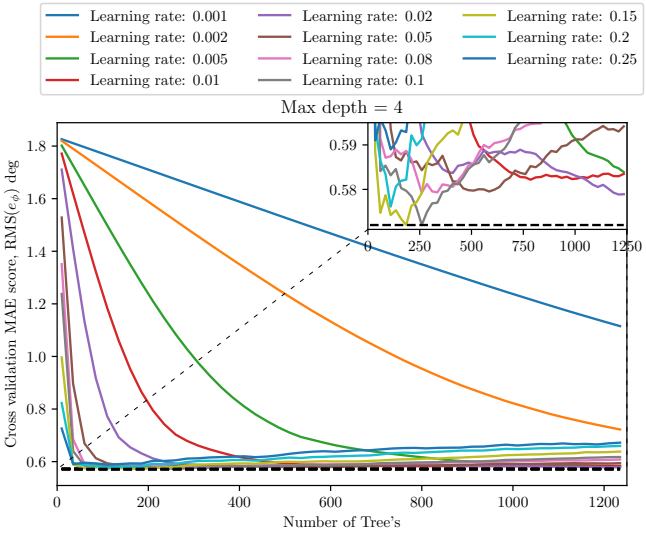


Fig. 12 Cross-validated MAE for the XGBoost model (all features are included) for varying *Learning rate* and *Number of trees*. The *Objective* is 'regression:squaredlogerror', while *Max depth* is set to 4.

Table 5 Sensitivity analysis hyperparameter ranges used in Step 3.

Hyperparameter	Upper limit	Lower limit	Increment
Learning rate	0.02	0.005	-
Number of trees	800	200	100
Max depth	4	1	1
Subsample	0.95	0.5	0.1
Min child weight	4	0	1
Colsample bytree	0.95	0.5	0.1

For performing the hyperparameter optimization with *Gridsearch CV*, the most suitable regression objectives were found to be: 1) pseudo-Huber, 2) squared log, and 3) gamma. Furthermore, it was found from the analysis that any potential relationships between the hyperparameters in relation to the training data set were difficult to pinpoint due to the selected greedy approach described in Section IV.B. The hyperparameter analysis aims to reach stable and optimal hyperparameter settings for each feature class while minimizing computational costs. It should be noted that truly optimal prediction performance is not guaranteed, due to the limited-resolution parameter space grid fed to *Gridsearch CV*, see Table 5. Consequently, this also complicates the comparison of prediction performance for the different feature

class, since both the used hyperparameters and the dataset are varied concurrently. Still, during Steps 3 to 5 of our analysis, we compare these different optimized XGBoost models for the different feature classes.

C. Step 3: Optimize Number of Features & Performance Analysis

1. Optimal Number of Features

For selecting the optimal number of features for each feature class, the SHAP rank as previously also considered in Section V.A is used. Fig. 13 shows the 15-fold cross-validated MAE for the XGBoost model's prediction of the retention test $RMS(e_\phi)$ with optimized hyperparameters for each feature class, see Section V.B. The different feature classes as defined in Table 2 are shown with different colored lines. Using stepwise regression, the first feature with the highest SHAP rank is first added to the model, while for the next iterations the features are added in descending order of their SHAP rank. Fig. 13 shows the results of this analysis for the (at most) 10 features with the highest SHAP rank in each feature class. For classes that consist of less than 10 features (e.g., *Learning Curve Data* only includes three features), the MAE is shown up to when all features are included.

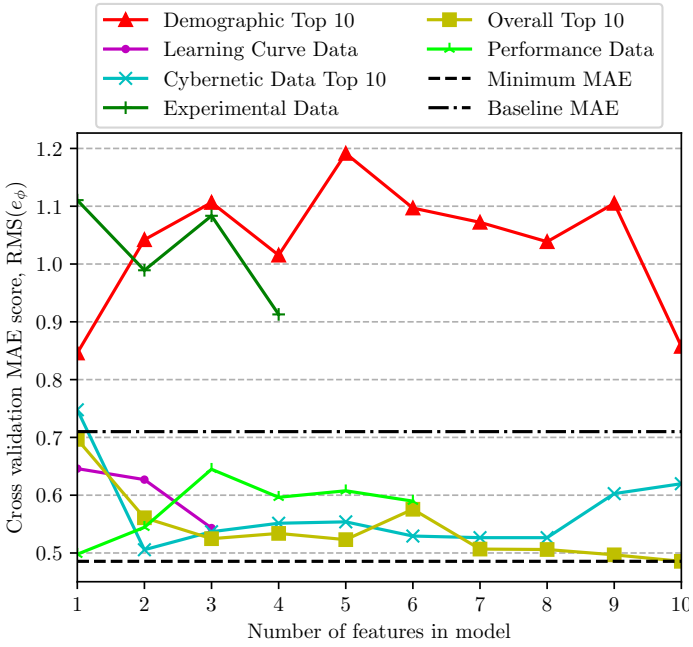


Fig. 13 Cross-validated XGBoost model performance for the different feature classes as a function of the number of included features.

Fig. 13 shows that increasing the number of features used in the XGBoost model does not improve the performance in all classes. For example, with only the classes of *Demographic Data* and *Experiment Data* the XGBoost model never achieves MAE values lower than the baseline (no drop in $RMS(e_\phi)$ compared to the end-of-training value), which implies that capturing the skill decay in the data with only these features is not possible. Furthermore, for the *Demographic Data* the MAE is lowest when only a single feature is included. The cross-validated performance of the optimized XGBoost model is better than the average baseline MAE of 0.71 for the other four feature classes. For these feature classes that do achieve prediction performance better than the baseline, the largest incremental improvement occurs up to the addition of four features. With more features, the MAE does not improve anymore or even worsens, a sign of overfitting. Overall, the best combination of feature class and number of features shown in Fig. 13 is the *Overall Top 10* with all 10 features included, which results in an MAE of 0.49.

Overall, the results shown in Fig. 13 for the XGBoost models with optimized hyperparameters are highly consistent with the observations made in Section V.A. The feature classes that result in the best achieved prediction performance include measures of task performance (e.g., *Performance Data* and *Learning Curve Data*), but also the *Cybernetic Data* shows the same acceptable performance. A more surprising outcome is that in Fig. 13 the feature class *Performance Data* performs best with only a single feature, as the MAE increases when more features are included.

2. Performance Analysis

In this section, the performance of a single (non-cross-validated) prediction of the XGBoost model for the data from Group 1 is analyzed to provide the potential accuracy for practical applications. Fig. 14 shows the prediction accuracy of the XGBoost model for a random train and test dataset and the different considered feature classes. In Fig. 14, the measured retention test level of task performance (i.e., *RMSe@retenttest*) of each sample is plotted against the corresponding prediction of the XGBoost model. The diagonal line indicates a 1-to-1 correlation and a histogram of the deviations of all samples from the 1-to-1 line is shown in the inset. Finally, the corresponding MAE value for each feature class is listed in the legend. It should be noted that the predictions shown in Fig. 14 are not cross-validated, which explains the lower MAE values compared to Fig. 13 (averaged over fifteen different test sample sets).

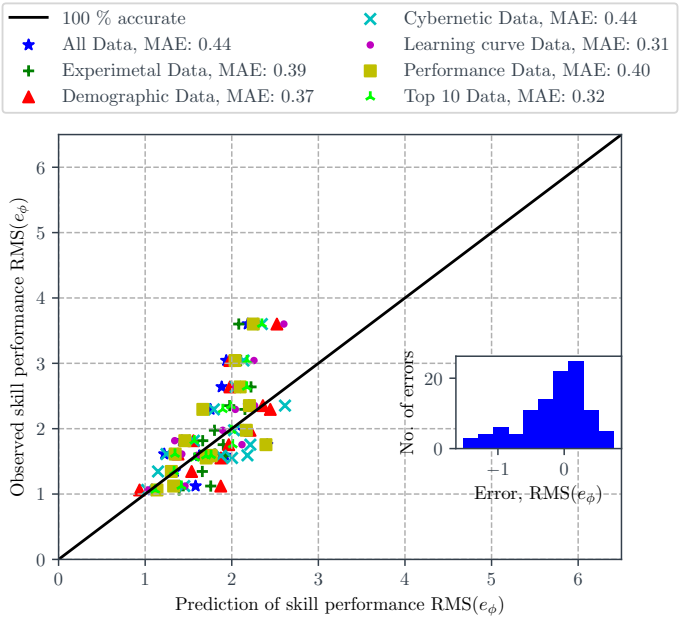


Fig. 14 Retention prediction of the XGBoost model trained and evaluated for all feature classes for the roll-axis data of Group 1.

Similar to Fig. 13, the *Learning Curve Data* class performs very accurately, i.e., with an MAE value just below that for the *Top 10 Data*. Surprisingly, the classes of *Performance Data* and *Cybernetic Data* perform the worst, together with the *All Data* case where all feature classes are included. This result may be explained by the fact that it is obtained for a single train and test set, for which the specific characteristics of the training samples strongly influence which features are affect the model's prediction. Furthermore, as opposed to previously presented results, the prediction shown in Fig. 14 includes the data from Group 1 of the experiment only, which may also contribute to this reduced importance of these classes. Finally, for the set of test samples shown in Fig. 14, the model slightly underestimates *RMSe@retenttest*, as most samples are found above the diagonal line. However, especially for participants that perform well in the retention tests (i.e., low *RMSe@retenttest* values) the predictions are quite accurate and positioned close to the 1-to-1 line.

3. Retention Curve Prediction

The main goal of the research described in this paper is to develop an approach to predict the skill retention curve (i.e., how quickly control skills are lost after initial training) for an individual. As explained in Section IV.D, in this paper we extract a retention curve prediction from the trained and optimized XGBoost model by evaluating its predicted output for the (individual) input feature sequence of Table 4. As explained before, the model is trained on only the (end-of-)training data features combined with the measured (degraded) performance in the first retention test to capture non-confounded skill retention. With this approach we ensure the XGBoost model predicts a retention curve dependent on an individual's specific characteristics, which is expected to correspond with the corresponding retention test data available in the dataset of [20]. Fig. 15 shows seven different skill retention predictions and fitted curves for participant 12, for the different considered feature classes. The black triangles show the measured $RMS(e_\phi)$ during the retention

tests of the experiment (*RMSe@Retenttest*) for this specific participant. Data for the 60-, 120-, and 180-day retention tests is shown for this participant, as he/she was in Group 3 of the experiment.

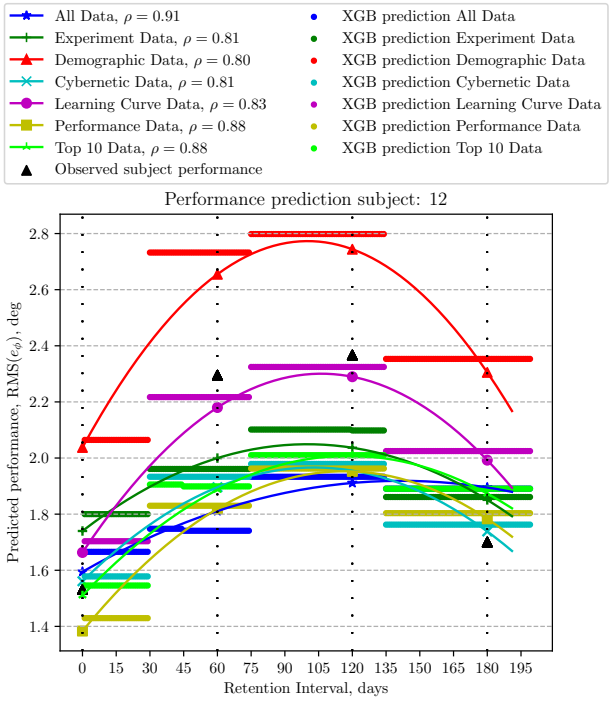


Fig. 15 Predicted retention curves for participant 12 for all feature classes compared to experiment data.

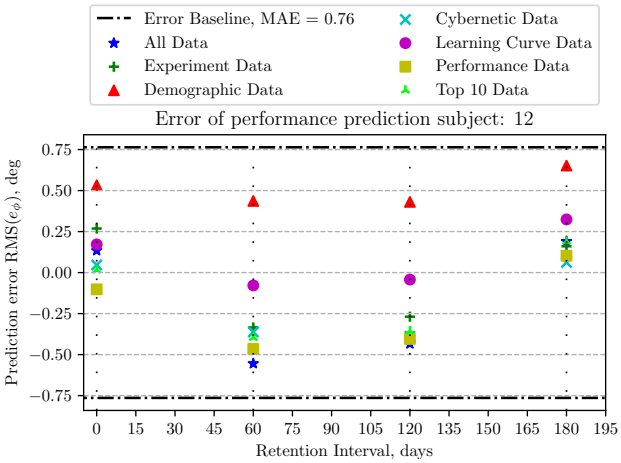


Fig. 16 Prediction errors for participant 12 for feature classes compared to the experiment data.

Fig. 15 shows that the performance level predicted by the XGBoost model (circular markers) as well as the fitted curves (lines) follow a parabolic trajectory for all feature sets. This parabolic pattern is not expected for a skill retention prediction, as the predicted drop after a retention interval of around 100 days would suggest skills improve (instead of deteriorate) over extended periods of time. This result is directly caused by the available training data for this study, for which a number of participants – especially in Group 1 of the experiment [20] – performed surprisingly well in the 180-day retention test. Fig. 15 shows that the same occurs in the measured data for participant 12 (black triangles), hence the predicted retention curves show a reasonable match with the experiment data. Finally, Fig. 15 also shows that the raw predictions from the XGBoost model have quite low resolution, as they consist of only four different segments across the range considered for the retention interval. This is a direct result of the low variety in retention intervals in the training data, which, in turn, implies the XGBoost model will only learn to provide separate performance predictions over 60-day intervals.

Fig. 16 shows the corresponding prediction errors for all feature classes. The black dashed lines indicate the prediction error level for the baseline prediction that assumes *RMSe@Retenttest* is equal to the end-of-training value of $RMS(e_\phi)$. For participant 12 the retention curve predictions have superior accuracy than the baseline at all moments where retention test experiment data is available. This was found to not be the case for the data from all participants, as not for all of them the parabolic shape of the predicted retention curve matches the experiment outcomes. Overall, it was found that the low resolution in the available retention test data for XGBoost model training is insufficient for extracting a meaningful individual retention curve for all participants, due to significant between-subject variability.

D. Step 4: Synthetic Data Generation

Section V.C.3 has shown that the experiment data from [20] lacks variety and resolution in the retention interval for XGBoost model training. Hence, to still verify the methodology, an extended synthetic dataset was generated using the methods described in Section IV.C. Fig. 17 shows the distributions and correlations of *RMSe@retenttest*, our XGBoost model's output, and the three highest ranked features for the experiment data – i.e., *Learning curve p_0*, *RMSe@Last5Trainingruns*, and *kp@Last5Trainingruns*. The experiment data is indicated in orange, while the

synthetic data (shown here with the same number of samples as available in the experiment data) is shown in blue. The figures on the diagonal of Fig. 17 show a comparison of the experimental and synthetic data distribtions, where the listed p-values indicate whether both distributions are statistically different. The off-diagonal figures show how these different selected features correlate. Overall, Fig. 17 shows that the synthetic data provides a reasonable match to the

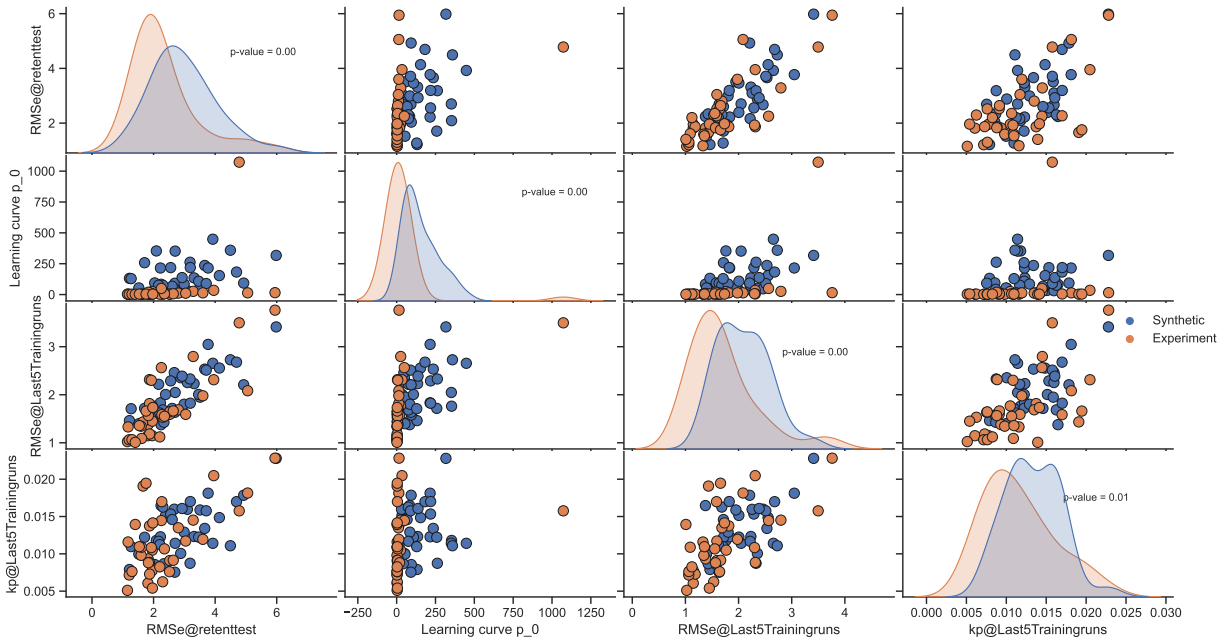


Fig. 17 Comparison of equal samples of experimental and synthetic data for *RMSe@retenttest* and three of the Top 10 contributing input features.

experiment data. The synthetic data shows more variety, as intended, but does retain the crucial correlations (e.g., of *RMSe@Last5Trainingruns* and *kp@Last5Trainingruns* with *RMSe@retenttest*). Furthermore, Fig. 17 shows that the experiment data distributions are all skewed towards zero compared to the generated (Gaussian) synthetic data. The higher-valued synthetic samples generated for *Learning curve p_0* can be attributed to the presence of outliers in the experiment data that increase the feature's measured mean used to generate the synthetic data. Overall, Fig. 17 shows that the experiment data distributions are all non-Gaussian, which also explains the fact that all listed p-values are very small, and hence the experimental and synthetic data distributions are found to be statistically different.

Similar to the analysis in Section V.A, the generated synthetic data was analyzed to determine the SHAP ranks of all included features for comparison to the experiment data. For an accurate match between experiment and synthetic data, also the extent to which the different features contribute to the XGBoost model's predictions is expected to be similar. Fig. 18 shows the SHAP ranks of all features for the experiment (*Real rank*) and synthetic data (*Syn rank*). The most important features are ranked close to 1 and Fig. 18 shows all features (for both datasets) ordered in descending rank following the experiment data results.

As shown in Fig. 18, 6 out of the 10 features in the top 10 ranked positions for the experiment data are also ranked in the synthetic data's top 10. Furthermore, also for the synthetic data the importance of the *Demographic Data* and *Experiment Data* feature classes for retention prediction remains low. The one exception is the *Retention Interval*, which is ranked 10th for the synthetic data (14 for experiment data): this is the desired result, as the synthetic data was generated to train the XGBoost model with a more high-resolution retention interval dataset. The most surprising difference between the experiment and synthetic data in Fig. 18 is that the *Learing curve p_0*, which was the number 1 ranked feature for the experiment data, is only ranked 16 for the synthetic data. This difference may be at least partially explained by the higher rank of the other main feature that captures the performance at the start of the training phase, *RMSe@First5Trainingruns*.

Overall, the results in this section show that our generated synthetic data provides a reasonable match to the experiment data, but does also show key differences due to assumptions made in the data generation, e.g., different statistical properties. In spite of these differences at the data distribution level, when training the XGBoost model with

synthetic data and analyzing the feature importance with SHAP a similar model with similar feature importance is obtained. Therefore, the synthetic data is still useful for verifying the potential of our propsed XGBoost model approach for individual retention prediction.

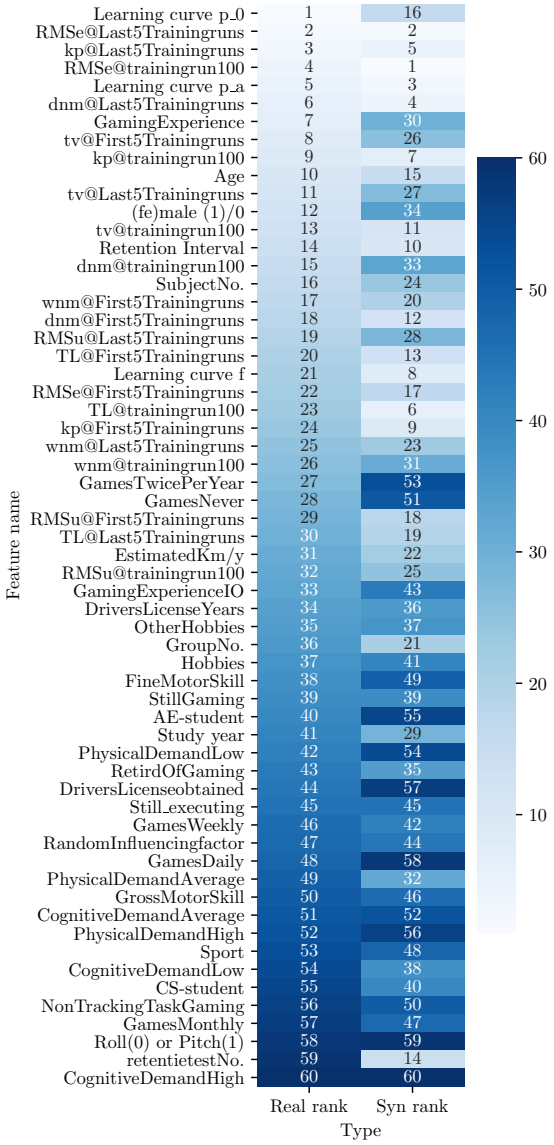
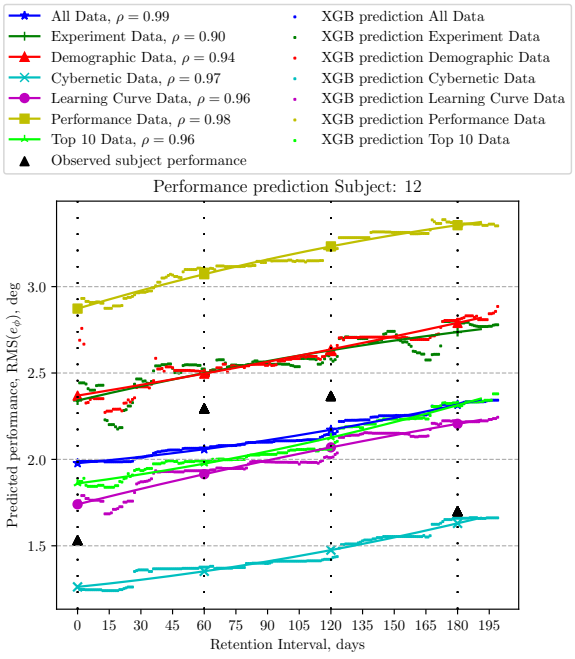


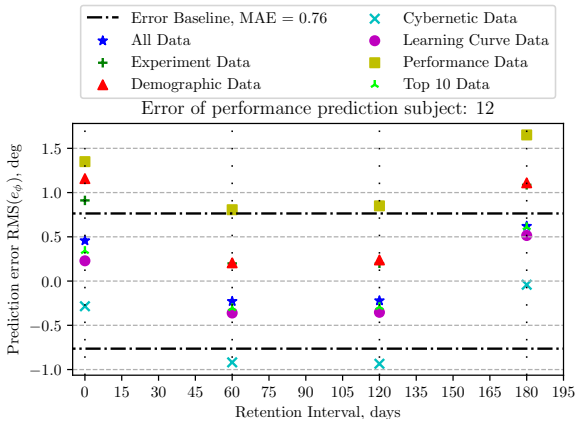
Fig. 18 Comparison of SHAP ranks for all features in the experimental and synthetic data (roll-axis data).

E. Step 5: Synthetic Data Retention Curve Prediction

Using the generated synthetic data from Step 4 of our workflow (see Fig. 10), in the final analysis step we further investigate our approach's capacity for retention curve prediction. For the analysis in this section, an XGBoost model that was trained on a dataset with all features from all feature classes was used. To evaluate the XGBoost retention curve construction, the results for two participants, 12 and 35, are analyzed and shown side-by-side in Fig. 19 and 20, respectively. In both figures, subfigure (a) shows the predicted retention curve with a parabolic fitted curve, while subfigure (b) shows the corresponding prediction errors compared to the experiment data. The black triangles represent the measured *RMSe@retenttest* of both participants at the retention intervals they tested in the experiment of [20], i.e., 60, 120, and 180 days for participant 12, and 90 and 180 for participant 35. Again, the prediction error figures show the baseline MAE value (dashed black lines), as also shown in Fig. 16, for reference.

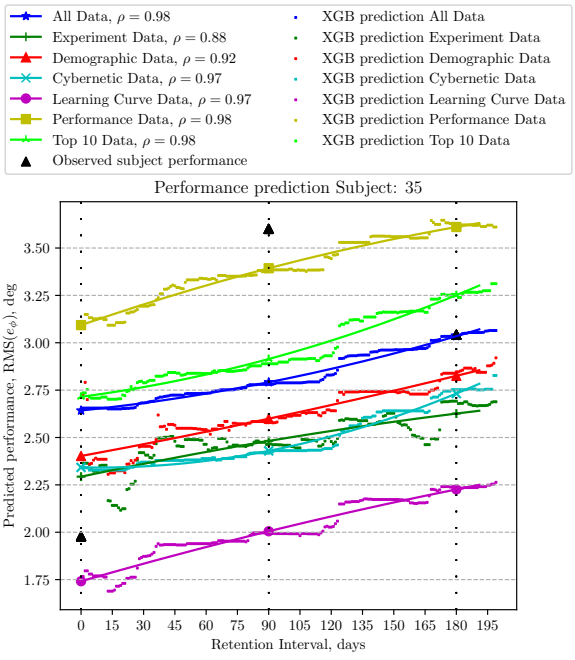


(a) Retention curve prediction

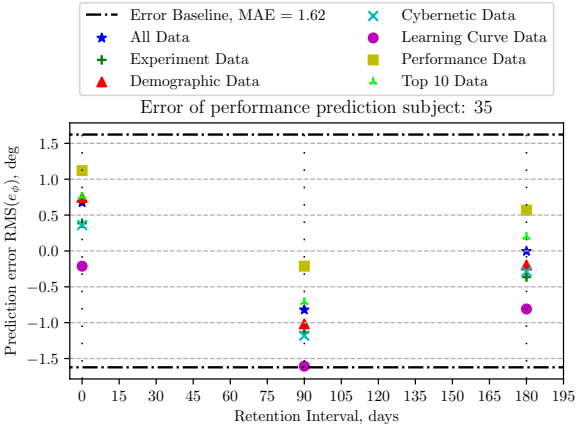


(b) Prediction error

Fig. 19 Synthetic data retention curve predictions for participant 12 (roll-axis data) for all feature classes.



(a) Retention curve prediction



(b) Prediction error

Fig. 20 Synthetic data retention curve predictions for participant 35 (roll-axis data) for all feature classes.

First, Fig. 19 and 20 show that for the synthetic data the predicted performance retention from the XGBoost model is still discontinuous and somewhat erratic, which is a characteristic of decision tree models. Compared to Fig. 15, however, the resolution of the prediction is much higher than obtained with the experiment data. Furthermore, for the synthetic data the predicted skill retention curves are found to be roughly linear, which not only holds for the examples shown in Fig. 19 and 20, but for all participants (see [44]). While this result is due to the assumptions made in the synthetic data generation, it is also more in line with the expected shape of a skill retention curve than the parabolic trajectory shown in Fig. 15.

Furthermore, Fig. 19 and 20 show that the shape of the predicted retention curves for all feature classes are equivalent for both subjects. This is directly explained by our methodology, where the XGBoost model is trained on the data from all participants, and then 'individualized' for individual retention prediction based on input feature values. This also causes different feature classes to be better predictors for some participants than for others: for example, while the *Cybernetic Data* features enable a reasonable fit (with prediction error well below the baseline value) for participant 35, the XGBoost model strongly under-predicts *RMSe@retenttest* for participant 12. Overall, the quality of the retention curve prediction was found to be dependent on the magnitude of the drop in performance (increased *RMSe@retenttest* compared to *RMSe@Last5Trainingruns*) for the different participants. Participant 35's performance ($RMS(e_\phi)$) degraded by a factor 1.62 and XGBoost predictions for this participant are considered good, as the predictions are better than baseline in 95.8% of the cases. Participant 12 showed much less skill decay and the XGBoost model only improves on the baseline prediction in 67.5% of the data points. Overall, these results thus show the capacity for individual predictions of skill retention curves using XGBoost, but also that the success for individual predictions strongly relies on the training data, as well as the extent to which participants match the 'average' trends in the training data.

Fig. 21 summarizes the quality of the retention curve prediction by XGBoost, averaged over all participants, for all feature classes. The color-coded boxplots in Fig. 21 show the MAE between the experiment data (*RMSe@retenttest*) and the XGBoost model's prediction, with a different color for the different retention interval values. The baseline MAE of 0.71 is again indicated with a horizontal black dashed line, for reference, to see where performance is improved/degraded compared to the baseline. The results are grouped for the different considered feature classes, and the total MAE values for all retention intervals (RI) and feature classes are listed in the figure legend.

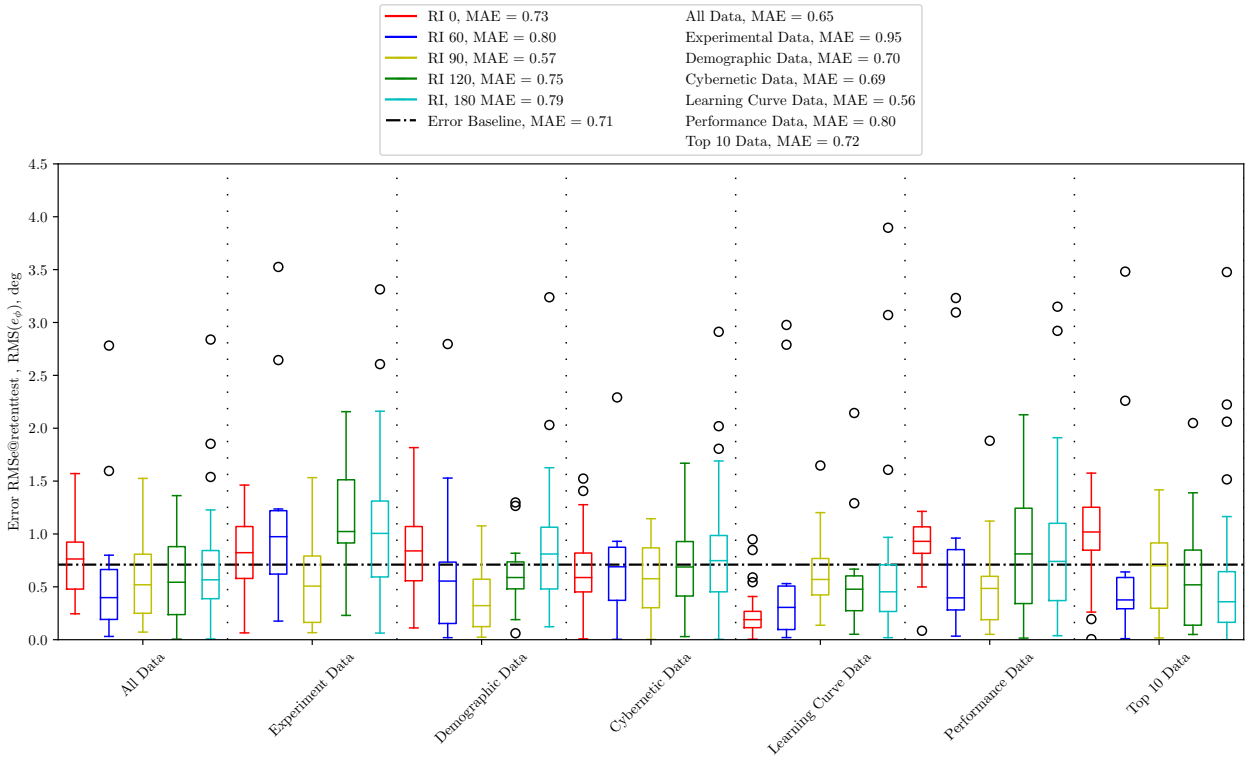


Fig. 21 Synthetic data prediction error boxplot for all participants and feature sets, separated for different retention intervals (RI).

Fig. 21 shows that, on average, the feature class with the lowest MAE (0.56) is the *Learning Curve Data*. With the smallest number of features (3), this feature class facilitates a prediction that is improved by 21% compared to the baseline. This result is consistent with the observations on the experiment data, where also the *Learning Curve Data* class was found to be important for the XGBoost model's predictions. Overall, this result also indicates that the most accurate individual XGBoost retention predictions may be obtained with only a relatively low number of input features. Furthermore, Fig. 21 shows that on average the lowest MAE values (0.57, averaged over all participants and feature classes) are obtained at the 90-day retention interval. This result may be explained by a central tendency of the XGBoost algorithm, which starts its prediction at an average across the whole training dataset, and adds branches and leaves to model trends and details in each constructed tree, see Fig. 7. For our training data, the average retention test performance data is centered on the retention interval at 90 days.

VI. Discussion

The goal of this paper is to determine the capabilities of XGBoost decision tree models for individual performance skill retention prediction for skill-based manual control behavior. For this research, use was made of the training and retention dataset collected for a two-axis (pitch/roll) tracking task in [20], with a focus on the roll-axis data for which the strongest performance degradation after training was observed. The XGBoost model was setup to only use features collected up to the end-of-training as model inputs, while it was trained to predict the (degraded) task performance in the first retention test measurement (*RMSe@retenttest*). Since the experiment dataset was found to be sparse and highly dimensional, the XGBoost and SHAP methods were selected as suitable tools for prediction model implementation and the analysis of its results, and use was made of a generated extended synthetic dataset next to the experiment data.

In our analysis, features that capture the level of tracking performance during initial training (*Performance Data* feature class) – such as *RMSe@Last5Trainingruns*, *Learning curve p_0*, and *Learning curve p_a* – were consistently found to have a high SHAP rank and were thus important for the XGBoost model's predictions. This expected dominant presence of training performance as a predictor of skill retention is in accordance with earlier findings in literature [6, 13, 45, 46]. The better the level of performance at the end-of-training, and the larger the performance improvement during training, the longer skills are generally maintained [20]. This also explains the success of the XGBoost model with the *Top 10* feature class considered in our paper, which includes a number of the initial and final training run performance features and is found to explain the model's predictions for 69.6% when trained with the experiment data. While these results indicate that such training performance-related features are crucial for predicting skill retention, further research is required to further optimize this and, for example, verify the effects of eliminating the multiple correlated features in our *Performance Data* class.

Based on literature on skill retention [6, 12–16, 47], the feature class of *Demographic Data* considered in this paper was expected to have a crucial role in predicting skill retention at the individual level. Our analysis based on the experiment data of [20], however, showed that the SHAP rank of the *Demographic Data* features was generally low, indicating they only had a marginal contribution to the XGBoost model's prediction. This low impact on retention prediction was consistently found based on both our experiment and synthetic datasets. While clear from our analysis of the data of [20], in our view this result is not expected to generalize to other tasks and datasets. Hence, it is of the utmost importance for further research on skill retention and its prediction with ML models that demographic data on the participants' background is still collected and included.

Overall, the analysis in our paper suggests that the XGBoost model achieves the best prediction performance with a small number of input features. For example, Fig. 13 shows that the model's retention predictions improve when successively adding up to four features. This suggests that the XGBoost model when applied to this type of data may be quite prone to overfitting, as with more features included also more features will contribute to the model's prediction, but at an overall decreasing SHAP rank, which may reduce consistency and susceptibility to data noise. This preferred low number of features is also consistent with the good prediction performance obtained with the class of *Learning Curve Data* features (see, e.g., Fig. 19) and the overall lowest MAE obtained by this class (0.56 $RMS(e_\phi)$) across the complete synthetic dataset. To further investigate this point, future work should directly compare XGBoost models with small matched sets of high and low SHAP-ranked features, to separate the effects of the number of input features and the input feature SHAP rank.

In this paper, the XGBoost model is trained and tested both on the original experiment data of [20], but also on a generated (and matched) synthetic dataset. In this paper, the different sizes of these two datasets, as well as key assumptions that had to be made when generating the synthetic data, complicate a direct comparison between the results obtained with both datasets. In this paper, our choice to use a multivariate Gaussian process tuned to match

the experiment data for synthetic data generation was found to result in mismatched data distributions, as well as inconsistencies in the identified importance of certain key data features (e.g., *Learning curve p_0*). As the use of synthetic data for model training would greatly benefit an inherently sparse data application as individual retention prediction, improved methods for synthetic data generation, such as advanced statistical bootstrapping methods or generative neural networks, should be investigated.

A skill retention curve – i.e., the trajectory along which task performance degrades during periods of inactivity – has been reported have a linear, positively-accelerating, or negatively-accelerating shape in literature [2, 20]. In this paper, we model the retention curves extracted from the XGBoost models as second-order polynomials, which are found to describe a parabolic and linear relation with the retention interval for our experiment and synthetic datasets, respectively. While the parabolic retention curves found when using the experiment data are an artefact resulting from the very good performance of many participants in the final (180-day) retention test in [20], especially the linear retention curves found from our synthetic data match the theory of [2]. Furthermore, extreme care should be taken using a one-dimensional fitted retention curve model as performed in this paper [16], as this occludes the model's inherent sensitivity to the underlying features besides the retention interval. For example, [48] states that skill decay is not only a function of time, but rather of external processes that occur over time. For this reason, using a true data-driven model, such as our considered XGBoost model, for retention prediction is clearly the safest, and superior, approach.

The selected greedy approach to determining the XGBoost model's hyperparameter ranges for the different feature classes restricts the freedom of the *GridSearch CV* approach and thereby eases application due to reduced computational cost. Moreover, due to the selected (large) step size (increment), the performed hyperparameter optimization was not fully optimal. A second difficult aspect in our analysis is the XGBoost model's hyperparameters needed to be optimized separately for each considered feature class. This implies that the comparisons between different feature classes in this paper may also be influenced by different hyperparameter settings between compared cases. To limit this effect, in this paper we always compare optimal XGBoost settings for all feature classes, to still see the overall best-performing feature classes. For follow-up studies, we advise to conduct a more in-depth and complete hyperparameter analysis, which would also enable for a single optimally-tuned XGBoost model to be compared across all feature classes. Furthermore, the use of Bayesian optimization techniques [49, 50] may be of use to further improve the prediction performance of the XGBoost model.

Overall, the XGBoost model structure considered in this paper for the prediction of individual retention curves shows some promise for further development. This study was performed as part of a collaboration between TU Delft and the Royal Netherlands Aerospace Centre (NLR), which focuses on skill training and retention to develop optimal training programs and novel approaches to 'learning analytics' [51]. While in the current paper we focus on the retention of purely skill-based manual control skills [20], for many practical applications, in- and outside of aerospace, this often is only one dimension of the more complex skills involved. While it may be expected that features falling into our classes of *Performance Data* and *Learning Curve Data* would also be important for skill retention prediction in more complex tasks, more research is needed to verify if the required additional number of input features would still enable meaningful and consistent XGBoost model predictions. For this, a similar approach as followed in this paper, making use of SHAP and generated synthetic data, could be applied.

VII. Conclusion

The goal of this paper was to determine the effectiveness of individual skill retention performance predictions obtained for a skill-based tracking task using XGBoost decision tree models. For this, a prediction model structure is used that only considers participant data up to the end-of-training, compared across six different feature classes, as the candidate input features to predict the resulting (degraded) level of task performance as a function of the retention interval (inactivity period). Since the considered experiment dataset is sparse and highly dimensional, the importance of the different considered input features for the XGBoost model's prediction was assessed using the SHAP method. From the performed SHAP analysis, it was found that especially the features that quantify the participants' performance during training and their learning curves are important for the model's retention performance prediction. In particular the considered feature class of *Learning Curve Data* is found to improve individual skill retention prediction over the full 180-day retention interval by 21% (0.56 $RMS(e_\phi)$ MAE) compared to a baseline for which no skill decay is assumed. For both the experiment (cross-validated) and synthetic data (non-cross-validated), unexpectedly negligible contributions were found for features that encode participants' individual characteristics (*Demographic Data*). Overall, with a full set of 60 input features, the *Top 10* in terms of their SHAP rank were found to account for 69.4% of the model's prediction and XGBoost was found to be most accurate and consistent with 4 or less input features. The approach to skill retention

prediction considered in this paper thus shows great potential for predicting an individual's performance over a period of inactivity and should be further developed towards more complex tasks and real-world applications in future work.

References

[1] Miller, L., "ASTD 2012 state of the industry report: Organizations continue to invest in workplace learning," , Association for Talent Development, 2012. URL https://www.td.org/magazines/td-magazine/astd-2012-state-of-the-industry-report-organizations-continue-to-invest-in-workplace-learning.

[2] Ebbinghaus, H., *Memory : a contribution to experimental psychology*, Teachers College, Columbia University, New York, 1913.

[3] Sitterley, T. E., and Berge, W. A., "Degradation of learned skills. Effectiveness of practice methods on simulated space flight skill retention," Report, The Boeing Company, Seatle Washington, 1972.

[4] Fleishman, E. A., and Parker Jr, J. F., "Factors in the retention and relearning of perceptual-motor skill," *Journal of Experimental Psychology*, Vol. 64, No. 3, 1962, pp. 215–226. https://doi.org/10.1037/h0041220.

[5] Youngling, E. W., Sharpe, E. N., Ricketson, B. S., and McGee, D. W., "Crew Skill Retention for Space Missions up to 200 Days," *Gardlin, G. R., Degradation of learned skills*, , No. F766, 1968.

[6] Schendel, J. D., Shields, J., and Katz, M., "Retention of motor skills: Review," *U. S. Army Research Institute for the Behavioral and Social Sciences*, 1978.

[7] Prophet, W. W., "Long-Term Retention of Flying Skills: A Review of the Literature," Final Report 76-35, Human Resources Research Organization, 300 North Washington Street, Alexandria (VA), 1976.

[8] Wright, R. H., "Retention of Flying Skills and Refresher Training Requirements: Effects of Non-Flying and Proficiency Flying," Tech. Rep. HumRRO-TR-73-32, Human Resources Research Organization, 300 North Washington Street, Alexandria (VA), 1973.

[9] Wexley, K. N., and Latham, G. P., *Developing and Training Human Resources in Organizations*, Prentice-Hall, 2002. URL https://books.google.nl/books?id=elHGQgAACAAJ.

[10] Arthur, W. J., and Bennett, W. J., "Skill retention and decay: A meta-analysis," Final Report AL/HR-TR-1996, Technical Training Division, Brooks Air Force Base (TX), 1996.

[11] Bjork, R., *Memory and Meta-memory Considerations in the Training of Human Beings*, Metacognition: Knowing about Knowing, The MIT Press, Cambridge (MA), 1994, pp. 185–205.

[12] Arthur, W., *Individual and Team Skill Decay: The Science and Implications for Practice*, Applied Psychology Series, Brunner-Routledge, 2013. URL https://books.google.nl/books?id=BFnvygAACAAJ.

[13] Farr, M. J., *The Long-Term Retention of Knowledge and Skills: A Cognitive and Instructional Perspective*, 1st ed., A Cognitive and Instructional Perspective, Springer-Verlag New York, 1986. https://doi.org/10.1007/978-1-4612-1062-7.

[14] Gardlin, G. R., and Sitterley, T. E., "Degradation of Learned Skills: A Review and Annotated Bibliography," NASA Contractor Report D180-15080-1, National Aeronautics and Space Administration, 1972.

[15] Hurlock, R. D., and Montague, W. E., "Skill Retention And Its Implications For Navy Tasks: An Analytical Review," Tech. rep., Navy Personnel Research and Development Center San Diego, California 92152, 1982.

[16] Naylor, J. G., and Briggs, G. E., "Long-Term Retention of Learned Skills: A Review of the Literature," ASD Technical Report AD0267043, US Air Force Systems Command, Aeronautical Systems Division, Wright-Patterson Air Force Base (OH), 1961.

[17] Gronlund, S. D., and Kimball, D. R., *Remembering and forgetting: From the laboratory looking out*, Applied psychology series., Routledge/Taylor and Francis Group, New York, NY, US, 2013, pp. 14–52.

[18] Sense, F., Wood, R., Collins, M. G., Fiechter, J., Wood, A., Krusmark, M., Jastrzembski, T., and Myers, C. W., "Cognition-Enhanced Machine Learning for Better Predictions with Limited Data," *Topics in Cognitive Science*, 2021. https://doi.org/10.1111/tops.12574.

[19] Riesterer, N., Brand, D., and Ragni, M., "Predictive Modeling of Individual Human Cognition: Upper Bounds and a New Perspective on Performance," *Topics in Cognitive Science*, Vol. 12, No. 3, 2020, pp. 960–974. https://doi.org/10.1111/tops.12501.

[20] Wijlens, R., Zaal, P. M. T., and Pool, D. M., "Retention of Manual Control Skills in Multi-Axis Tracking Tasks," *Proceedings of the AIAA Modeling and Simulation Technologies Conference, Orlando (FL)*, 2020. https://doi.org/10.2514/6.2020-2264.

[21] Barendswaard, S., Pool, D. M., Van Paassen, M. M., and Mulder, M., "Dual-Axis Manual Control: Performance Degradation, Axis Asymmetry, Crossfeed, and Intermittency," *IEEE Transactions on Human-Machine Systems*, Vol. 49, No. 2, 2019, pp. 113–125. https://doi.org/10.1109/thms.2019.2890856.

[22] Mulder, M., Pool, D. M., Abbink, D. A., Boer, E. R., Zaal, P. M. T., Drop, F. M., Van Der El, K., and Van Paassen, M. M., "Manual Control Cybernetics: State-of-the-Art and Current Trends," *IEEE Transactions on Human-Machine Systems*, Vol. 48, No. 5, 2018, pp. 468–485. https://doi.org/10.1109/THMS.2017.2761342.

[23] McRuer, D. T., and Jex, H. R., "A Review of Quasi-Linear Pilot Models," *IEEE Transactions on Human Factors in Electronics*, Vol. HFE-8, No. 3, 1967, pp. 231–249. https://doi.org/10.1109/thfe.1967.234304.

[24] Wijlens, R., "Retention of Manual Control Skills in Multi-Axis Tracking Tasks," Master's thesis, Delft University of Technology, Faculty of Aerospace Engineering, 2019. URL http://resolver.tudelft.nl/uuid:db667573-b54e-4754-bf31-2749757f3053.

[25] Pool, D. M., Harder, G. A., and Van Paassen, M. M., "Effects of simulator motion feedback on training of skill-based control behavior," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 4, 2016, pp. 889–901. https://doi.org/10.2514/1.G001603.

[26] Zaal, P. M. T., and Sweet, B. T., "Identification of Time-Varying Pilot Control Behavior in Multi-Axis Control Tasks," *Proceedings of the AIAA Modeling and Simulation Technologies Conference 2012, Minneapolis (MN)*, 2012.

[27] Zaal, P. M. T., and Pool, D. M., "Multimodal Pilot Behavior in Multi-Axis Tracking Tasks with Time-Varying Motion Cueing Gains," *Proceedings of the AIAA Modeling and Simulation Technologies Conference, National Harbor (MD)*, 2014. https://doi.org/10.2514/6.2014-0810.

[28] Zaal, P. M. T., "Manual control adaptation to changing vehicle dynamics in roll-pitch control tasks," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 5, 2016, pp. 1046–1058. https://doi.org/10.2514/1.G001592.

[29] Zaal, P. M. T., and Mobertz, X. R. I., "Effects of Motion Cues on the Training of Multi-Axis Manual Control Skills," *Proceedings of the AIAA Modeling and Simulation Technologies Conference, Denver (CO)*, 2017. https://doi.org/10.2514/6.2017-3473.

[30] Chen, T., and Guestrin, C., "XGBoost: A Scalable Tree Boosting System," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, p. 785–794. https://doi.org/10.1145/2939672.2939785.

[31] Wan, Z., Xu, Y., and Šavija, B., "On the Use of Machine Learning Models for Prediction of Compressive Strength of Concrete: Influence of Dimensionality Reduction on the Model Performance," *Materials*, Vol. 14, No. 4, 2021, p. 713. https://doi.org/10.3390/ma14040713.

[32] Friedman, J., Hastie, T., and Tibshirani, R., "Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors)," *The Annals of Statistics*, Vol. 28, No. 2, 2000, pp. 337–407. https://doi.org/10.1214/aos/1016218223.

[33] Hastie, T., Tibshirani, R., and Friedman, J. H., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer series in statistics, Springer, 2009. URL https://books.google.nl/books?id=eBSgoAEACAAJ.

[34] Yang, F., Wang, D., Xu, F., Huang, Z., and Tsui, K.-L., "Lifespan prediction of lithium-ion batteries based on various extracted features and gradient boosting regression tree model," *Journal of Power Sources*, Vol. 476, 2020, p. 228654. https://doi.org/10.1016/j.jpowsour.2020.228654.

[35] Hak Lee, E., Kim, K., Kho, S.-Y., Kim, D.-K., and Cho, S.-H., "Estimating Express Train Preference of Urban Railway Passengers Based on Extreme Gradient Boosting (XGBoost) using Smart Card Data," *Transportation Research Record: Journal of the Transportation Research Board*, 2021. https://doi.org/10.1177/03611981211013349.

[36] Yang, C., Chen, M., and Yuan, Q., "The application of XGBoost and SHAP to examining the factors in freight truck-related crashes: An exploratory analysis," *Accident Analysis and Prevention*, Vol. 158, 2021, p. 106153. https://doi.org/10.1016/j.aap.2021.106153.

[37] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, Vol. 12, No. 5 January, 2011, pp. 2825–2830.

[38] Lundberg, S., and Lee, S., "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems*, Vol. 30, 2017. URL http://arxiv.org/abs/1705.07874.

[39] Qin, C., Zhang, Y., Bao, F., Zhang, C., Liu, P., and Liu, P., "XGBoost Optimized by Adaptive Particle Swarm Optimization for Credit Scoring," *Mathematical Problems in Engineering*, Vol. 2021, 2021, pp. 1–18. https://doi.org/10.1155/2021/6655510.

[40] Parsa, A. B., Movahedi, A., Taghipour, H., Derrible, S., and Mohammadian, A., "Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis," *Accident Analysis and Prevention*, Vol. 136, 2020. https://doi.org/10.1016/j.aap.2019.105405.

[41] Sherstinsky, A., "Fundamentals of Recurrent Neural Network RNN and Long Short-Term Memory LSTM Networks," *Physica D: Nonlinear Phenomena*, Vol. 404, 2020. https://doi.org/10.1016/j.physd.2019.132306.

[42] Gron, A., *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'Reilly Media, Inc., 2017.

[43] Efroymson, M., "Multiple regression analysis," *Mathematical methods for digital computers*, 1960, pp. 191–203.

[44] Van Leeuwen, B. A. A., "Personalised Prediction of Skill Development and Retention," Master's thesis, TU Delft, Faculty of Aerospace Engineering, 2022. URL http://resolver.tudelft.nl/uuid:6d284d4b-c3ca-4381-8c33-ade7872263d5.

[45] Grimsley, D. L., *Acquisition, Retention, and Retraining: Group Studies on Using Low Fidelity Training Devices*, Technical report, George Washington University, Human Resources Research Office, 1969. URL https://books.google.nl/books?id=_5tkp1UVCikC.

[46] Purdy, B. J., and Lockhart, A. S., "Retention and Relearning of Gross Motor Skills after Long Periods of No Practice," *Research Quarterly. American Association for Health, Physical Education and Recreation*, Vol. 33, 1962, pp. 265–272.

[47] Osgood, C. E., "The similarity paradox in human learning: a resolution," *Psychological Review*, Vol. 56, No. 3, 1949, pp. 132–143. https://doi.org/10.1037/h0057488.

[48] McGeoch, J. A., "Forgetting and the law of disuse," *Psychological Review*, Vol. 39, 1932, pp. 352–370.

[49] Appleby, G., Espadoto, M., Chen, R., Goree, S., Telea, A., Erik, and Chang, R., "HyperNP: Interactive Visual Exploration of Multidimensional Projection Hyperparameters," *arXiv pre-print server*, 2021. URL https://arxiv.org/abs/2106.13777.

[50] Sun, L., "Application and Improvement of Xgboost Algorithm Based on Multiple Parameter Optimization Strategy," *Proceedings of the 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, IEEE, 2020. https://doi.org/10.1109/icmcce51767.2020.00400.

[51] Van der Pal, J., and Toubman, A., *An Adaptive Instructional System for the Retention of Complex Skills*, Springer International Publishing, 2020, pp. 411–421. https://doi.org/10.1007/978-3-030-50788-6_30.